OUTER RACE FAULT IDENTIFICATION UNDER UNKNOWN ROTATIONAL SPEED AND LOW SAMPLING RATE

A Dissertation
Presented to
The Academic Faculty

By

Fabia U. Bayer

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

Georgia Institute of Technology

August 2020

OUTER RACE FAULT IDENTIFICATION UNDER UNKNOWN ROTATIONAL SPEED AND LOW SAMPLING RATE

Approved by:

Dr. Christopher Saldana, Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Thomas Kurfess, Co-Advisor
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Oliver Sawodny
Institute for System Dynamics
University of Stuttgart

Dr. Cristina Tarín
Institute for System Dynamics
University of Stuttgart

Date Approved: July 1, 2020

Curiosity is the desire to be surprised.

M. Touissant

ACKNOWLEDGEMENTS

I would like to especially thank my advisor, Dr. Christopher Saldana. Not only did his guidance and our discussions help me develop many crucial academic and professional skills, I am also very grateful for his support in making my thesis such a unique trans-atlantic adventure due to the COVID-19 pandemic.

I thank Dr. Oliver Sawodny for his steady and enthusiastic involvement in the joint degree, for making sure the optimal oppurtunities can be presented to the students, and for his continued and genuine interest in our success.

Further, I thank Dr. Thomas Kurfess and Dr. Cristina Tarín for agreeing to serve on my committee and to spend time reviewing my thesis.

I am also very grateful to our partners at Georgia-Pacific, particularly Kevin Ray, Gary Burkett, Jeff Peterson, Sylvester Williams, David Hutcherson, Samuel Coyne, Paul Frederickson and Eymard Prevost. The time that I was able to spend working on the project was very enlightening and enriching. Our cooperation inspired this thesis and played a crucial part in equipping myself with the right knowledge and methods to contribute to the scientific field.

A great thanks is also dedicated to my lab colleagues of the digital team, especially Pierrick Rauby and Marcel Neumann, for fruitful discussions, all the help in difficult situations and just friendship in general.

Last but not least, I would like to thank my family for their unfailing help and emotional support.

Fabia U. Bayer

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

BPFO	Ball Pass Frequency: Outer Race
BPFI	Ball Pass Frequency: Inner Race
FTF	Fundamental Train Frequency
BSF	Ball Spin Frequency
FFT	Fast Fourier Transform
IFFT	Inverse Fast Fourier Transform
DFT	Discrete Fourier Transform
STFT	Short-Time Fourier Transform
TFR	Time-Frequency Representation
RMS	Root Mean Square
SNR	signal to noise ratio
TFCE	Time-Frequency Curve Extraction
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative

SUMMARY

Rolling element bearing fault identification is an important sub-topic of predictive health monitoring. Most state-of-the-art fault identification approaches utilize bearing configuration, shaft rotational speed and/or bearing harmonics that are only present in the spectrum if the sampling rate is sufficiently high. In industrial application, these three factors are often not available. This thesis investigates the performance of various state-of-the-art bearing fault identification approaches under unknown rotational speed and bearing configuration for a range of sampling rates. The aim of this thesis is to give recommendations for effective bearing fault identification under uncertain low-sampling rate circumstances. The recommendations are based on simulated and experimental data.

CHAPTER 1
INTRODUCTION

1.1 Motivation

Sophisticated machine design approaches try to maximize the longevity of the considered machinery. However, all machinery wears down over time and develops faults, eventually rendering the machine unable to continue performing the task for which it was designed. Naturally, the point in time when a machine part is replaced greatly influences the overall cost of maintaining that machine. If a damaged part is not replaced early enough, this can lead to catastrophic failure, damaging other machine parts or manufactured items or inducing costs through unplanned machine downtime. On the other hand, if a part is replaced too early before it has developed a fault, avoidable cost for increased machine downtime as well as the acquisition and installation of replacement parts is caused. In some cases, unwarranted maintenance can even increase the probability of a fault appearing [1]. Methods for finding the optimal replacement time can be divided into time-based and condition-based or predictive maintenance [2].

Time-based maintenance is a traditional maintenance strategy wherein maintenance decisions such as expected runtime and preventive repair intervals are determined based on statistical values like failure time data or use-based data [2]. Because catastrophic failure is generally less desired than early replacement, the scheduled replace time is chosen earlier than the median estimated failure time, often leading to replacement of potentially healthy parts [3]. Condition-based maintenance, on the other hand, is a more modern maintenance approach wherein maintenance actions are recommended based on information obtained by a condition monitoring process [4]. Almost all equipment failures are preceded by certain signs, conditions or indications in vibration, temperature and noise levels [5]. The amount of maintenance departments in the United States that employ a form of condition-based maintenance has increased in recent years [1].

To be able to perform condition-based maintenance effectively, it is critical to have powerful tools for the detection of bearing defects. Many typical machinery faults can be detected by vibration analysis [6]. As bearing faults induce impulses into the bearing, the vibration pattern of the bearing and its surroundings changes when a fault develops, which can be identified by vibration measurements. Among the defects that are responsible for a large amount of machine failures in rotating machines are imbalance faults [7] and misalignment [1]. However, the majority of problems in rotating machinery are caused by bearing faults [8], one of which is the ball pass frequency outer race (BPFO) defect.

State-of-the-art bearing fault detection methods usually make two primary assumptions: The sampling rate is high enough that the harmonic oscillation of the bearing can be measured, and parameters of shaft rotation speed and bearing configuration are known [3, 9, 10]. However, in application, these assumptions are not necessarily true considering real world shop maintenance scenarios. With regard to the former, accelerometers and measurement systems capable of capturing and storing high frequency measurements are significantly more expensive than those with lower frequency ranges. With regard to the latter, potential unavailability of tachometers can limit availability of shaft rotation data and rapid replacement of bearings with alternate hardware or insufficient documentation can lead to unknown bearing configurations.

Therefore, the focus of this research is to understand performance of various bearing defect detection approaches with respect to their applicability to low sampling rates and unknown system configurations (e.g., rotational speeds, bearing parameters).

1.2 Aim and Scope

The present work aims to provide insights on suitable BPFO fault detection methods for unknown shaft rotation speeds and low sampling rates. Two main research questions are to be answered:

1. What is the impact of sampling rate on effectiveness of state-of-the-art detection methods for ball bearing fault identification under unknown shaft rotational speeds?

2. Which BPFO detection method is best suited to deal with low sampling rates?

To answer these questions, three detection methods with increasing degree of complication are considered. The first approach is a novel straightforward peak finding approach for the lowest BPFO harmonics. The second approach also employs peak finding of the lower harmonics, but in a preprocessing step, the main impulsive frequency band is identified with help of the spectral kurtosis and then extracted by a bandpass filter. After this, envelope demodulation is carried out before searching for peaks. The third approach tries to identify the first bearing harmonic as well as the shaft rotation signature from a time-frequency representation. The ratio of both in time is compared because it should be constant for a BPFO fault. A variety of experimental and simulated datasets are used and these approaches are compared using scores like precision and recall, as well as analyzing the progression of experimental data over time.

1.3 Structure

The remainder of the present work is structured as follows. In Section 2, background topics concerning rolling element bearings, spectral analysis and performance measures of classifiers are revisited. Subsequently, the three fault detection algorithms compared in the present work are introduced in Section 3. The introduced approaches are the novel direct peak finding approach, an envelope demodulation approach as well as a Time-Frequency Curve Extraction (TFCE) approach. In Section 4, the methodology of the experiments employed in this work is visited. Simulated data generation as well as the experimental dataset are introduced, and the central assumptions are stated. Section 5 covers the central results and discussion. After validating the approaches with simulated data, the confusion matrices and scores of the various approaches at varying sampling rates are given. After analyzing possible causes of misclassification, recommendations for choosing an approach are drawn. Finally, in Section 6, an overall conclusion is drawn and some future work is suggested.

CHAPTER 2

THEORETICAL BACKGROUND

The analysis of bearing faults requires some knowledge of bearing characteristics as well as signal processing. This chapter provides a short overview about the necessary foundational concepts used in later chapters. For a more in-depth treatment of these topics, [3] and the references therein are recommended.

2.1 Defects in Rolling Element Bearings

A rolling element bearing consists of four main components: outer and inner race, cage and rolling elements [11]. A schematic image of a ball bearing can be found in Figure 2.1. When the inner race turns against the outer race, the rolling elements are set into a rolling motion and are held in place by the cage. The rolling results in a significant friction reduction and therefore rolling element bearings are very common in manufacturing machines, where parts turning against each other are crucial.

Important geometric parameters of rolling element bearings are shown in Figure 2.1 for a ball bearing. Important parameters for a rolling element bearing are shaft frequency, number of rolling elements, pitch and rolling element diameter, load angle and the resonance frequency of the bearing. The parameters and their meanings are summarized in

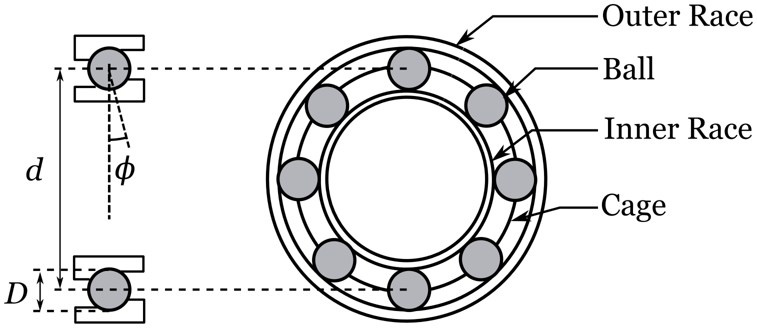

Figure 2.1: Structure of a ball bearing [12] with modified parameter identifiers.

Table 2.1: Bearing geometry parameters.

Symbol	Meaning
n	Number of rolling elements
f_r	Shaft frequency
D	Pitch diameter
d	Rolling element diameter
ϕ	Load angle from radial plane
ω_B	Primary resonance frequency of the bearing

Table 2.1.

Bearing failure is one of the most common reasons for catastrophic machine breakdown in manufacturing machines [3]. A bearing fault occurs when one of the above mentioned bearing parts develops a spall or crack. When another part of the bearing, usually a rolling element, hits the crack, this results in a small impulsive hit on the bearing, and this adds an impulse and the corresponding high-frequency impulse response to the vibration signature of the bearing [3]. For constant rotational speeds of the bearing, the rolling elements hit the crack repeatedly and approximately periodically, with a frequency depending on the rotational speed and the bearing configuration. This is illustrated in Figure 2.2 for an outer race defect. Two impulsive hits of the bearing and the resulting vibration with the bearing resonance frequency, exponentially decaying in amplitude, can be observed. Outer race defects incur impulse trains at the BPFO frequency, similarly inner race defects incur impulses at the Ball Pass Frequency: Inner Race (BPFI) frequency. Cage defects occur at Fundamental Train Frequency (FTF) and rolling element defects at Ball Spin Frequency (BSF).

According to Randall [3], the defect frequencies can then be determined as shown in Table 2.2, assuming no slip. Slip between bearing elements usually induces a period time deviation of around $1\% - 2\%$ around the calculated period time [3], which is the reciprocal of the fault frequency. The main part of the present work will concentrate on outer race (BPFO) fault identification. It is notable that the BPFO, as well as BPFI and FTF, depend linearly on the shaft frequency f_r and on bearing geometry parameters. However, in indus-

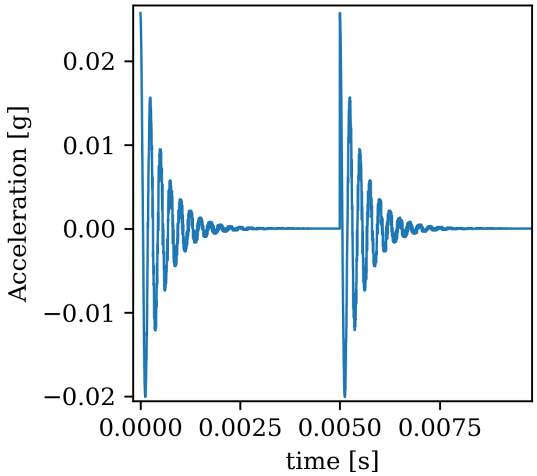

Figure 2.2: Schematic ideal signature of a BPFO fault.

Table 2.2: Defect frequencies for common bearing defects.

BPFO	$\frac{nf_r}{2}\left(1-\frac{d}{D}\cos\phi\right)$	(2.1)
BPFI	$\frac{nf_r}{2}\left(1+\frac{d}{D}\cos\phi\right)$	(2.2)
FTF	$\frac{f_r}{2}\left(1+\frac{d}{D}\cos\phi\right)$	(2.3)
BSF	$\frac{D}{2d}\left[1-\left(\frac{d}{D}\cos\phi\right)^2\right]$	(2.4)

trial application, this information may be unknown for various reasons, such as insufficient documentation of bearing replacements and lack of tachometers.

2.2 Signal Processing and Spectral Analysis

Vibration signals can be expressed by various different physical quantities. More precisely, the displacement, velocity and acceleration of a component can be measured. In vibration analysis for bearing fault detection, accelerometers are the most common sensors in practice, due to their frequency range, high accuracy and simple installation [13]. However, accelerometers are highly sensitive to ambient and measurement noise. While technical insight into accelerometers is not within the scope of this work, interested readers are referred to [6, 14] and references therein.

An accelerometer produces an analog voltage signal [14]. The signal is then quantized and sampled into a resulting digital signal [15], which is then directly processed or stored for further processing. The Fourier Transform is used to identify frequency components in a time signal. For a discrete-time (i. e., already sampled) signal $s[n]$ of length N, its Discrete Fourier Transform (DFT) is defined to be:

$$S[k] = \sum_{n=0}^{N-1} s[n] e^{-j\frac{2\pi kn}{N}} \tag{2.5}$$

and its inverse transform is:

$$s[n] = \frac{1}{N} \sum_{k=0}^{N-1} S[k] e^{j\frac{2\pi kn}{N}} . \tag{2.6}$$

The discrete function $S[k]$ is called the spectrum of the signal $s[n]$ [16].

The DFT spectrum as obtained from (2.5) is complex, even when the signal $s[n]$ is real. However, in that case, to obtain a real-valued signal from the Inverse Fast Fourier Transform (IFFT) (2.6), the frequency components larger than $\frac{N}{2}$ are exactly the complex conjugates of their mirrored counterparts and when combining them in the synthesis equation (2.6), the real-valued synthesis equation:

$$s[n] = \frac{1}{N} \sum_{k=0}^{\frac{N}{2}} A[k] \cos\left(\frac{2\pi kn}{N} \varphi[k]\right) \tag{2.7}$$

results, where $A[k]$ is the amplitude and $\varphi[k]$ is the phase of the complex variable $S[k]$. For real signals, it is therefore usually sufficient to regard only the amplitudes of the nonnegative frequency DFT components to determine vibration frequencies of certain components.

From the synthesis equation (2.7), it is evident that the k-th sample can be associated with frequency $k\frac{f_s}{N}$ [3], where f_s is the sampling frequency of the original signal $s[n]$. Due to the implicit N-periodicity of the DFT [16], the second half of the frequency components can be shifted down one period, yielding a spectrum whose frequencies are distributed symmetrically around the origin and stretch from $-\frac{f_s}{2}$ to $\frac{f_s}{2}$.

Computing the DFT directly using Equation (2.5) has a computational complexity of $\mathcal{O}\left(n^2\right)$ [16]. However, certain symmetry properties of the DFT can be exploited to reduce the computational complexity to $\mathcal{O}\left(N \log_2 N\right)$, without changing any mathematical properties. Two of the most common algorithms that use these reductions are the D-I-T and the radix-2-algorithm [16]. If the DFT is computed in this efficient manner, it is called Fast Fourier Transform (FFT). To emphasize the importance of using these more efficient algorithms, the term FFT is used for computation of the discretized spectrum from now on, but still all introduced DFT properties hold.

2.2.1 Fast Fourier Transform Pitfalls

When using the FFT to make statements about an analog process that was measured and then discretized, as is the case with the vibration measurements considered in the present work, some pitfalls have to be avoided and kept in mind. In particular, aliasing effects, leakage effects and picket fence effects can occur.

Aliasing effects From the Real Synthesis Equation (2.7), it is clear that only vibrations in frequencies lower than half the sampling frequency can be reconstructed from the FFT spectrum. As stated in the Nyquist-Shannon-Theorem [3], the information about higher frequency vibration is lost in the sampling process and is consequently misinterpreted as lower frequency components afterwards. This is illustrated in Figure 2.3. The analog signal at a high frequency is pictured in black. The sampling points are denoted by circles, spaced at a sampling rate lower than half the frequency of the black signal. If all samples are connected by the most low-frequency periodic signal possible, the red signal results. Therefore, it is impossible to determine whether the black or the red signal was present

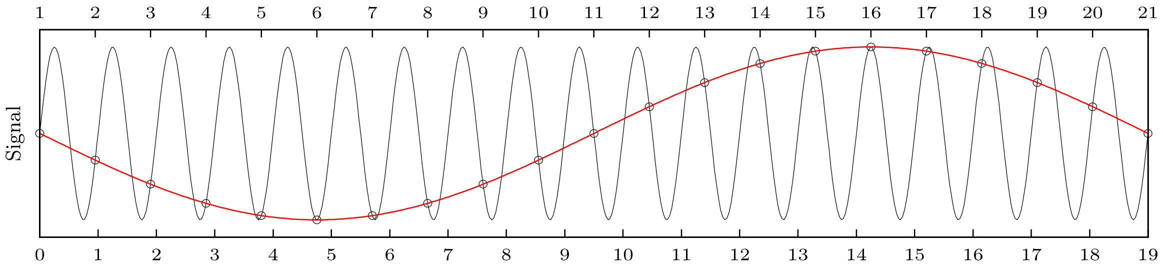

Figure 2.3: Illustrative figure of aliasing effect [17].

originally, if only the samples are known. Since this issue appears directly when sampling the analog signal, it must be addressed in the analog domain before sampling and storing the sampled data. Usually, an analog lowpass-filter with a cutoff frequency lower than the Nyquist frequency $\frac{f_s}{2}$ is applied before sampling to at least get an accurate representation of all frequency components lower than the Nyquist frequency.

Leakage Effects The FFT only considers a finite length section of a potentially infinite original signal. This can be interpreted as a multiplication by a rectangle window of length N [16]. However, in the frequency domain, this multiplication corresponds to a convolution with the FFT of the rectangle window, which has the characteristics of a sinc function. This leads to a smearing of a frequency peak across neighboring frequencies, which is referred to as leakage. To reduce leakage, the implicit rectangle window can be replaced by explicit multiplication with a window with more desirable frequency characteristics. The most widely used example of such a window, which will also be used for the computations in this work, is the Hann window [3].

Picket Fence Effect Since the frequency domain is sampled, information about very narrowband frequency peaks which are not covered by a sampling point is obscured. This effect can be viewed as looking at a scenery through a picket fence – hence the name [3]. The most obvious way to reduce the influence of the picket fence effect is to increase the frequency resolution by increasing the signal length N.

2.2.2 Short-Time Fourier Transform

While an untransformed signal carries the time information with highest possible time resolution, it does not contain any frequency information explicitly. Similarly, the FFT contains the frequency information of the signal in the highest possible frequency resolution, but for non-stationary signals, the information is averaged in time and therefore is not able to show any progression over time. A compromise between both domains used mainly for non-stationary signals is the Short-Time Fourier Transform (STFT). Instead of performing one FFT on the whole signal, the signal is split into overlapping segments in the time domain, which are then transformed individually [3]. However, there is a trade-off between time and frequency resolution since the frequency resolution is antiproportional to the window length. The edge cases of this trade-off are the time signal for window length 1 and the FFT for maximal window length. The maximum amount of possible overlapping between consecutive signals is determined by the window that is employed. The window decreases the signal energy at the beginning and end of a segment, thus the overlap restores this energy loss. For a Hann window, 50% overlapping between two consecutive signals is generally recommended as a trade-off between increased time resolution and invertibility [18].

2.2.3 Hilbert Transform and Envelope Demodulation

As introduced earlier, BPFO faults are usually characterized by impulse trains with BPFO frequency and exponentially decaying oscillation at the bearing resonance frequency [3]. This characteristic can be regarded as a multiplication between two signals:

$$x(t) = m(t) \cos \omega_b t \ , \tag{2.8}$$

where the oscillating part $\cos \omega_b t$ with bearing resonance frequency ω_b characterizes the oscillating response of the bearing, and $m(t)$ characterizes the impulse train and exponential decay. Equation (2.8) is a typical example of amplitude modulation, where a harmonic carrier signal, in this case $\cos \omega_b t$, is modulated by a modulating function or envelope, in this case $m(t)$. It is widely known that analysis of the signal envelope $m(t)$ can be significantly more fruitful than direct analysis of the original signal spectrum [3, 19, 20]. For this reason, the strategy of envelope demodulation by use of the Hilbert transform [21] will be explained briefly.

The Fourier transform of real signals, as introduced earlier, will generally have complex values, but to ensure realness of the inverse Fourier transform, the spectral values at a negative frequency will be the complex conjugate to its positive frequency counterpart [22]. In contrast, a signal whose frequency components are zero for all frequencies smaller than zero is called an analytic signal [23]. Consequently, an analytic signal has complex values in the time domain if it is not the trivial zero signal. For every real signal $x(t)$, there is a exactly one analytic signal $X(t)$ whose real part is $x(t)$ [24]. This analytic signal can be calculated from the spectrum of $x(t)$ by setting all negative frequency components to zero and then doubling the positive frequency part of the spectrum in amplitude. The inverse Fourier transform of the resulting one-sided spectrum is exactly the desired analytic signal $X(t) = x(t) + i\tilde{x}(t)$. The imaginary part $\tilde{x}(t)$ is called the Hilbert transform of the signal $x(t)$ [24]. Similar to Euler's formula, which states that a complex number can be split up into an amplitude and a phase instead of real and imaginary part, the analytic signal can also be divided into a real amplitude signal and a real phase signal. This amplitude of the complex signal is the envelope of the original signal and the described approach is used to calculate signal envelopes in application [3, 21].

This approach is qualitatively illustrated by a simple example in Figure 2.4. In Figure 2.4a, the initial signal as well as the desired envelope are illustrated. The corresponding spectrum is presented in Figure 2.4b. The modulation effects, resulting in a center peak at the carrier frequency with two peaks around the center peak at a distance of the envelope frequency, are visible. Also, it is clear that the negative frequencies are the complex conjugates of the positive frequencies. The spectrum of the corresponding analytic signal

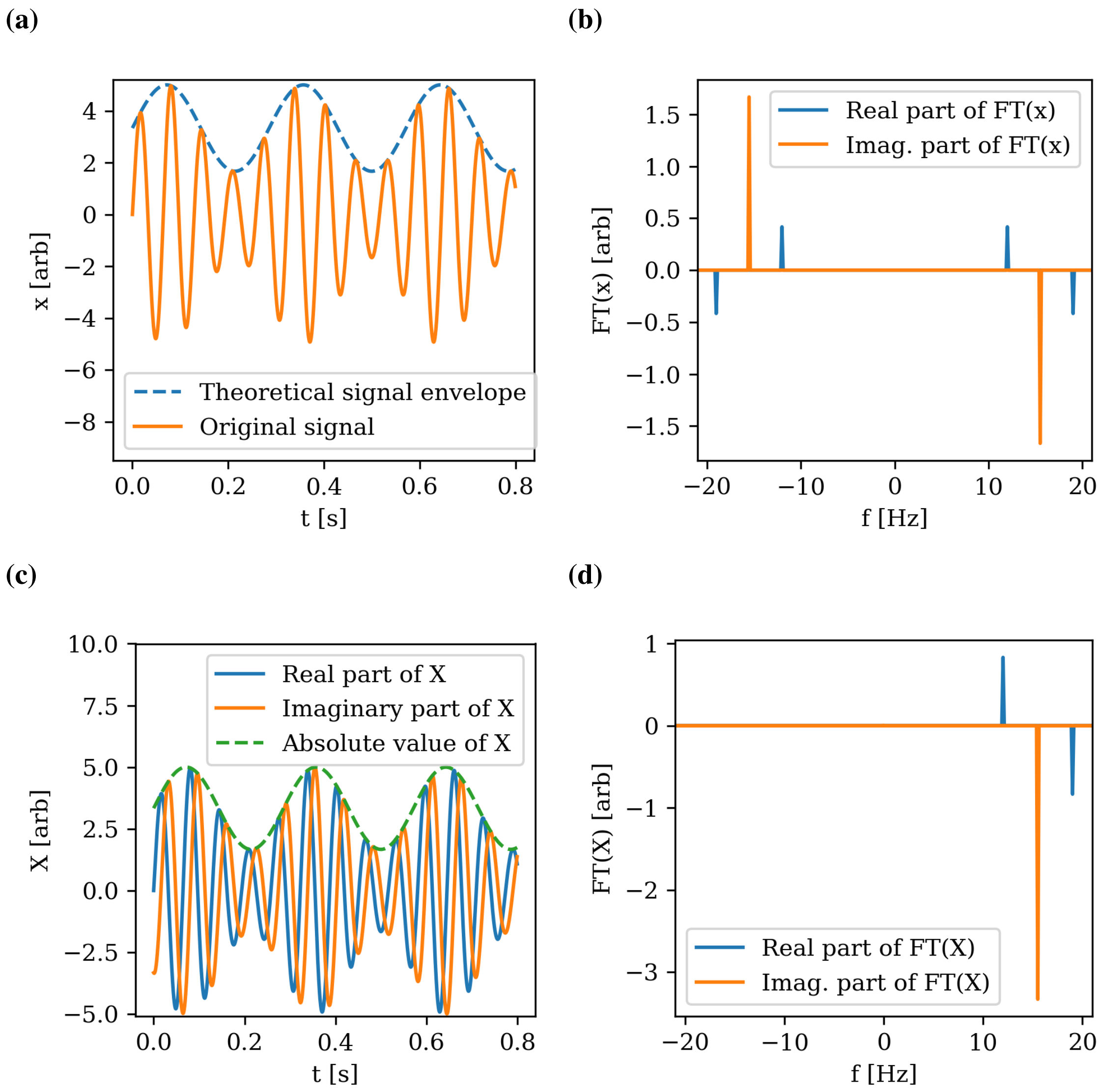

Figure 2.4: Hilbert Transform steps in time (left) and frequency (right) domain. **(a)** Step 1: original signal and expected envelope, **(b)** step 2: FFT of original signal, **(d)** step 3: One-sided spectrum created by removing the negative and doubling the positive frequencies, **(c)** step 4: IFFT of the one-sided spectrum.

is given in Figure 2.4d. The negative frequency parts are removed, while the positive frequency parts are doubled. Lastly, the IFFT of the analytic signal is shown in Figure 2.4c. This signal is complex. It is apparent that the absolute value of the analytic signal indeed traces the envelope of the original signal.

The relationship between analytic signal, original signal and Hilbert Transform signal is further visualized in Figure 2.5. However, signal spikes and random noise can easily

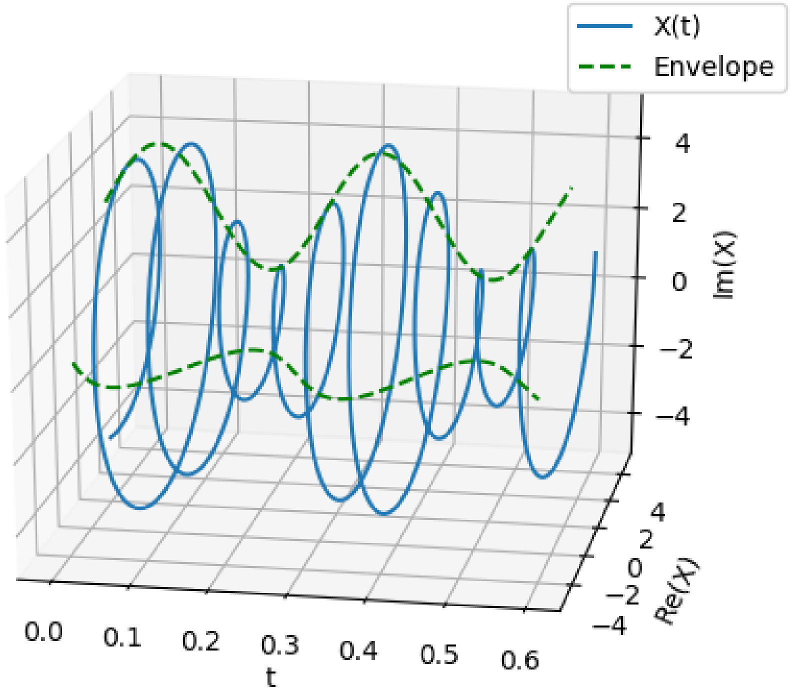

Figure 2.5: 3D visualization of Hilbert Transformation.

corrupt the presented basic signal envelope approach. For noisy signals, the envelope might easily be more complicated than the original signal itself [21]. The fault identification approach explained below therefore makes additional use of targeted bandpass-filtering of the signal.

2.3 Performance Measures of Classifiers

The approaches compared in the present work all perform a binary classification task – dividing signals into the classes ”healthy” and ”defected”. Especially with the recent rise of machine learning, there are many popular performance metrics for classifiers [25]. Among the most well-known metrics are confusion matrix, precision, recall and F1-score, all of which will be used throughout the present work to compare performances. These metrics are explained in this section.

The confusion matrix is a very comprehensive way to display classification results [26]. A classification test is usually performed by running the classifier on a test set, which contains signals from all desired classes. In the case for this work, the test sets must contain healthy and defected signals. The classification itself also divides the test set into two subsets: Positive, (i.e., flagged signals) and Negative (i.e., unflagged signals). In total, this gives four possible types. These types are usually displayed in a matrix form, which is called the confusion matrix. An exemplary confusion matrix is given in Table 2.3. All confusion matrices in this document follow this given layout.

Table 2.3: Exemplary confusion matrix.

	Defected signal	Healthy signal
Positive (Flagged as defected)	True Positive (TP)	False Positive (FP)
Negative (Flagged as healthy)	False Negative (FN)	True Negative (TN)

The precision P of a classifier is given by

$$P = \frac{TP}{TP + FP} \tag{2.9}$$

where TP and FP are true and false positives, respectively. In other words, the precision is the likelihood of a flagged signal being relevant with respect to the flag [26]. A high precision indicates that the number of FP is relatively low. If a signal is flagged by a classifier with high precision, then it is likely that this signal is defected.

The recall R of a classifier is given by

$$R = \frac{TP}{TP + FN} \ . \tag{2.10}$$

It indicates the percentage of all relevant signals that were found. A high recall indicates that the number of FN is relatively low. If a classifier has high recall, it is unlikely that a defected signal is not flagged.

Clearly in the optimal case, both precision and recall are sought to be high, or equivalently, the number of FN and FP is sought to be minimal. However, in reality, precision and recall are related antiproportionally [26]. This becomes clear in the edge cases. A classifier that blindly flags every signal it comes across will have a recall of 100%, since there cannot be any false negatives if there are no negatives. However, this classifier will have low precision since there will be many false positives if every healthy signal is flagged as well. Similarly, a classifier that has an extremely strict flag policy may obtain a precision of 100% at the expense of a high rate of FN, lowering recall values. This tradeoff makes clear that neither precision nor recall are suitable as sole measures of a good classifier. The harmonic mean of precision and recall is calculated by

$$F_1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{2.11}$$

and is called the F1-score of a classifier. Since the F1-score takes into account precision as well as recall, it can be a better suited metric than precision or recall alone [26].

CHAPTER 3
DETECTION APPROACHES

A basic detection approach for bearing faults is the time-domain approach. Since the overall machine vibration increases when a fault is present, statistical features such as mean, Root Mean Square (RMS), peak value, skew and kurtosis can indicate presence of a defect [27]. Kurtosis is used often since a high kurtosis value indicates high impulsivity of a signal [3]. The kurtosis is a scaled version of the fourth statistical moment and is computed by:

$$K = \frac{\int_{-\infty}^{\infty}[x-\mu]^3 p(x)dx}{\sigma^3} - 3 \, , \tag{3.1}$$

where μ is the mean and σ is the standard deviation of the signal. Advantages of these intuitive statistical approaches include their ease of implementation and the relatively low computational effort. However, they also have some significant disadvantages. Increase in vibration levels can be caused by various reasons, bearing defects being only one of them. With purely statistical methods, it is very difficult to find the actual part on an asset that needs to be replaced. Statistical features also have to be measured over time and compared to healthy values for specific bearings. This means that the healthy state of the bearing must be known and analysis of snapshots of various machines that are not connected is impossible.

Frequency-based detection approaches promise to address these issues. While such approaches come with the cost of higher computational effort, causes of particular defect can be determined more accurately due to distinct characteristics within the spectral data. As established above, bearing defects are characterized by periodic, with impulsive spikes in the waveform and similarly periodic peaks in the spectrum. This signature can be detected by multiple approaches, which will be presented below.

3.1 Harmonic Peak Finding

A straightforward fault detection approach is direct harmonic peak finding. This approach is developed directly with medium to low sampling rates in mind, where the eigen frequencies of the bearing are not represented. Since the bearing signal is periodic in nature for constant rotational speeds, a BPFO defect signature has frequency components at the BPFO frequency and its harmonics [3].

For a strong BPFO fault, these peaks can be assumed to be among the highest peaks in the spectrum. This condition is what the harmonic peak finding approach utilizes in determining faults. First, a fixed number N of highest amplitudes and their corresponding base frequencies in the spectrum are determined. For each frequency $f_n, n \in \{1, \ldots, N\}$ corresponding to one of these high peaks, the harmonic frequencies $h_{n,k} = kf_n, k \in \mathbb{N}$ are examined. Since a BPFO fault has a spectrum signature with harmonics, local extrema or peaks in the spectrum at frequencies $h_{n,k}$ are expected if f_n corresponds to a BPFO base frequency. However, due to the above-mentioned leakage and picket fence effects, the determined frequency f_n can be slightly shifted compared to the true BPFO frequency, which also means that the harmonics can be shifted from the exact integer multiples of f_n. Therefore, local harmonic extrema are searched in a range of $k \cdot 3\,\mathrm{Hz}$ around the expected peak location $h_{n,k}$. If one or multiple extrema are found, then the maximum of them is treated as possible BPFO harmonic. All bases f_n with at least 4 consecutive identified harmonics are treated as possible BPFO base signals.

This approach is illustrated in Figure 3.1 for an example spectrum and $N = 3$. The three leftmost peaks are the highest peaks in the spectrum and are therefore identified as possible base peaks. The blue, yellow and red lines indicate the identified peak locations of the projected harmonics for each of the three base peaks. In the example in Figure 3.1, the second, yellow peak would not be regarded as possible BPFO because only one consecutive

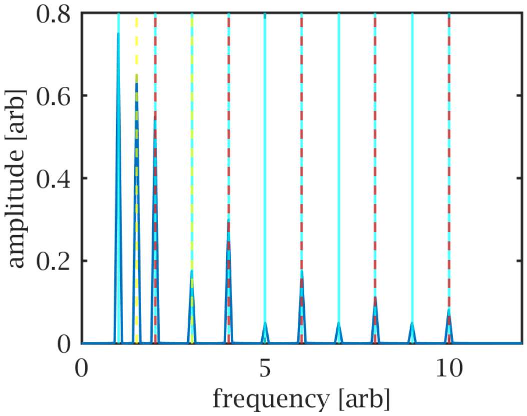

Figure 3.1: Example of harmonic peak identification.

harmonic can be found, whereas enough peak harmonics can be identified for the first (blue) and third (red) peak.

This approach so far does not need a lot of a priori information, however there are many false positive cases where a signature that does not correspond to BPFO is flagged. For instance, noise can lead to closely spaced local extrema, which can be misinterpreted as harmonics. Additionally, the first harmonic could be labeled as BPFO base, leading to misidentification of the BPFO frequency. There could also be ambiguous situations like the example in Figure 3.1, where multiple BPFO bases can be identified. The harmonic score and the harmonic certainty are introduced to resolve these problems.

Frequency spectrum peaks introduced by BPFO defects are expected to be distinct in the sense that the next higher amplitude value in the spectrum should correspond to a neighboring peak. This is the case for the red harmonic sequence in Figure 3.1, but not for the blue one. This property is utilized by the harmonic score and certainty. To compute the harmonic score and certainty, the distance $d_{n,k}$ between the considered peak $h_{n,k}$ and the closest frequency with a higher amplitude value is calculated via:

$$d_{n,k} = \min_{f:\ a(f)>a(h_{n,k})} \{|f - h_{n,k}|\} \ , \tag{3.2}$$

where f is a frequency in the considered frequency range and $a(f)$ describes the amplitude

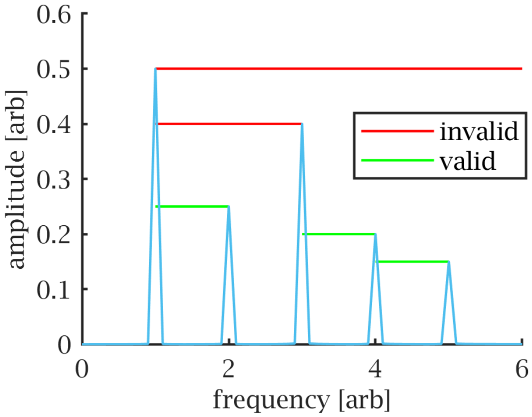

Figure 3.2: Example of harmonic score distance determination.

spectrum. An illustration for this distance value can be found in Figure 3.2. The length of the green or red lines corresponds to the harmonic distance. In this simple example, all green lines have a distance $d_{1,2} = d_{1,4} = d_{1,5} = 1$. The red lines correspond to larger distances $d_{1,3} = 2$ and $d_{1,1} = \infty$. If the closest higher point in the spectrum corresponds to a neighboring peak as discussed above, then the distance $d_{n,k}$ should be equal to the distance between to harmonics, i. e. the BPFO base frequency f_n. All peaks k where this is the case are denoted valid peaks, all others invalid peaks. The second, fourth and sixth peak marked green in Figure 3.2 are valid since their distance is 1.

The harmonic certainty HC_n is defined as the ratio of the number of valid peaks to all peaks. The harmonic score HS_n is the sum of the amplitudes of all valid peaks. They are calculated as follows:

$$HC_n = \frac{n_{\text{valid}}}{n_{\text{all}}} \tag{3.3}$$

$$HS_n = \sum_{k=1, d_{n,k}=f_n}^{N_n} a(h_{n,k}) \; . \tag{3.4}$$

For the illustrating example in Figure 3.2, the harmonic certainty would be $\frac{3}{5}$ and the harmonic score would be approximately 0.63. The harmonic certainty HC_n indicates how many of the harmonic peaks are prominent as expected. Its values range from 0 to 1, high values indicating a structure as expected, where the harmonic peaks are prominent. Low values indicate a signal where higher peaks between the harmonic peaks are present, indicating that the wrong BPFO base was chosen or that the signal is noisy. The harmonic score HS_n indicates the energy of the peak features and can therefore be used to find the most BPFO-like peak sequence if a signal has multiple candidates.

The whole procedure is summarized in the flowchart presented in Figure 3.3. For the BPFO defect identification approach, all candidates with a harmonic certainty of less than a specified certainty threshold are discarded. From all other candidates, the one with the highest harmonic score is picked and a defect is flagged. To further reduce the number of false positives, it can be useful to define the signal to noise ratio (SNR) as the ratio of all identified harmonic peaks and the rest of the frequency spectrum. If this SNR is lower than a certain threshold, then the signal is not flagged. A priori information about the expected noisiness of a signal can help determine an appropriate minimal SNR.

3.2 Kurtogram-Based Envelope Analysis

One basic assumption of the above mentioned peak finding approach is that the BPFO base frequency peak is among the highest values of the spectrum. However, for signals with high sampling rates, this is not necessarily the case. In Figure 3.4, the spectrum of the ideal BPFO signal from Figure 2.2 is presented. Due to the modulation effects explained previously, the highest peaks from the BPFO spectrum are in fact spaced around the natural resonance frequency of the bearing, and the base frequencies are low in comparison. For these systems, it is therefore unlikely that the basic peak finding approach presented above yields the desired results and pre-treatment of the signal and spectrum is necessary.

One particular approach of pre-treatment is targeted demodulation of the signal to reduce the modulation effects at high frequencies and therefore shift the frequency peaks to a lower range [3], where they can be detected by the peak finding algorithm as introduced earlier. This demodulation is carried out with help of the Hilbert transform introduced pre-

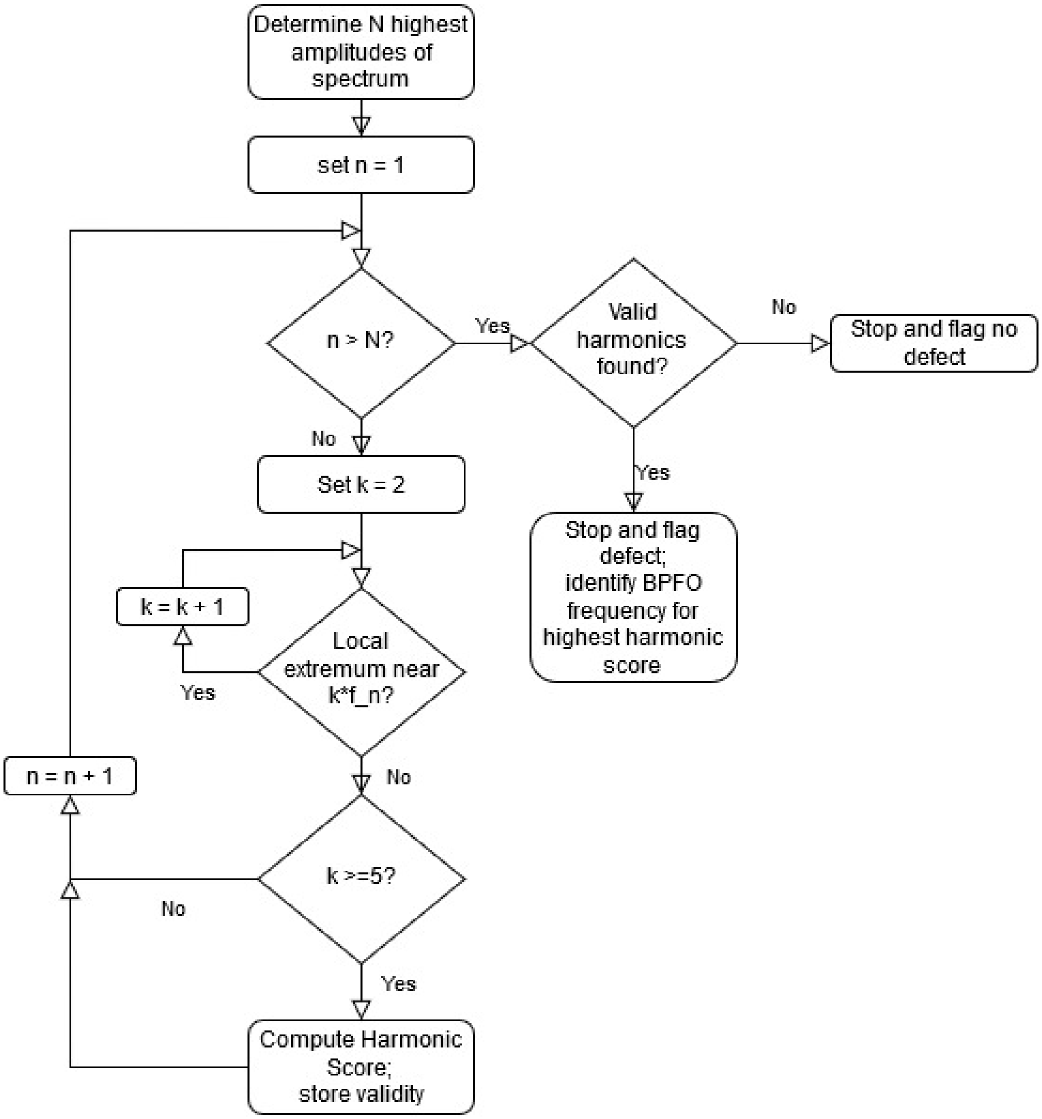
Determine N highest amplitudes of spectrum
set n = 1
n > N?
Yes
Valid harmonics found?
No
Stop and flag no defect
Yes
Stop and flag defect; identify BPFO frequency for highest harmonic score
No
Set k = 2
k = k + 1
Local extremum near k*f_n?
Yes
No
n = n + 1
k >=5?
No
Yes
Compute Harmonic Score; store validity

Figure 3.3: Flowchart for peak finding approach.

viously. For the ideal BPFO signal introduced in Figure 2.2, the resulting demodulated waveform and spectrum can be found in Figure 3.5. From Figure 3.5a, it is obvious that the demodulation is successful and the signal envelope traces the outline of the signal well. The spectrum in Figure 3.5b validates that the highest harmonics are indeed transported to lower frequency by the envelope demodulation. However, even small layers of noise can throw off the demodulation algorithm and result in a signal that is even more modulated and noisy than the original signal [21]. Therefore it is crucial to filter the original signal to remove most of the noise while leaving the BPFO signature intact, before applying envelope

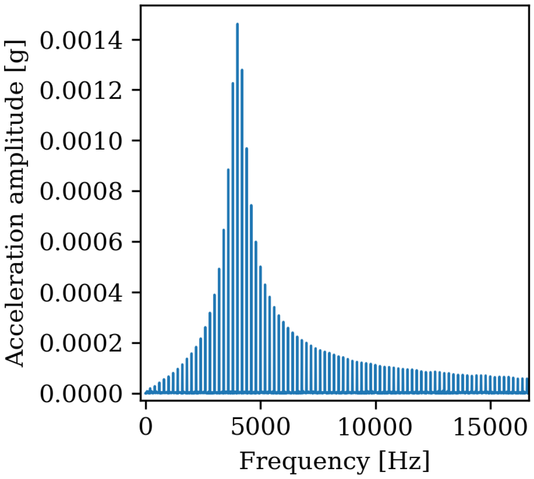

Figure 3.4: Spectrum of ideal bearing signature from Figure 2.2.

demodulation.

3.2.1 Spectral Kurtosis and Kurtogram

The kurtogram based on spectral kurtosis is a very valuable tool to determine the best bandpass filter specifications to keep impulsive signal components while eliminating most of the other noise [28]. The discrete spectral kurtosis is a property of a signal that is evaluated with help of the STFT that was introduced above. The usual way to look at the STFT is regarding it as a sequence of spectra for successive time instances. However, since the represented frequency values are constant over time, the STFT also can be regarded as a partition of the original signal into various frequency bins over time. This concept is illustrated in Figure 3.6.

These output time signals can be regarded as the output of quasi-analytic filters with the central frequency f and the bandwidth Δf, where Δf is the frequency resolution of the STFT. The discrete spectral kurtosis can be computed as the kurtosis of these signal partitions, depending on the frequency f. By the Bedrosian theorem [29], this estimator acts like a bandpass filter around the center frequency $f \pm \Delta f$ if the signal is sufficiently stationary.

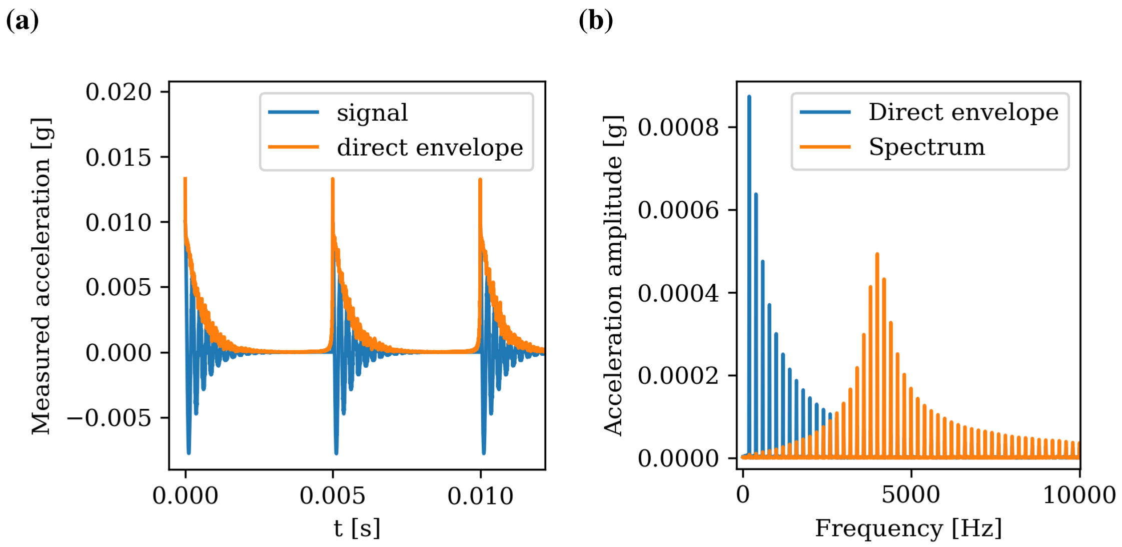

Figure 3.5: Ideal time signal from Figure 2.2 and computed envelope in **(a)** time and **(b)** frequency domain.

To optimally filter the signal components with high spectral kurtosis, it is necessary to perform the above-mentioned steps for various frequency resolutions. Antoni [30] proposes a cascading scheme to compute multiple levels of spectral kurtosis with frequency spacing multiples of 2 and 3, which utilizes the same methods as the FFT and has a complexity order of only $\mathcal{O}(N \log N)$. For implementation of the kurtogram, the interested reader is referred to [30, 31]. Now the total combined envelope demodulation and peak identification approach can be stated. First, the kurtogram of the signal is computed and the $f, \Delta f$ combination with the highest kurtosis value is selected. The Hilbert transform is then used to demodulate the filtered signal and afterwards the peak finding approach explained earlier is used to determine whether possible BPFO harmonics are present.

3.3 Time-Frequency Curve Extraction

When dealing with unknown rotational speeds, it is often assumed that the rotational speed may not be constant over time, since there is no way to ensure this. For this reason, literature for fault detection under unknown rotational speed often assumes non-constant shaft

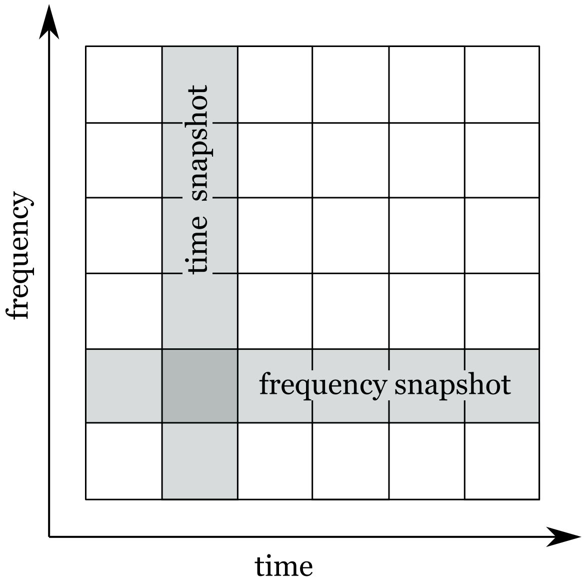

Figure 3.6: STFT interpretation possibilities as snapshots in time or frequency.

rotation speeds [32] and employ time-frequency methods such as the STFT or the wavelet transform [20]. Both previously mentioned approaches rely on the FFT and use detection of peaks in the spectrum. This approach can be flawed if peaks created by noise or other stochastic processes are misinterpreted as bearing fault harmonics. Time-frequency methods can also be of use here, since these processes can be transient and therefore not represented constantly in a time-frequency diagram. One approach that utilizes these benefits is TFCE. An overview of the approach introduced by Huang, Baddour, and Liang, 2018 will be given in this section; for more insight the interested reader is referred to [33, 34, 35].

In a first step, the Time-Frequency Representation (TFR) of the signal is obtained using the STFT. For each time instance, a number of peaks, i. e. local maxima in the frequency spectrum corresponding to the time instance, are determined. Afterwards, a fast path optimization algorithm [36] determines a fixed number of frequency peak curves throughout time. The schematic functionality of the fast path optimization is illustrated in Figure 3.7. It is important to note that peaks which have already been assigned to a curve are removed

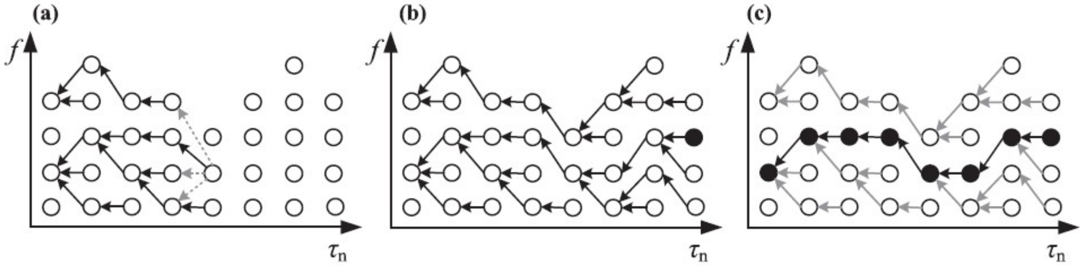

Figure 3.7: Illustration of path optimization algorithm [37]. **(a)** Recursive determination of predecessors, **(b)** identification of paths, **(c)** determination of best path.

from the TFR to reduce interference for the subsequent curves to be identified.

The crucial next step for bearing fault identification is employing bearing frequency characteristics introduced previously. As is apparent from Table 2.2, both for BPFO and for BPFI defects, the defect frequencies are proportional to the shaft rotation frequency. It is therefore reasonable to require that the quotient between potential bearing frequency and shaft frequency is constant throughout the time signal.

Huang, Baddour, and Liang additionally require that this quotient is approximately the fault frequency factor listed in Table 2.2. However, since one of the crucial assumptions of the present work is unknown bearing configuration, this step had to be omitted. If such a constant harmonic is found, then the signal is reported as defected. The approach leaves some tunable parameters to change the performance depending on the respective requirements. Those tunable parameters include the STFT window length and the overlap, the allowed ratio variance and the number of extracted curves. The choice of parameters used in this work is explained in Section 4.4.

CHAPTER 4

METHODOLOGY

4.1 Data

To investigate the classification performance of the bearing defect detection approaches, both simulated and experimental data were utilized. This section reviews characteristics of the simulated and experimental vibration data utilized.

4.1.1 Simulated Data Generation

To be able to generate a diverse test set for the different approaches, simulated data was considered. The simulated data consisted of BPFO, shaft vibration and noise components. Due to the linearity of the FFT in all considered approaches, independence of these three components was assumed and they can simply be superposed to result in one simulated machine signal. The first component is the perfect BPFO signature as introduced in Figure 2.2 and is given by:

$$s_{\mathrm{bpfo}}(t) = e^{-\lambda(t \bmod T)} \sin\left(2\pi\omega t\right) \ , \tag{4.1}$$

where λ is a damping coefficient for the introduced impulse, T is the reciprocal of the BPFO frequency introduced in Table 2.2 and ω is the resonance frequency of the bearing. The operator mod refers to the modulo operator.

For the purpose of this research, the bearing configuration of the bearing NTN NUP 2305E [38] was modeled. For this specific bearing, the BPFO frequency factor is 4.26, as in Table 2.2. An exponential decay factor of $\lambda = 2000$ and a bearing resonance frequency of $4\,\mathrm{kHz}$ was assumed. Caution is necessary when going from the analytical expression in Equation (4.1) to the discretely sampled signal necessary for the classifications. Identical to an analog signal measured by a vibration sensor, the direct sampling of the simulated signal in Equation (4.1) with a specific sampling frequency f_s is subject to aliasing effects. Signal components at higher frequencies than the Nyquist frequency of $\frac{f_s}{2}$ are mapped to their modulus with respect to the Nyquist frequency. This aliasing phenomenon was explained earlier in more detail. When dealing with measured signals, an analog anti-aliasing low-pass filter with a cutoff frequency lower than the Nyquist frequency is used to combat the aliasing effect. The most accurate equivalent method for the mathematical signal in Equation (4.1) would be analytically convolving the expression with an analytic expression of a suitable low-pass filter and the result of this computation would have to be sampled. However, this approach entails significant mathematical and computational effort.

A simplified approach involves identifying a high sampling rate such that signal components above the corresponding Nyquist frequency are small and their aliasing effects are negligible. After sampling at this high frequency, a digital low-pass filter emulating the analog anti-aliasing filter can be applied to the signal, which can then be downsampled to the original desired sampling rate. With bearing configurations as introduced before, the amplitudes of the spectrum begin to decrease after $4\,\mathrm{kHz}$ (c. f. Figure 3.4), and for frequencies larger than $40\,\mathrm{kHz}$, the spectrum contributions are negligible. For increased computational efficiency, it is advisable to choose a high frequency which is a common multiple of many desired sampling rates, and additionally larger than $40\,\mathrm{kHz}$. This base signal can then easily be filtered by a digital anti-aliasing filter and then downsampled to the desired sampling rate.

The second component for the simulated signal is the shaft rotation component. Due to light imbalance and/or misalignment, the first few harmonics of shaft rotation will almost always be visible in the spectrum [3]. For the simulated signals, the shaft rotation was modeled directly in the frequency domain, specifying the shaft rotation frequency and the first five harmonics with exponentially decaying amplitude. To vary the signal, the phases for these six summands were added as random values between 0 and 2π. The third component is Gaussian noise. Noise was added to the simulated data to represent influences of other machine parts such as gears or felt, and additional noise introduced in the measurement or sampling process.

All fixed parameters of the simulated base signals are summarized in Table 4.1. The three signal components were added to obtain a simulated vibration signal. The BPFO component was scaled such that its energy is one. A diverse test set of simulated signals was created by varying the energy of noise and shaft rotation components in relation to the BPFO energy. Subsequently, simulated signals as introduced in this work can be characterized by their shaft and noise energy relative to the BPFO energy and by their sampling rate. Due to the relative character of all classification approaches, scaling of the resulting signal does not impact the classification results. This way of generating simulated signals is therefore representative of a wide range of overall energies. For every combination of considered sampling rate, relative shaft energy and relative noise, a defected signal was created using the above-mentioned approach. Additionally, a corresponding healthy signal was created by only adding shaft rotation and noise, while omitting the BPFO component. This signal generation process is illustrated in Figures 4.1 and 4.2 for a relative shaft energy of 0.5 and a relative noise energy of 200. The signals can be characterized by their sampling rate, relative shaft energy and relative noise energy. For the sake of brevity, those three factors can be combined in set notation. For example, the above-mentioned signal with sampling rate $20.48\,\mathrm{kHz}$ is represented by the notation $\{20.48\,\mathrm{kHz}, 0.5, 200\}$.

To facilitate generation of unique test sets, signals were divided into noise categories of low, medium and high, as indicated in Table 4.2. Relative shaft energies between 0 and 2 were considered. Since the noise was distributed over a significantly wider frequency range than the shaft signal, the noise energy factors must be much higher than the shaft energy factors for the same effect.

Table 4.1: Fixed parameters for simulated base signals.

Base signal type	Parameter		Value
BPFO	BPFO frequency	$\frac{1}{T}$	128 Hz
	Exponential decay factor	λ	2000
	Bearing resonance frequency	ω	4 kHz
Shaft rotation	Shaft rotation frequency		30 Hz
	Harmonics exponential decay factor		-1
	Number of harmonics		6
Noise	Standard deviation (unscaled)		1

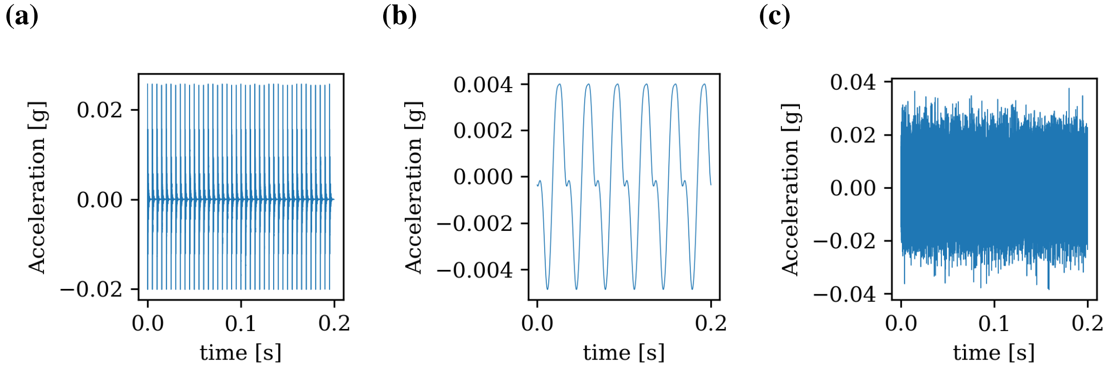

Figure 4.1: Simulated signal components for signal parameters $\{20.48\,\mathrm{kHz}, 0.5, 200\}$: **(a)** BPFO, **(b)** shaft and **(c)** noise.

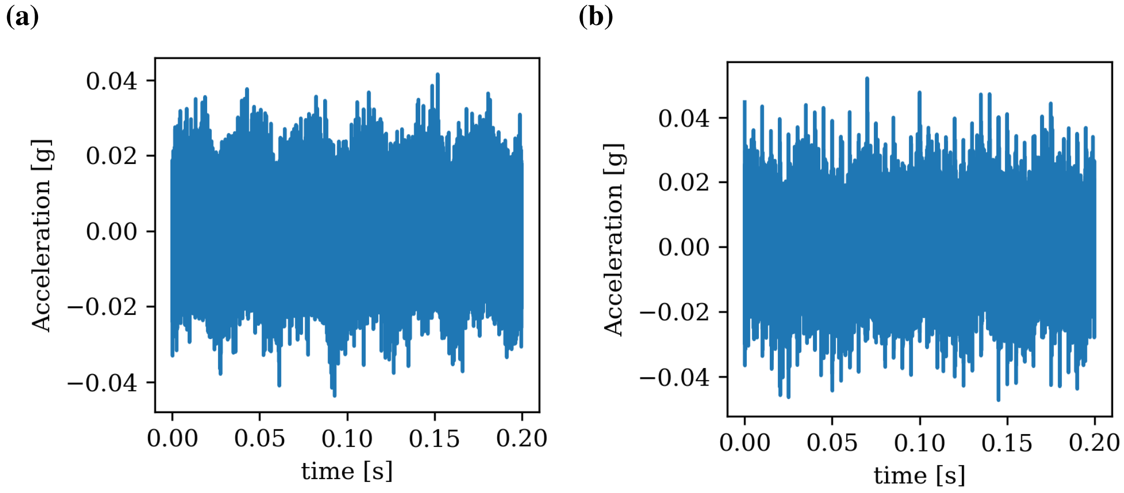

Figure 4.2: Simulated signals for signal parameters $\{20.48\,\mathrm{kHz}, 0.5, 20\}$ as sum of components: **(a)** healthy and **(b)** defected.

Table 4.2: Noise range limits for simulated data.

Category	Minimal relative energy	Maximal relative energy
Low noise	0	230
Medium noise	300	615
High noise	692	1000

4.1.2 Experimental Data

To further compare the bearing defect detection approaches, experimental data [39] was also considered. Run-to-failure tests performed under normal conditions on a test rig were considered. The data collection is explained in [19]. The data was collected during an endurance experiment at a dedicated test rig at the University of Cincinnati. A bearing test rig with four bearings on one shaft was considered. A schematic drawing of the test rig can be found in Figure 4.3. The shaft was coupled to an AC motor by a rub belt, driving the system with $2000\,\mathrm{RPM}$, i. e. approximately $33\,\mathrm{Hz}$. The shaft was loaded by 6000 lbs. with help of a spring mechanism. The four bearings used were Rexnord ZA-2115 bearings with a BPFO frequency of $236\ \mathrm{Hz}$ at the rotation speed of $33\,\mathrm{Hz}$. For clarity, all important numbers are summarized in Table 4.3. Each bearing housing had a PCB 353B33 high sensitivity quartz ICP accelerometer attached to measure the vibration.

Table 4.3: Important parameters for IMS experimental data [39, 40].

Shaft rotation frequency	$33\,\mathrm{Hz}$
BPFO frequency	$236\,\mathrm{Hz}$
Sampling rate	$20\,\mathrm{kHz}$

According to Qiu, Lee, Lin, and Yu [19] and the reference document of the experiment [39], the sampling rate was $20\,\mathrm{kHz}$ with a signal length of 20480 samples. However, as Liu and Gryllias [40] point out, the time vector indicates that the sampling rate likely is $20.48\,\mathrm{kHz}$. Since bearing configuration and rotation speed are assumed unknown in the present work, the dispute in these sampling rates had no impact on performance. Therefore

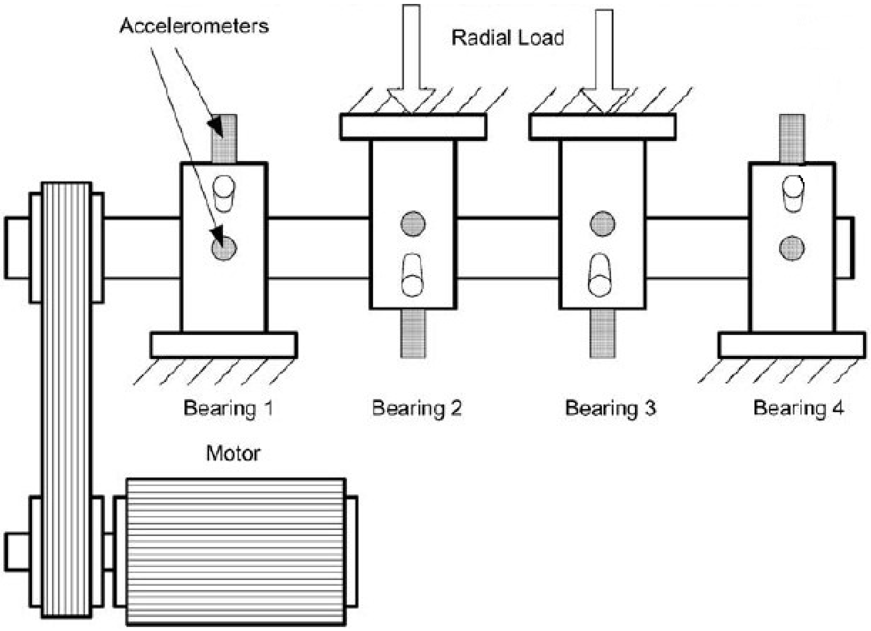

Figure 4.3: Test rig for run-to-failure experiment [19].

it is not problematic to use the originally indicated sampling rate of $20\,\mathrm{kHz}$. Three test runs are represented in the dataset [39]. Acceleration measurements were taken every 10 minutes. While the test rig was running, a magnet plug collected debris from the oil, indicating bearing degradation. When the amount of debris on the plug exceeded a certain level and bearing faults were inevitable, the rig was turned off and the bearings examined.

The second test was carried out for approximately seven days. At the end of the test, an outer race fault in Bearing 1 was found [39]. This test is used in the present work. A dataset of healthy signals was curated from measurements 100-199, and a defected dataset was curated from measurements 840-939. According to the survey given by Liu and Gryllias, most approaches start detecting the fault in Bearing 1 between measurement 500 and 600, considering RPM, bearing configuration and the signal progression [40]. Thus, it is justified to take measurements 840-939 as defected bearings.

4.2 Assumptions

There are important assumptions and restrictions to consider for the experiments of this research. The fixed signal parameters can be found in Tables 4.1 and 4.3 for the simulated and experimental data, respectively. For comparability, the simulated data was designed in a way to mirror the experimental data directly, using the similar shaft rotation frequency. However, in practice, other combinations of shaft rotation frequency, BPFO frequency and bearing resonance frequency will likely be the case. A change in these frequencies will affect the exact behaviors of the classification algorithms with respect to sampling rate. These effects and how to apply the results of this work to the input signals with other frequency ranges will be discussed in Section 5.3.3. However, since none of the approaches use absolute amplitude values in any detection step, the present approach is robust towards vibration signals with various amplitudes. Another important assumption of the present work is that machinery faults other than BPFO are not considered in the data. The approaches are analyzed in their ability to distinguish BPFO signals from healthy signals. In real application, signals like gear noise from other machine parts can interfere with the detection [41].

4.3 Design of Experiment

As the main goal of the present work is to analyze the behavior of various approaches under low sampling rate conditions, classification experiments were conducted for various sampling rates ranging from $2\,\mathrm{kHz}$ to the standard literature value of $20\,\mathrm{kHz}$.

Table 4.4 summarizes all sampling rate steps as well as the relative shaft energy and noise energy ranges used for experiments. Sampling rates lower than the shown values of $2\,\mathrm{kHz}$ are not suitable for the task since a minimal number of bearing frequency harmonics must be present in the spectrum for any of the approaches. The simulated data for the experiment was designed by full factorial design, i. e. all possible combinations of sampling rates and relative shaft and noise energies introduced in Table 4.4 were combined. For the

Table 4.4: Design of Experiment.

		Simulated data		
Parameter	Qualitative range	Numerical range	Step size	Experimental data
Sampling rate	very low	–		2 kHz
	low 1	4.096 kHz		4 kHz
	low 2	6.144 kHz		5 kHz
	medium	10.24 kHz		10 kHz
	high	20.48 kHz		20 kHz
Rel. shaft energy	–	0 – 2	0.142	0.7 – 15.4
Rel. noise energy	low	0 – 230	76.9	–
	medium	300 – 615	76.9	0.08 – 129
	high	692 – 1000	76.9	–

experimental data, this is not the case. Relative noise and shaft energy ranges were determined by computing and comparing the relative noise and shaft energy of the experimental signals.

The shaft energy of a signal was approximated by the sum of the squared amplitudes of the first six multiples of the shaft frequency 30 Hz. Similarly, the BPFO energy was approximated by the sum of the squares of the amplitudes at all multiples of the BPFO frequency. Every amplitude value not contributing to either shaft or BPFO energy contributed in the same manner to the noise energy. To obtain accurate relative energies, the maximal and minimal relative shaft and noise energy were evaluated for all signals later than the 700th signal, where the BPFO signature was already present [40].

As the experimental data was from a run-to-failure experiment, the progression between healthy and defected signals is continuous. However, to be able to validate the classification algorithms in terms of their scores, it is important to have a validation set, whose ground truth is clear. According to Liu and Gryllias [40] and references therein, the defect can be detected between samples 500 and 600, and therefore the samples 100–199 and 840–939 are good indicators for certainly healthy and certainly defected signals, respectively.

4.4 Parameter Choices

All three approaches employ some design parameters that can be tuned by the user. All chosen design parameters are summarized in Table 4.5. These parameters proved optimal through a series of test runs in obvious simulated cases. In the peak finding approach, the number of possible base peaks is a trade-off between computational efficiency and likelihood to find the true BPFO base peak among the highest values. Similarly, the minimal number of harmonics and minimal harmonic certainty are fixed in a way to eliminate peaks caused by noise, while keeping BPFO signatures who are overlaid by noise. For the envelope analysis method, a maximal kurtogram step of 7 proves sufficient exactness. From the examples that will be discussed in detail below, it is clear that higher step sizes would not serve for clearer results. An FIR filter of order 20 provides a good frequency response for many standard applications.

Table 4.5: Fixed design parameters for the three approaches.

Approach	Parameter	Value
Peak finding	Possible base peaks	23
	Minimal harmonic certainty	0.1
	Minimal number of harmonics	4
Envelope analysis	Max. kurtogram step	7
	Order of bandpass filter	20 (FIR)
TFCE	min. STFT frequency resolution	5 Hz
	Number extracted curves	4
	Maximal variance for BPFO ratio	0.001

The minimal STFT resolution of 5 Hz is necessary to provide sufficient exactness and to reduce the amount of smearing and leakage in the frequency domain of the STFT. The STFT window length yielding a frequency resolution of 5 Hz is given by:

$$l_w = \frac{5\,\mathrm{Hz}}{f_s}\,, \tag{4.2}$$

where f_s is the sampling rate of the signal. For increased computational efficiency, the power of 2 closest to and higher than l_w is chosen as applied window length with a guaranteed frequency resolution of less than $5\,\mathrm{Hz}$. While the chosen number of extracted values 4 is recommended by [33], the maximal variance was significantly reduced to 0.001 after simulative studies. The additional step of comparison to the expected BPFO to shaft rotation can not be performed if the bearing configuration is not known, demanding a low variance to reduce false positives.

CHAPTER 5
RESULTS AND DISCUSSION

This chapter illustrates results obtained by applying the three approaches to the experimental and simulated signals introduced previously. First, the functionality of the three approaches is illustrated step by step using exemplary signals. Subsequently, direct classification results and confusion matrices are given to facilitate a first comparison between the three approaches. From the results of this section, it will become clear that the different approaches exhibit various restrictions and misclassifications for certain parameter values. As a first discussion, the next section uses those results to explain reasons for those limitations. Next, when methodic strengths and weaknesses of the approaches are clear, the influence of the parameters noise, shaft energy and sampling rate is studied and discussed. Finally, in the last section of this chapter, all insights from the previous sections are synthesized to come up with suitable recommendations for usage of the three approaches.

5.1 Classification Behavior

To validate the three approaches, various steps in their execution are illustrated graphically in the subsequent section. As explained previously, the direct peak finding approach utilizes the first harmonics of the BPFO frequency in the signal. To illustrate how the approach locates defect-related peaks, results are generated for a simulated signal consisting of a bearing defect. For this purpose, a simulated BPFO signal with a relative shaft energy of 1.5 and a relative noise energy of 205 with a sampling rate of $4.096\,\mathrm{kHz}$ and its spectrum were utilized. This simulated signal is displayed in Figure 5.1. In the figure, the BPFO harmonics are visible in the spectrum and their amplitude is approximately constant due to the low sampling rate and the anti-aliasing methods used.

The first step for the direct BPFO detection approach is determining the N highest

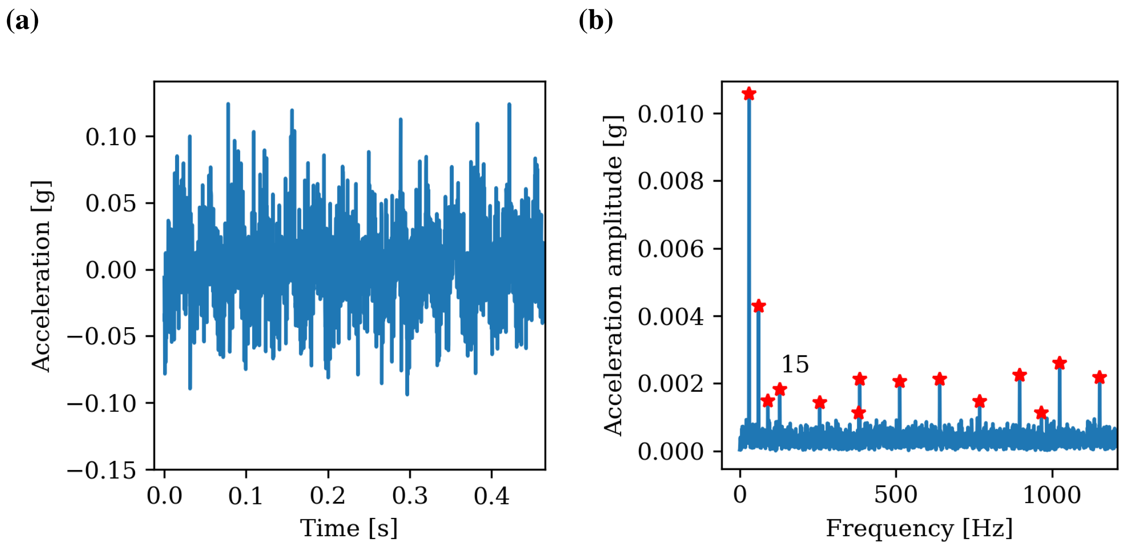

Figure 5.1: Exemplary simulated signal with defect and parameters $\{4.096\,\mathrm{kHz}, 1.5, 205\}$: **(a)** waveform and **(b)** spectrum. Identified high peaks indicated by red stars; true BPFO peak and rank indicated by Number 15.

points in the spectrum. They are indicated by red stars in the exemplary spectrum of Figure 5.1b. These peaks will be treated as potential BPFO base peaks for the following steps of the peak finding algorithm. The 15-th highest peak in the spectrum with an amplitude of almost $0.002\,\mathrm{g}$ at a frequency of $128\,\mathrm{Hz}$, which is indicated in Figure 5.1b, corresponds to the true BPFO base frequency. Next, the possible harmonics in the spectrum are determined for each of the possible base peaks indicated in Figure 5.1. If local extrema are present in a range around the predicted location of the harmonic, the highest of these local extrema would be assumed to be the harmonic peak, regardless of its absolute height. This is visualized in Figure 5.2 for the highest, the 12th highest and the 15th highest (i. e., the true BPFO) base peak, respectively. Since the highest base peak, which corresponds to the shaft rotation frequency, is located at a relatively low frequency of $30\,\mathrm{Hz}$ and there is some noise present, many small local extrema in the noise are misidentified as possible harmonic peaks, as obvious from Figure 5.2. In the extreme case of the highest base peak, depicted in Figure 5.2a, this leads to 69 identified possible harmonics in total. For the other cases, the number of identified possible harmonics is much lower with three in Figure 5.2b for the

12th highest base peak and 15 in Figure 5.2c for the true BPFO peak, respectively.

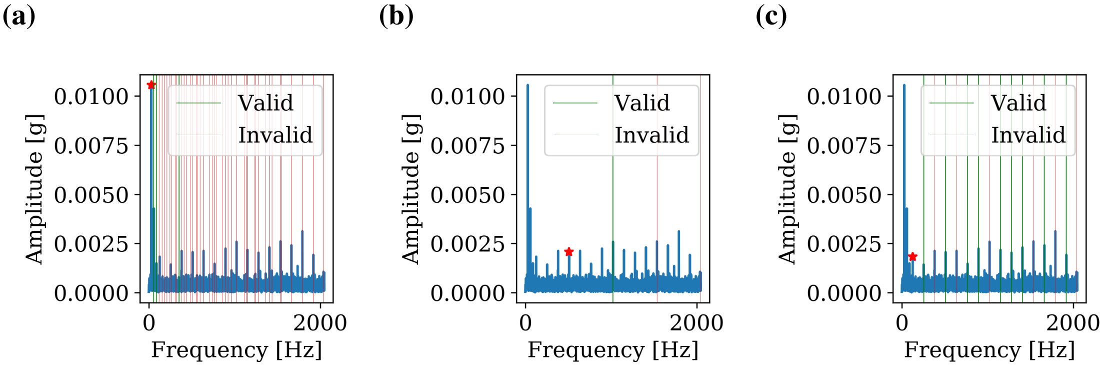

Figure 5.2: Predicted harmonics for defected signal with parameters $\{4.096\,\mathrm{kHz}, 1.5, 205\}$: **(a)** highest base peak, **(b)** 12th highest base peak and **(c)** real BPFO base peak. Red stars indicate base peak location. Red and green lines indicate predicted harmonics.

To identify the correct BPFO base peak from the identified candidate base peaks, the harmonic score and harmonic certainty of the associated harmonics are considered. For the true BPFO base peak in Figure 5.2c, 9 harmonic peaks, i. e. three fifths of all predicted harmonic peaks, are valid. In contrast, for the highest and 12th highest base peak, only a small fraction of harmonic peaks (three in 69 and one in three, respectively) are valid. The progression of the harmonic score over all identified candidate base peaks is shown in Figure 5.3. From this graph, it is apparent that there may be other candidate base peaks with more identified (valid or invalid) harmonics, such as the peak with 69 harmonics in this example, but the corresponding harmonic score of 0.0175 for the true BPFO base peak is significantly larger than for any other candidate base peak. This validates the utilization of the harmonic score to identify the correct BPFO frequency.

The peak finding approach was also verified by consideration of the experimental dataset [39]. The waveform and spectrum of the 847th measurement, taken 2004/02/18 at 07:42:00 and decimated as explained above to a sampling rate of $10\,\mathrm{kHz}$, is given in Figure 5.4. This measurement was taken late enough so that the BPFO fault was certainly present. The base peak corresponding to the BPFO frequency at $236\,\mathrm{Hz}$ is indicated by a star in Figure 5.4b. In contrast to the simulated data in Figure 5.1, it is apparent that the noise was less evenly

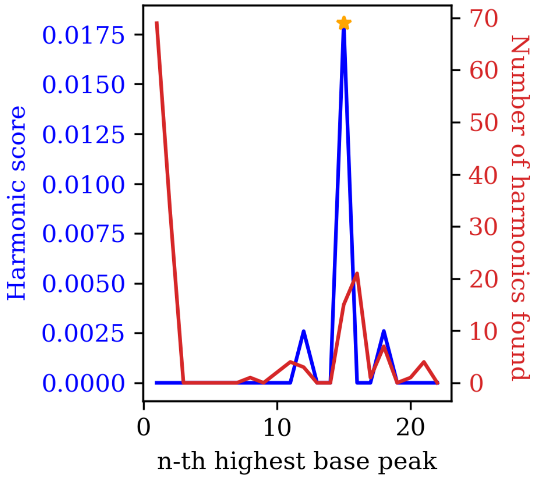

Figure 5.3: Harmonic score and harmonic peaks identified for all potential base peaks. True BPFO peak marked by star.

distributed across frequencies. In particular, there was a large peak with an amplitude of more than $0.03\,\mathrm{g}$ at approximately $1000\,\mathrm{Hz}$, which was neither caused by shaft rotation nor by BPFO. Additionally, from the figure, the damping introduced by the anti-aliasing filter is visible for frequencies greater than $4000\,\mathrm{Hz}$.

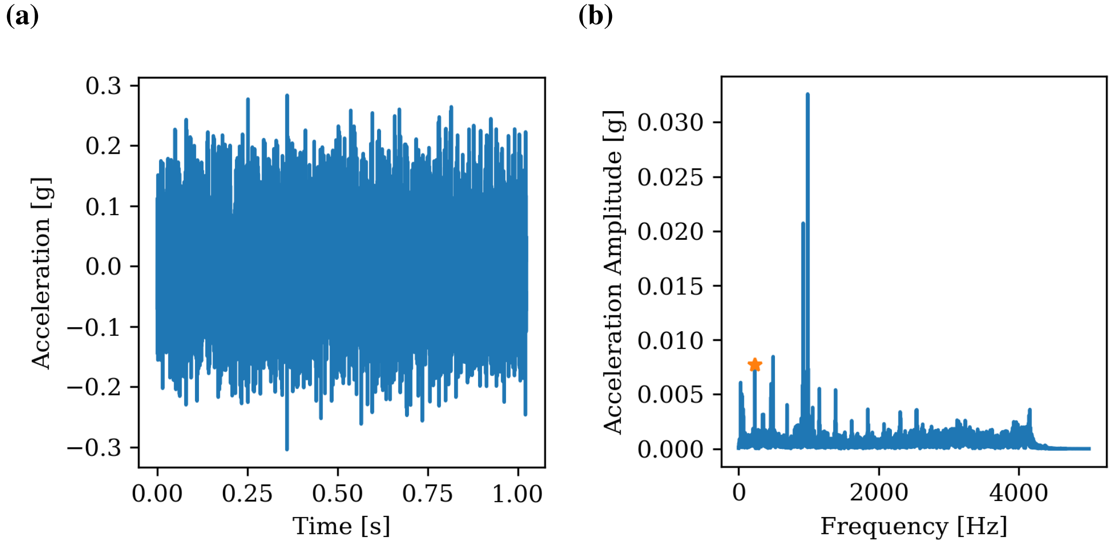

Figure 5.4: **(a)** waveform and **(b)** spectrum for 847th measurement of experimental data [39]. Star indicates true BPFO base peak.

As in the case for the simulated signals, harmonics were detected for many possible candidate base peaks, including the true BPFO base peak. Exemplary harmonic detection results for the highest candidate base peak and the true BPFO base peak are illustrated in Figure 5.5. The three harmonics predicted for the highest candidate base peak presented in Figure 5.5a are not visibly distinct in the spectrum, therefore all of them are invalid. In contrast, for the true BPFO base peak presented in Figure 5.5b, four out of 17 harmonics are classified as valid. The higher-order harmonics are not classified as valid since they are buried by noise, and even though the lower-order harmonics are distinct in the spectrum, some are not classified as valid because the high candidate base peak at approximately $1000\,\mathrm{Hz}$ is interfering.

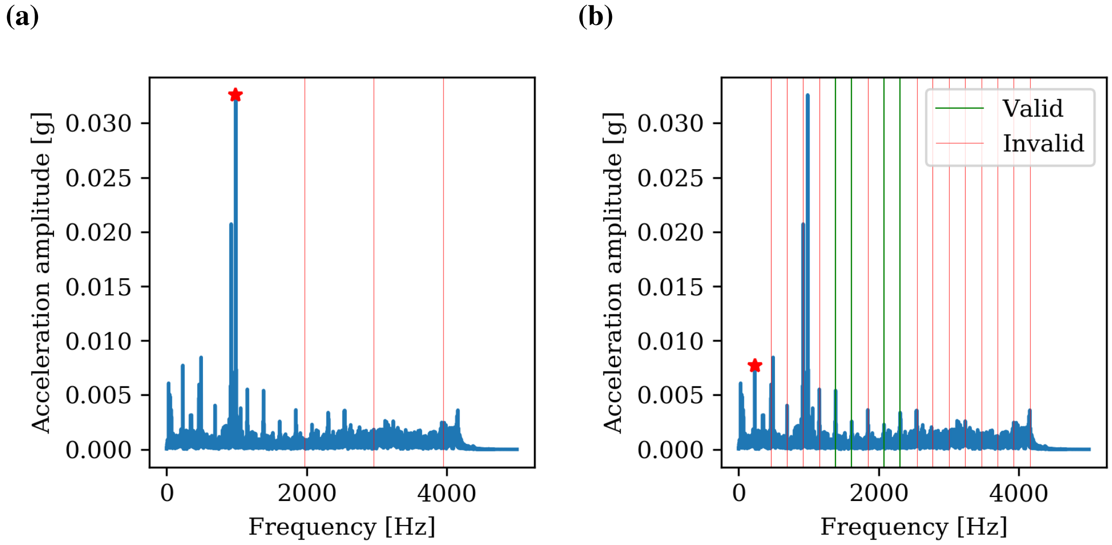

Figure 5.5: Predicted harmonics in spectrum of 847th signal of experimental data [39]: **(a)** highest base peak and **(b)** true BPFO base peak. Star indicates base peak location. Predicted harmonics indicated by red and green lines.

Across all identified possible base peaks, the true BPFO base peak exhibits the largest harmonic score of approximately 0.0138, as is apparent from Figure 5.6. Again, the BPFO base peak was not the base peak for which the most total harmonics are found, however of the many harmonics found for the 7th or 12th highest base peak (150 and 100, respectively), less than 10% each were found to be valid. From these exemplar data, it is clear that the direct peak finding approach using the harmonic score and harmonic certainty metrics is appropriate for identifying BPFO defects both in simulated and experimental [39] data.

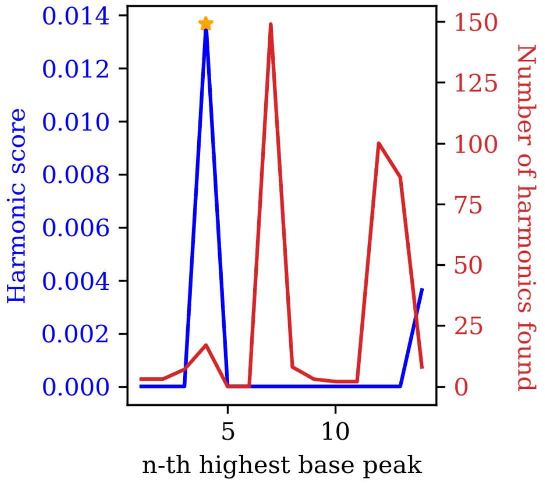

Figure 5.6: Harmonic score and harmonic peaks identified for all potential base peaks in experimental data [39]. True BPFO peak indicated by star.

As mentioned above, the envelope approach is a pre-treatment for the peak finding approach to enhance the bearing fault harmonics and bring them to lower frequencies in the spectrum. The first step is computation of the kurtogram [28]. To illustrate the behavior of the envelope demodulation approach, simulated and experimental signals with noise were utilized. A simulated signal with similar specifications as previous was utilized, i. e. a relative shaft energy of 1.5 and a relative noise energy of 300, but with a high sampling rate of $20.48\,\mathrm{kHz}$. The waveform of this signal can be found in Figure 5.7a. The kurtogram in Figure 5.7b shows the frequency band with the maximum kurtosis of approximately 3.2 at level 2.6 with a center frequency of $4.3\,\mathrm{kHz}$ and a bandwidth of $1.7\,\mathrm{kHz}$.

In Figure 5.8, the waveforms and spectra of the three pre-treatment steps for the considered simulated signal are compared. The bandpass filter filters out all signal portions corresponding to the shaft rotation and most of the noise, while keeping the relevant BPFO signal components intact, as apparent in Figure 5.8b. This is also apparent in the waveform presented in Figure 5.8a. The slow oscillations caused by the shaft rotation component are not visible in the filtered or the envelope signal. From the figure, the envelope demodula-

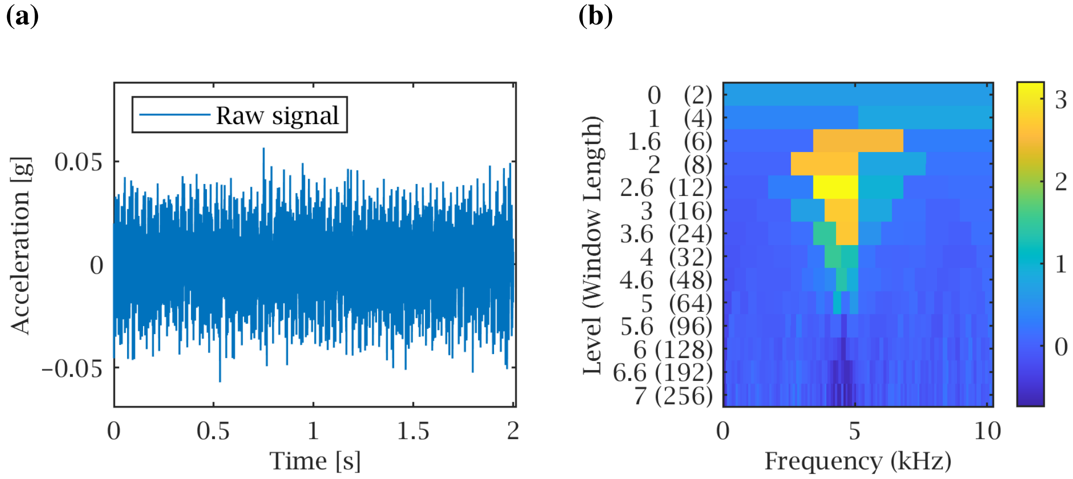

Figure 5.7: **(a)** waveform and **(b)** kurtogram for simulated signal with defect and parameters $\{20.48\,\mathrm{kHz}, 1.5, 300\}$.

tion does well tracing the outline of the filtered signal, indeed shifting the frequency peaks to the low frequencies in the spectrum where they can be picked up by the peak finding algorithm. It is important to note that, since the envelope traces an absolute value, its mean larger than zero. This corresponds to a high peak of $0.005\,\mathrm{g}$ in the spectrum at the frequency zero, which is not to be misinterpreted as a possible BPFO base peak. This validates the envelope approach as introduced in [3] and [28].

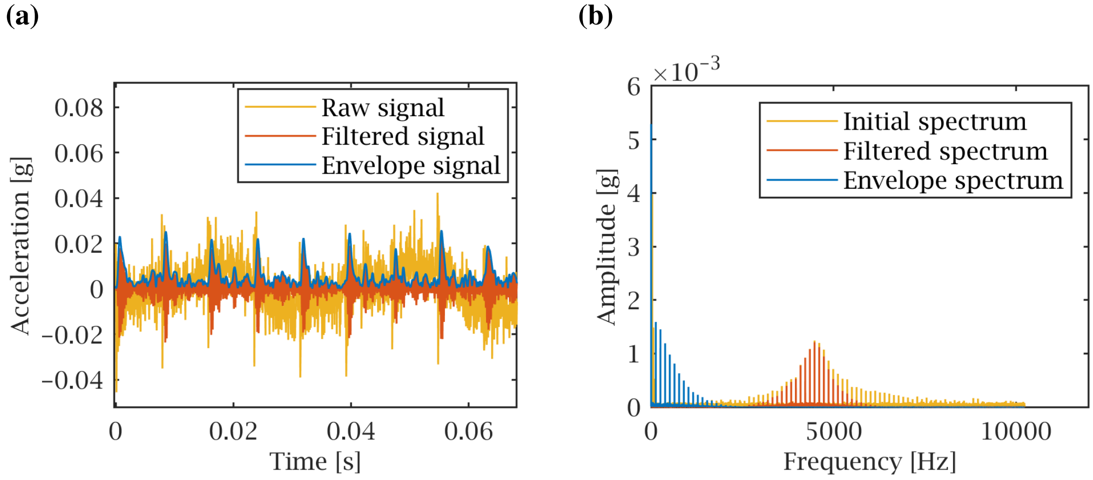

Figure 5.8: **(a)** waveforms and **(b)** spectra for the three different stages of the envelope demodulation algorithm in simulated signal with defect and parameters $\{20.48\,\mathrm{kHz}, 1.5, 300\}$.

The experimental signals further validate the envelope demodulation approach. Again, the 847th signal of the IMS bearing data [39] is analyzed, however for validation of the envelope demodulation approach, it is not downsampled. Similar to the simulated signal, the highest kurtosis value of 2.9 is found in a rather low level of the kurtogram depicted in Figure 5.9b. According to the kurtogram, the optimal bandwith is 3.3 kHz at a center frequency of 8.3 kHz. Figure 5.10 shows the raw signal, the filtered signal and the demodulated signal in the time and frequency domain. The bandpass filter retains the highest frequency band in the spectrum, eliminating the high peaks at 1000 Hz and 2000 Hz completely. Although the remaining filtered spectrum looks very noisy and no peaks are identifiable, envelope demodulation was successful at isolating the BPFO peaks and transporting them to the front of the spectrum. Further, these peaks are indeed spaced corresponding to distances of 236 Hz.

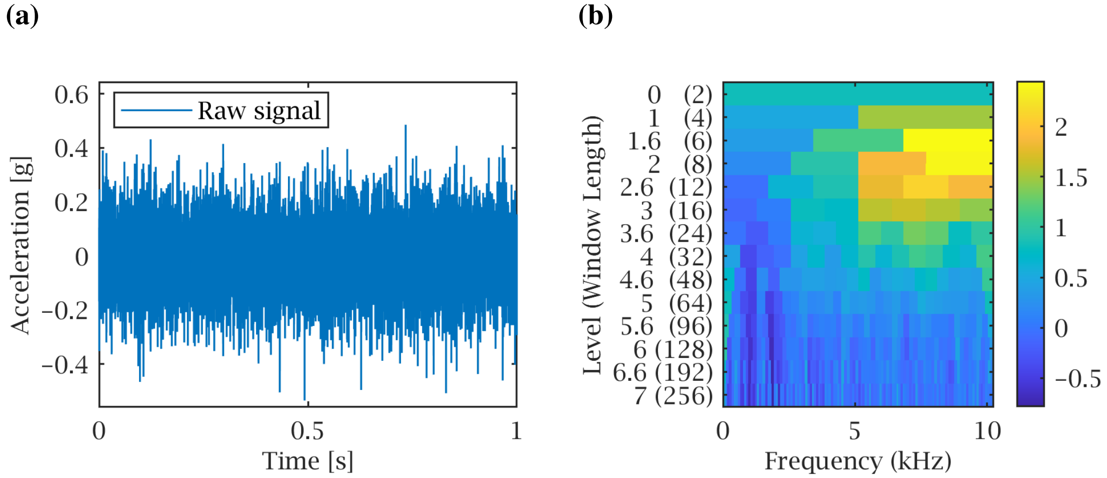

Figure 5.9: **(a)** waveform and **(b)** kurtogram for 847th experimental signal [39].

The TFCE approach employs a STFT representation of the signal. Once again, a simulated signal with a relative shaft energy of 1.5, a relative noise energy of 305 and a sampling rate of 20.48 kHz was used for approach validation. The waveform and spectrum of the simulated signal can be found in Figure 5.11. Again, in the spectrum in Figure 5.11b, the highest peaks are spaced around the resonance frequency of the bearing at 4000 Hz. In contrast, in the STFT presented in Figure 5.12a, only the frequencies up to 500 Hz are represented. The shaft rotation frequency at 30 Hz as well as the first three BPFO harmonics at

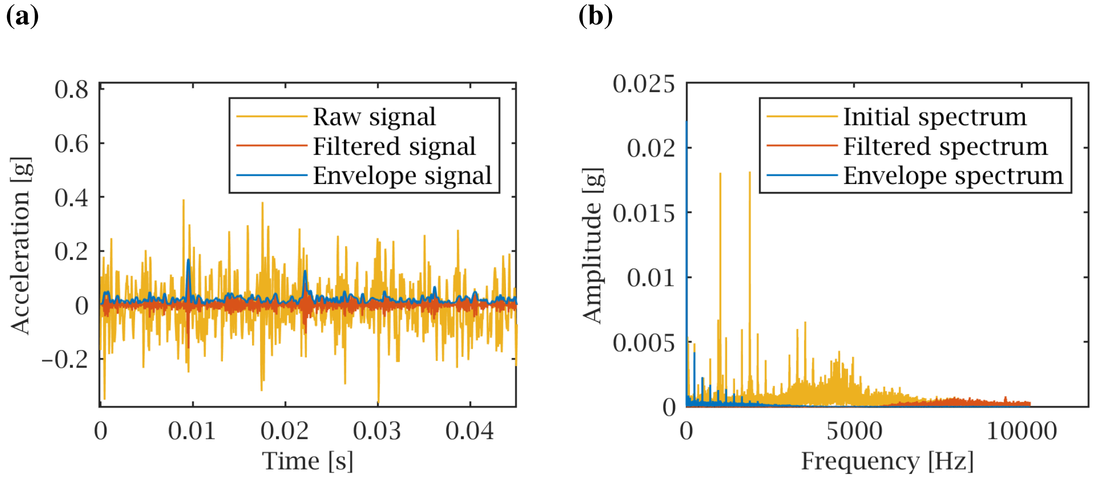

Figure 5.10: **(a)** waveforms and **(b)** spectra for the three different stages of the envelope demodulation algorithm in 847th experimental signal [39].

128, 256 and $384\,\mathrm{Hz}$ are clearly visible as horizontal red lines in the STFT, indicating constantly high amplitude values at these frequencies throughout time. The curve extraction algorithm also extracts these exact horizontal lines as curves, as visible in Figure 5.12b. The purple curve at $30\,\mathrm{Hz}$ corresponds to the shaft rotation and the blue, orange and yellow curves correspond to the BPFO frequency and the first two harmonics. Since these curves are very horizontal, the variance of the ratios between the lowest (orange) and second lowest (blue) curve values throughout time is smaller than 0.001, flagging this signal as defected.

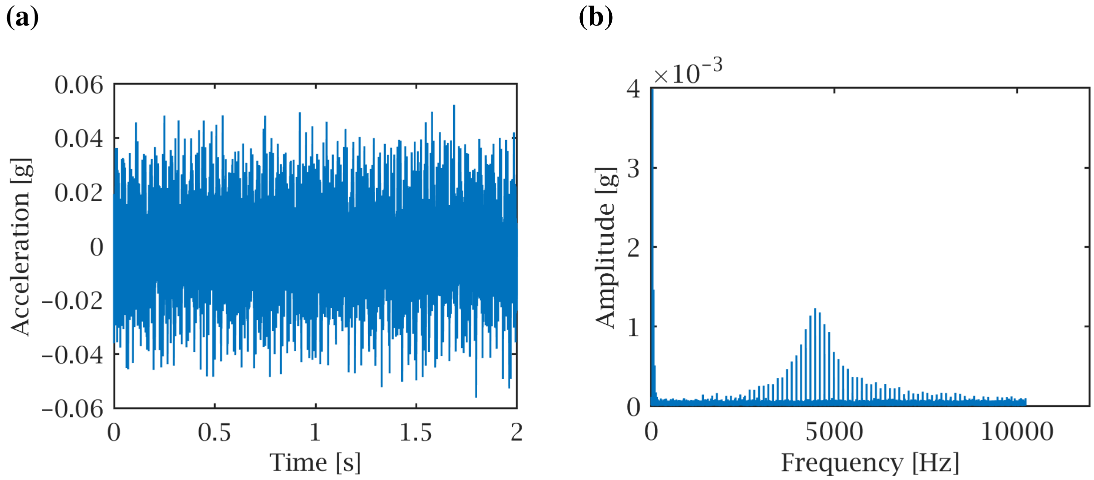

Figure 5.11: **(a)** waveform and **(b)** spectrum of simulated defected signal with parameters $\{20.48\,\mathrm{kHz}, 1.5, 305\}$.

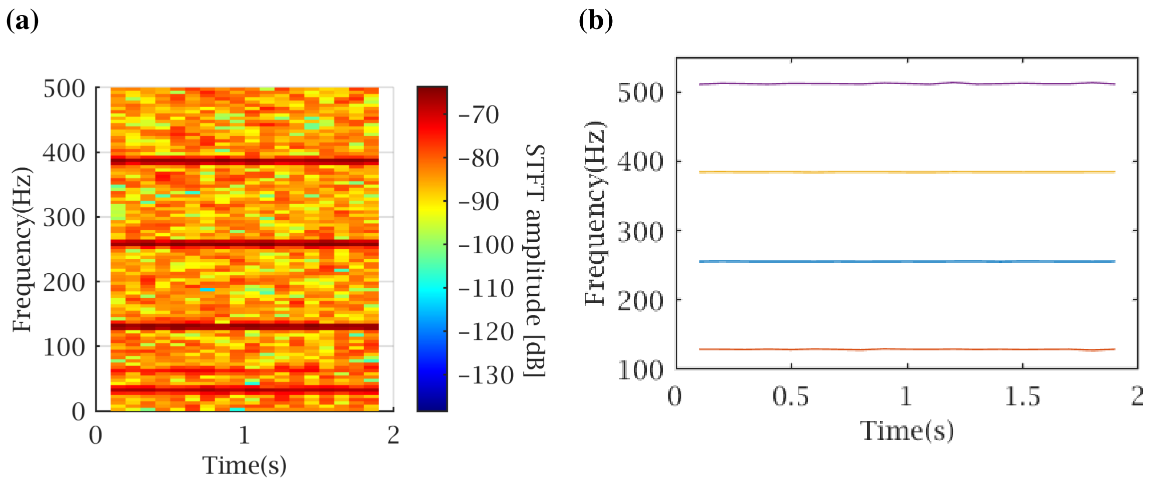

Figure 5.12: **(a)** Envelope STFT and **(b)** identified time-frequency curves of simulated defected signal with parameters $\{20.48\,\text{kHz}, 1.5, 305\}$.

5.2 Results

Until now, the approaches have been validated only by few samples. In this section, more comprehensive results for a larger number of samples are given. Table 5.1 shows the confusion matrix for the case of a high sampling rate of $20.48\,\text{kHz}$ and low noise in simulated signals. The data set for this confusion matrix consists of 75 defected signals, created by accumulating signals of each combination of relative shaft energy and low noise as indicated in Table 4.4. The 74 healthy signals were created from the defected signals by removing the BPFO component, ensuring that direct comparison between a healthy signal and its defected counterpart was possible. There are only 74 healthy signals because if the BPFO signature is removed from a signal with zero shaft and zero noise component, then only zero remains and the zero signal can not be interpreted by the algorithms. Since a zero signal will never occur in practice, this signal was not considered in the analysis.

All confusion matrices in this section adhere to the layout and interpretation introduced above. From Table 5.1, the envelope approach performs best for the high sampling rate and low noise, with a perfect TP rate in classifying all 75 defected signals as defected while only misclassifying (FP) two healthy signals as defected. The TP rate of the TFCE approach is high as well, 73 out of 75 defected signals were properly classified as defected. However, the FP rate is very high in this case, with 54 healthy signals labeled as defected by the TFCE algorithm. In contrast, the direct peak finding approach did not flag any signals as defected, therefore yielding no FP, but also no TP. The reasons for this will be explained later in the discussion.

Table 5.1: Confusion matrices for classification of simulated datasets with sampling rates of $\{20.48\,\text{kHz}, 4.096\,\text{kHz}\}$ and low noise.

	20.48 kHz		4.096 kHz	
Flag raised	Defected	Healthy	Defected	Healthy
Peak finding: True	0	0	7	6
Peak finding: False	75	74	68	68
Envelope: True	75	2	33	23
Envelope: False	0	72	42	51
TFCE: True	73	54	52	21
TFCE: False	2	20	23	53

The performance of these classification algorithms for simulated data with low noise deteriorates under lower sampling rates. The confusion matrices for low noise and a sampling rate of $4.096\,\text{kHz}$ are given in Table 5.1. The performance of the envelope approach was found to decrease significantly in terms of every metric. For example, the TP rate decreased to 33 of 75 defected signals accurately flagged. Similarly, misclassification of healthy signals increased to 23 of 74 healthy signals flagged as defected. The TFCE approach also exhibits a significantly higher rate if FP and FN with 23 and 21, respectively. Also, the direct peak finding approach does classify more healthy signals as defected than in the previous cases. Confusion matrices for the other simulated cases mentioned in Table 4.4 can be found in Appendix B.1.

The performance of the classification algorithms on the experimental dataset is provided in the confusion matrices of Table 5.2. As mentioned above, the test set was created to yield 100 signals per class. Similarly to the simulated signal with high sampling rate, again, the peak finding approach had low TP and high TN classification rates. The envelope approach and the TFCE approach again perform much better in comparison, each flagging more than 90 out of 100 signals correctly in each category. It is interesting to note that the TFCE approach did not exhibit as many FP as in the simulated low-noise case. Reasons for this will be provided in the discussion.

Table 5.2: Confusion matrices for classification of experimental dataset with sampling rates of $\{20\,\mathrm{kHz}, 4\,\mathrm{kHz}\}$.

	$20\,\mathrm{kHz}$		$4\,\mathrm{kHz}$	
Flag raised	Defected	Healthy	Defected	Healthy
Peak finding: True	4	0	8	0
Peak finding: False	96	100	92	100
Envelope: True	94	10	32	25
Envelope: False	6	90	68	75
TFCE: True	98	1	99	7
TFCE: False	2	99	1	93

For experimental data with a lower sampling rate of $4\,\mathrm{kHz}$, the confusion matrix is given in Table 5.2. Again, very few signals are labeled defected by the direct peak finding approach, but all signals that are flagged are indeed defected signals. It is also interesting to note that the number of TP for the direct peak finding approach doubles from four TP at the full sampling rate of $20\,\mathrm{kHz}$ to eight at the lower sampling rate of $4\,\mathrm{kHz}$. In contrast, the performance of the envelope approach degrades significantly, giving almost as many false positives as true positives. In the $4\,\mathrm{kHz}$ case for the experimental data, the TFCE approach still yields good results with 99 TP and 93 TN. Confusion matrices for the other experimental cases are found in Appendix B.2.

Since the experimental data [39] stems from a run-to-failure-experiment, it can also be used to study time progression of the classification behavior. Figure 5.13 shows the time progression of the (defected) flag rate for sampling rates of $20\,\mathrm{kHz}$ and $4\,\mathrm{kHz}$ in the experimental data. The flag rate was computed as the percentage of all signals out of a given 20-signal-interval that were classified as defected. For runtimes larger than $100\,\mathrm{h}$, a defect is present in the signal and the flag rate corresponds to the recall. For runtimes smaller than $100\,\mathrm{h}$ the flag rate does not have any direct correspondence to one of the introduced classifiers since the signals can be considered healthy. In Figure 5.13a, the progression for a sampling rate of $20\,\mathrm{kHz}$ is shown. Both the envelope approach and the peak TFCE approaches exhibit a steep increase to a detection rate of 1 after a runtime of 100 hours. In contrast, the peak finding approach does increase detection as well, albeit much later after approximately 150 hours runtime. Also, the rate does not increase to one, but only to 0.2. The flag rate progression for a lower sampling rate of $4\,\mathrm{kHz}$ is also shown in Figure 5.13b. Again, the TFCE approach exhibits a steep increase in flag rate up to one, however in contrast to the higher sampling rate, this increase only occurs after 130 hours. The direct peak finding approach on the other hand starts to increase its flag rate earlier than for high sampling rates, after approximately 140 hours runtime. Also, it is notable that flag rates for the direct peak finding approach increase up to 0.6, which is significantly higher than the maximum flag rate of 0.2 observed for the high frequency case. For the envelope approach at low sampling rates, no change in behavior is visible across the whole runtime. It should be noted that from the progression in Figure 5.13, a distinction between false positive flag rates at the beginning of the scan and true positive flag rates at the end cannot be made for the envelope approach.

5.3 Discussion

The present study evaluated the performance of various defect classification algorithms (e.g., direct peak finding, envelope demodulation, TFCE) under unique conditions of noise and sampling rate considerations. The ensuing discussion will elucidate understanding regarding limits of the approaches imposed by these conditions, as well as recommendations for generalization to other bearing configurations of interest. To achieve this, an analysis of unsuccessful classifications for each classification algorithm is conducted and discussed. This knowledge is then applied to a study of a range of sampling rates, shaft energies and noise to investigate areas of good and bad performance. Finally, all these findings are tied

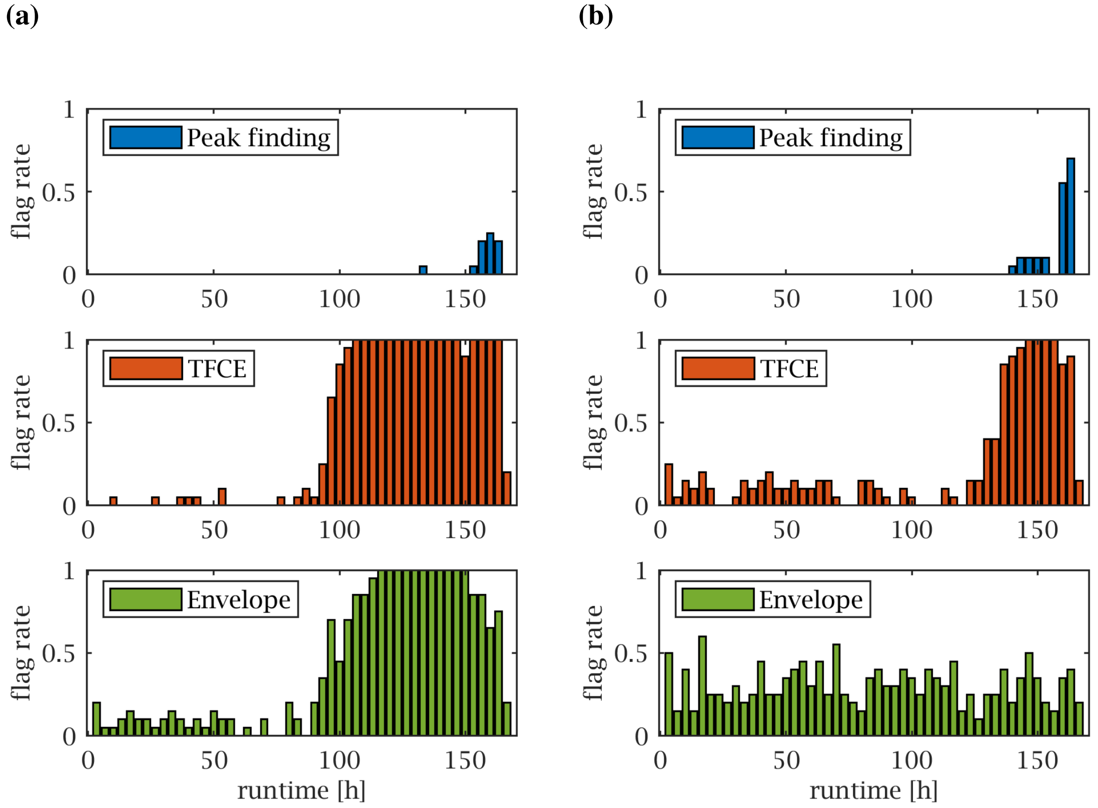

Figure 5.13: Flag rate progression for experimental data [39] at sampling rates of **(a)** 20 kHz and **(b)** 4 kHz.

together to yield conclusive recommendations.

5.3.1 Classification Approach Limitations

In the above results, the performance of each of the classification approaches was validated in assessing their behavior both from simulated as well as from experimental data. These approaches exhibit varying degrees of misclassifications, in terms of false positives and false negatives, depending on the conditions of the signal measurement. In this section, possible causes for such misclassifications will be discussed and analyzed for each classification approach.

Direct Peak Finding Approach

The confusion matrices for high sampling rates given in Tables 5.1 and 5.2 indicate that for high sampling rates and low noise, the direct peak finding approach is not capable of identifying BPFO harmonics from the signal. The reason for this is clear from the spectrum representation given in Figure 5.14 for a simulated signal with low relative noise of 100 and high sampling rate of $20.48\,\mathrm{kHz}$. The bearing resonance frequency modeled at $4\,\mathrm{kHz}$ is clearly present in the frequency spectrum and because of the modulation effects that are exploited by the envelope approach, the highest peaks in the spectrum correspond to the shaft rotation at very low frequencies or to peaks near the resonance frequency of $4\,\mathrm{kHz}$. These peaks are indicated in Figure 5.14a by red stars. It is obvious that there is no corresponding base peak at the BPFO frequency of $128\,\mathrm{Hz}$.

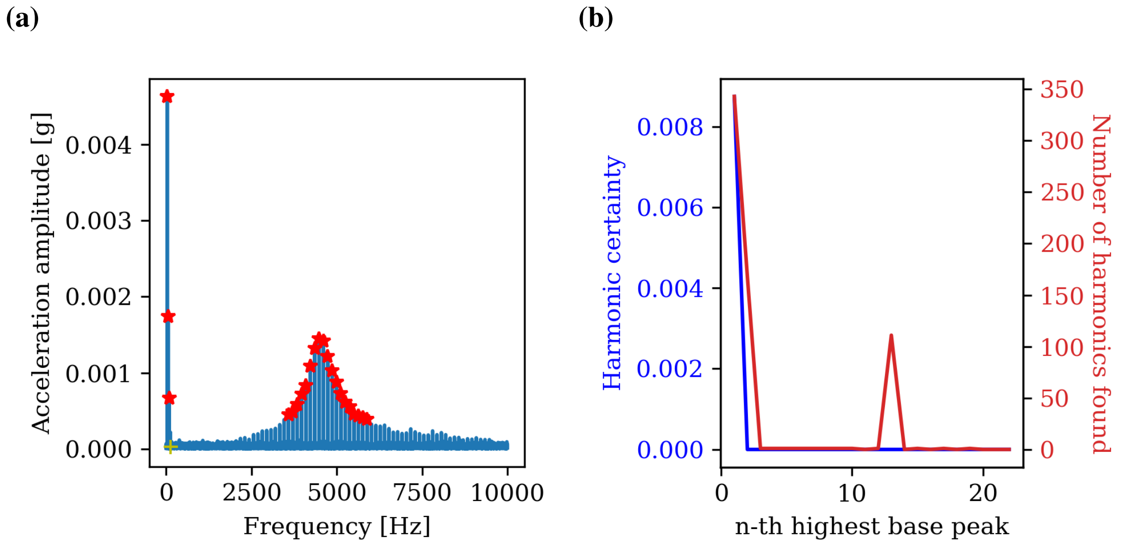

Figure 5.14: **(a)** Spectrum with location of potential base peaks indicated by stars and true BPFO frequency indicated by yellow cross, **(b)** harmonic certainty for direct peak finding approach in simulated signal with defect and parameters $\{20.48\,\mathrm{kHz}, 1.5, 100\}$.

This can also be observed with the number of identified harmonics for each possible candidate base peak presented in Figure 5.14b. The two highest candidate base peaks as well as the 13th highest candidate base peak with a large number of identified harmonics corresponding to the three first shaft rotation harmonics that are identified in the spectrum.

However, as expected, the harmonic certainty for these high numbers of harmonics is low (0.0085 or zero), as the harmonics are not distinct in the way expected for BPFO harmonics. For all other possible candidate base peaks distributed around the bearing resonance frequency, the number of identified harmonics is close to zero since even the fourth harmonic cannot be placed into the length of the spectrum that is resolved.

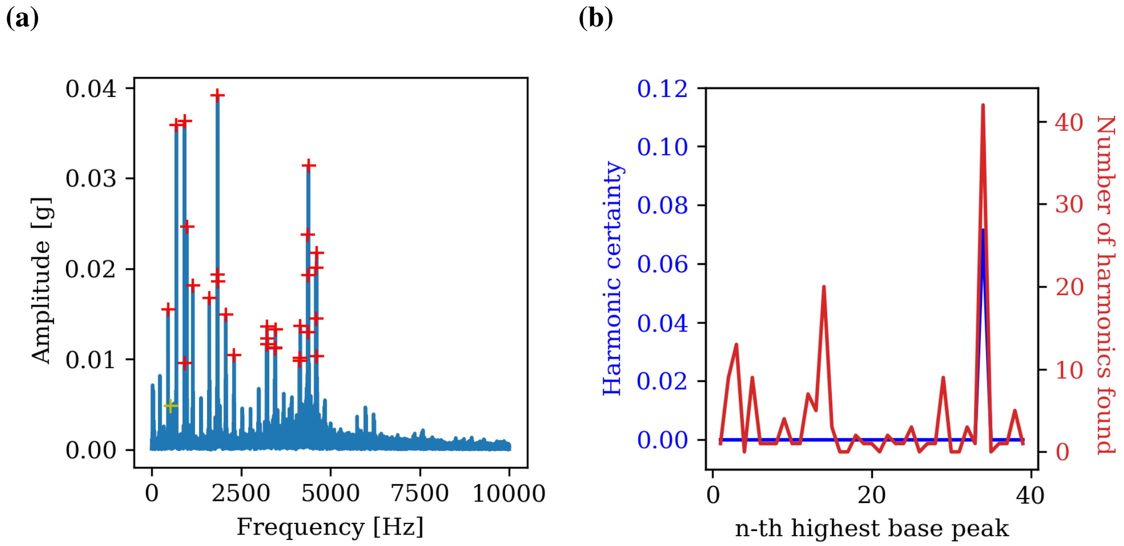

Figure 5.15: **(a)** Spectrum with potential base peaks indicated by red crosses and true BPFO peak indicated by yellow cross, **(b)** harmonic certainty for direct peak finding approach in 856th experimental signal [39] with high sampling rate of $20\,\mathrm{kHz}$.

Similar behavior is visible for the experimental data [39] as well. Figure 5.15 shows the spectrum and the progression of harmonic certainty for the 856th signal of the acquisition experiment [39], with the full sampling rate of $20\,\mathrm{kHz}$. Since this was measured in a late stage of the experiment, it is defected. Again, the identified possible candidate base peaks are indicated by red crosses in Figure 5.15a. It is apparent that the true BPFO base peak at $236\,\mathrm{Hz}$ is small compared to various other candidate base peaks in the spectrum, therefore it is not considered to be a possible candidate base peak. Even though the number of harmonics that is found varies strongly with the possible candidate base peak, no distinct harmonics are visible in the spectrum. The harmonic certainty therefore is lower than 0.1 for all possible candidate base peaks, similarly to the cases analyzed previously. This can be seen from Figure 5.15b. Thus, the BPFO is not detected and the signal is counted as FN.

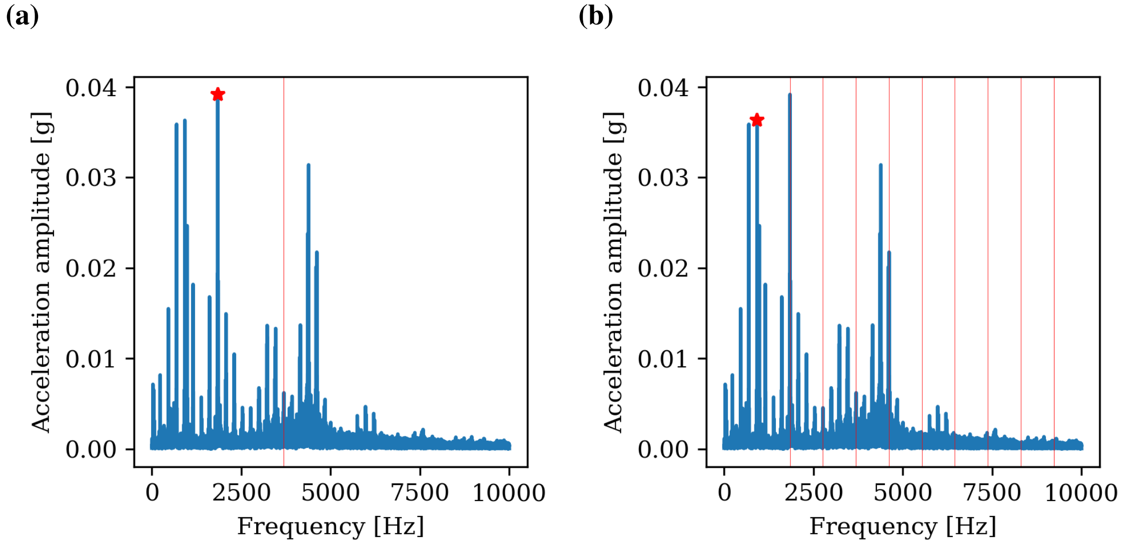

Figure 5.16: Identified possible harmonics for **(a)** highest and **(b)** second highest peak in spectrum of 856th signal of experimental data [39] with high sampling rate of $20\,\mathrm{kHz}$. Identified base peaks marked by red stars. Red lines correspond to identified harmonics.

This detection of invalid harmonics is further illustrated for the highest and second highest candidate base peak in the considered spectrum in Figure 5.16. For the highest candidate base peak at a frequency of approximately $1800\,\mathrm{Hz}$ represented in Figure 5.16a, only the first harmonic at approximately $3600\,\mathrm{Hz}$ is found. This harmonic is labeled invalid as it corresponds to a relatively low amplitude peak closely surrounded by significantly higher peaks in the spectrum. For the second highest candidate base peak at approximately $900\,\mathrm{Hz}$, more harmonics in the spectrum are found, as shown in Figure 5.16b. The first harmonic even corresponds to the highest peak in the spectrum. However, again, all harmonic peaks are labeled invalid since they are surrounded by close peaks of similar height. In summary, a likely cause for a FN with the direct peak finding approach is not identifying the BPFO base peak accurately, if it is not among the N highest points in the spectrum. A straightforward solution would be to increase N to a higher value than 23, however this leads to high computational effort as well as to an increase in FP.

Envelope Demodulation

The confusion matrices of Tables 5.1 and 5.2 indicate false classifications by the envelope demodulation algorithm for signals with lower sampling rates or with noise present, as in the experimental signal. The waveform and kurtogram of a defected simulated signal with a low sampling rate of $4.096\,\mathrm{kHz}$ are given in Figure 5.17. In comparison to the high sampling rate kurtograms in Figures 5.7b and 5.9b, the kurtogram in this case is very different. The frequency band with highest kurtosis is now a very small frequency band of $16\,\mathrm{Hz}$ at the highest level 7 with a center frequency of $1.58\,\mathrm{kHz}$.

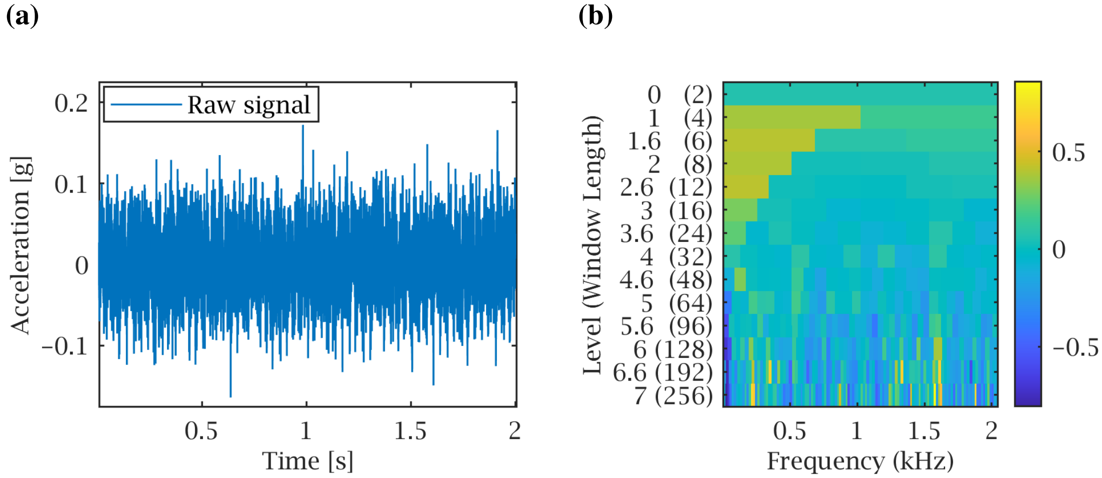

Figure 5.17: **(a)** Waveform and **(b)** kurtogram for simulated signal with defect and parameters $\{4.096\,\mathrm{kHz}, 1.5, 300\}$.

The waveform, filtered and envelope signal for this condition are shown in Figure 5.18a. From the spectrum represented in Figure 5.18b, it is obvious that due to the properties of the digital FIR lowpass filter, the filtered signal has a bandwidth of approximately $500\,\mathrm{Hz}$, which is significantly larger than the ideal bandwidth of $16\,\mathrm{Hz}$ required from the kurtogram. The 11th, 12th, 13th and 14th harmonic of the base peak at $1408\,\mathrm{Hz}$, $1536\,\mathrm{Hz}$ $1664\,\mathrm{Hz}$ and $1792\,\mathrm{Hz}$, respectively, are still distinctly visible in the filtered spectrum. Nonetheless, the envelope demodulation is unable to transport these frequencies to the front in this case. Even though the envelope does enfold the filtered signal, this implies that no peaks at the BPFO frequency and its harmonics are visible in the envelope spectrum.

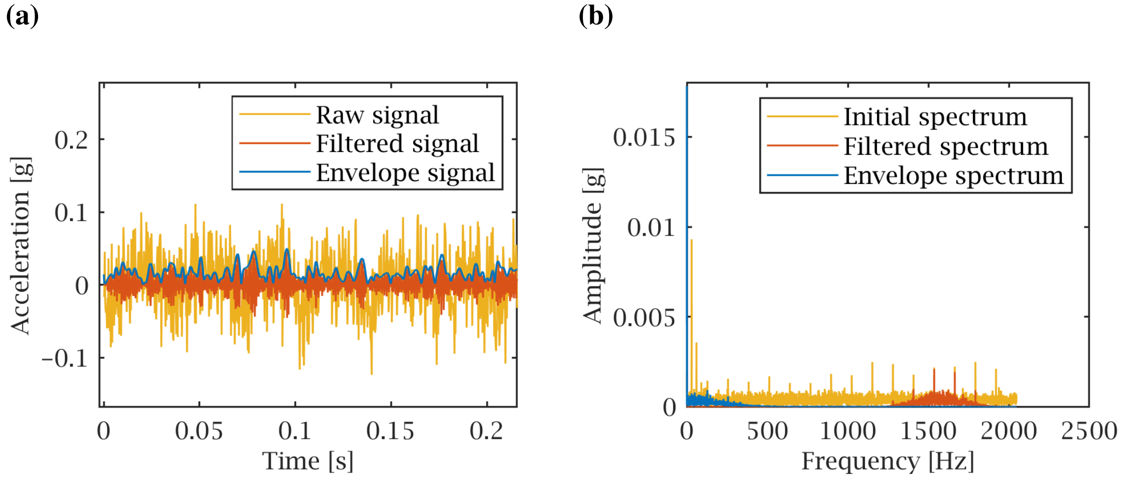

Figure 5.18: Waveform **(a)** and spectrum **(b)** for the three pre-treatment steps in simulated signal with defect and parameters $\{4.096\,\mathrm{kHz}, 1.5, 300\}$.

Time-Frequency Curve Extraction

The TFCE algorithm also is prone to false positives, as the confusion matrices in Tables 5.1 and 5.2 indicate. Figure 5.19 presents a simulated healthy signal with a low relative noise energy of 150 and a relative shaft energy of 1.5. This results in a shaft rotation amplitude of $0.004\,\mathrm{g}$ as visible in Figure 5.19b, which is high in contrast to the noise. The envelope STFT is presented in Figure 5.20a. The impact of the shaft rotation to the STFT representation is large. The strong horizontal red line at $30\,\mathrm{Hz}$ represents the shaft rotation, and the first two harmonics at $60\,\mathrm{Hz}$ and $90\,\mathrm{Hz}$ can be seen as fainter red horizontal lines. These curves were also identified by the curve detection algorithm as illustrated in Figure 5.20b. In addition, one fluctuating curve at approximately $800\,\mathrm{Hz}$ was found. However, because the first three identified curves correspond to the shaft harmonics, they do have a constant ratio towards each other and are consequently misinterpreted as BPFO base frequency and harmonics. Huang, Baddour, and Liang [33] addressed this issue by comparing the constant curve ratio to the expected ratio that is given by the bearing configuration as introduced above. However, one assumption of the present work is insufficient knowledge of bearing configuration, so this is not an option.

For lower sampling rates, false negatives also begin to occur for the TFCE classifier.

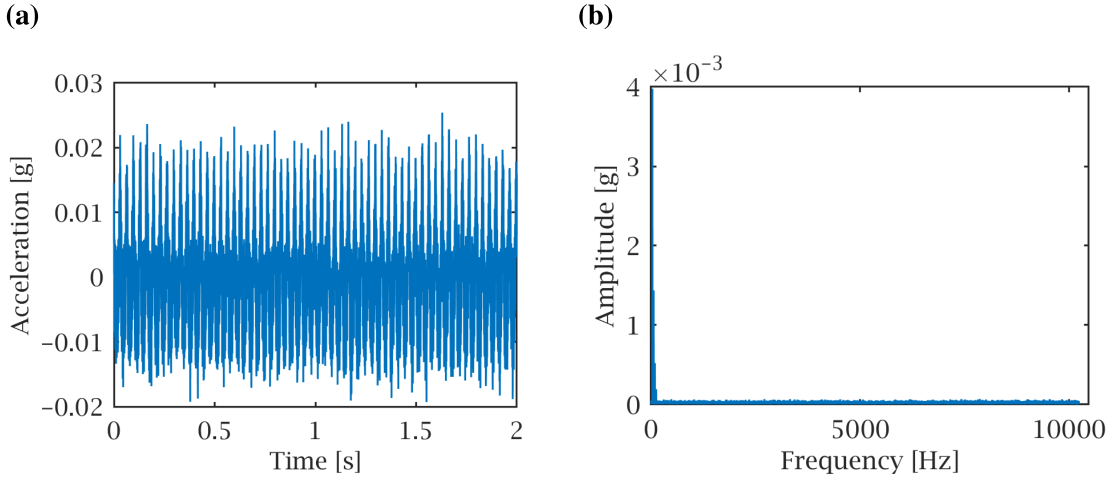

Figure 5.19: **(a)** Waveform and **(b)** spectrum of simulated false positive signal with sampling rate $20.48\,\mathrm{kHz}$, relative noise energy of 150 and relative shaft energy of 1.5.

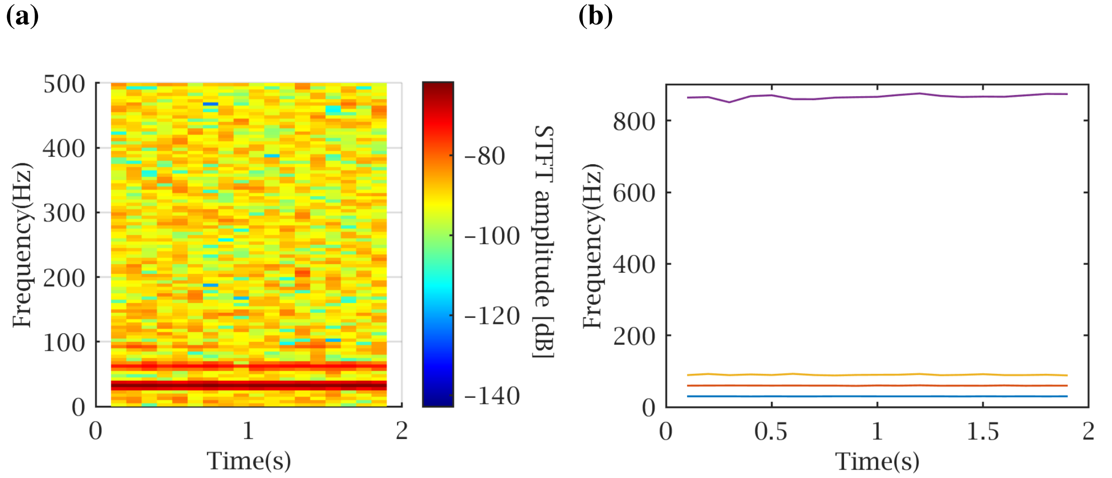

Figure 5.20: **(a)** Envelope STFT and **(b)** identified time-frequency curves of simulated false positive signal with sampling rate $20.48\,\mathrm{kHz}$, relative noise energy of 150 and relative shaft energy of 1.5.

Figure 5.21 presents the spectrum and the STFT of a simulated signal with defect, sampled at $4.096\,\mathrm{kHz}$. In the spectrum of Figure 5.21b, peaks at the BPFO frequency of $128\,\mathrm{Hz}$ and the harmonics are clearly identifiable by the human eye. However, the STFT spectrum in Figure 5.22a looks different. Overall, the STFT of this signal exhibits very high values, indicated by many red tones. There is no clear line visible, even though the frequency resolution of $4\,\mathrm{Hz}$ is capable of resolving peaks spaced at $128\,\mathrm{Hz}$. A possible explanation why the peaks are not visible in the spectrum would be the picket fence effect introduced

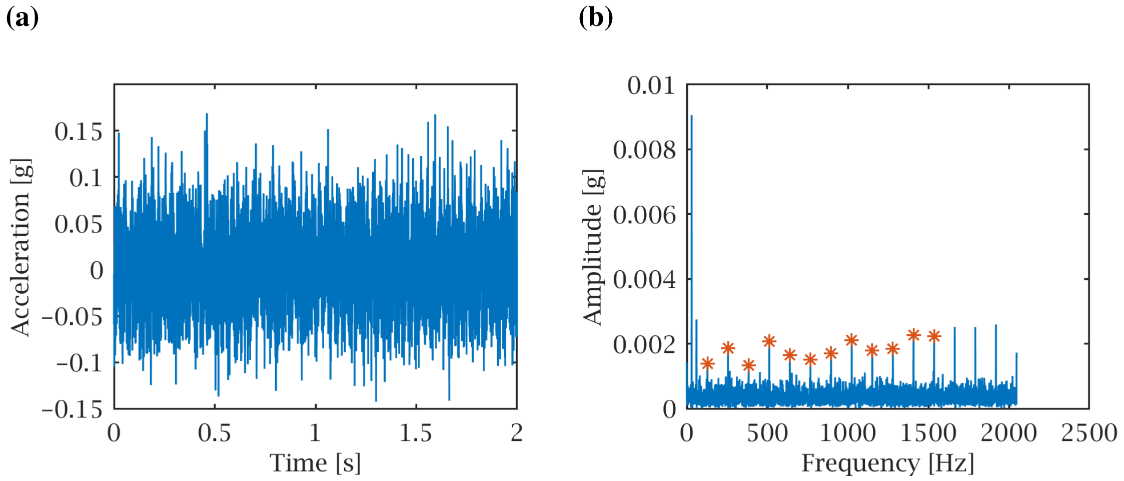

Figure 5.21: **(a)** Waveform and **(b)** spectrum of simulated false negative signal with sampling rate $4.069\,\mathrm{kHz}$, relative noise energy of 305 and relative shaft energy of 1.5. BPFO peak and first 11 harmonics indicated by stars.

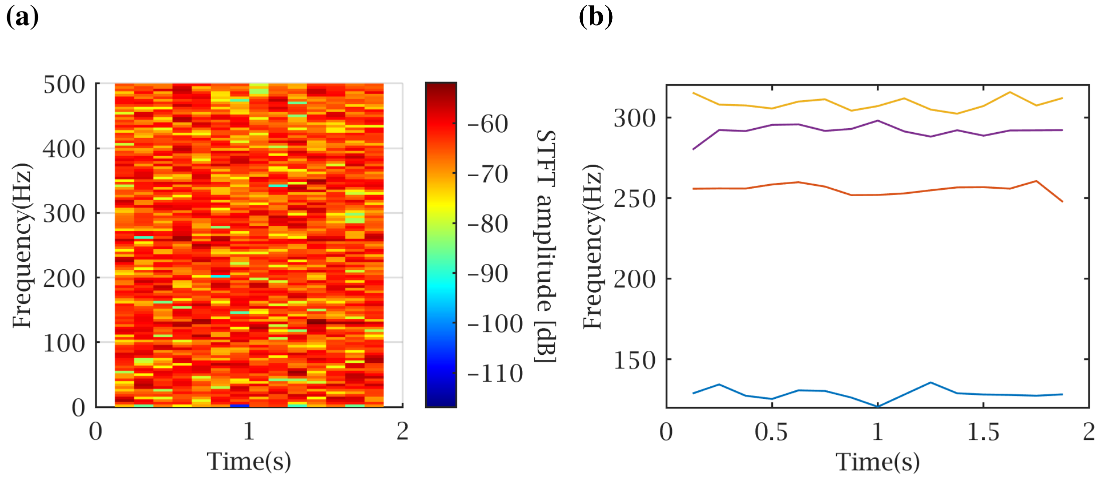

Figure 5.22: **(a)** Envelope STFT of simulated false positive and **(b)** identified time-frequency curves in simulated false negative signal with sampling rate $4.096\,\mathrm{kHz}$, relative noise energy of 305 and relative shaft energy of 1.5.

previously. The narrowly spaced peaks are not represented in the STFT frequency samples, which are too far apart.

Accordingly, the curves identified by the curve finding algorithm and illustrated in Figure 5.22b are not horizontal, but instead quite erratic. A constant ratio is not identified and therefore the signal is falsely flagged negative. A way to mitigate this issue would be to increase the frequency resolution, at the same time reducing the time resolution even further. This is a good option if the signal is long and an acceptable time resolution can be obtained at the same time as a better frequency resolution, but for rather short signals, a minimum number of time steps must be contained to ensure that the variance can get high for non-constant ratios.

5.3.2 Effects of Shaft Energy, Sampling Rate and Noise

From the previous section, it is evident that shaft energy, noise and sampling rate all can cause false negatives or false positives. The ensuing analysis quantifies and discusses the extent of these parameters for classification performance in the three approaches. The bearing operating and noise conditions for the simulated data used in this case are described in Tables 4.1 and 4.2, respectively. Since the parameters used to generate the simulated signals are known a priori, it is possible to study the impact that relative shaft energy and sampling rate together have on the classification results. Figure 5.23 shows flag rates for various sampling rate and relative shaft energy combinations in a low noise range for the three approaches. For every sampling rate and shaft energy combination, five signals with noise levels between 0 and 230 and BPFO signature were analyzed in the defected case, and the corresponding signals without BPFO signature were analyzed in the healthy case. In the first row, which considers defected signals, a high flag rate indicates desired performance of the classifier, whereas in the second row with healthy signals, a high flag rate indicates a high number of false positives (i. e., undesired behavior).

Generally, the influence of relative shaft energy on the detection behavior is smaller than the influence of sampling rate. For the direct peak finding approach presented in Figure 5.23a, the performance for a low sampling rate of $4.096\,\mathrm{kHz}$ decreases for relative shaft energies higher than one. This can be explained by the same phenomenon as discussed for high sampling rates earlier. The high rotational peaks may lead to the fact that the actual BPFO base peak is not represented in the spectrum.

Similarly, the performance of the TFCE approach also worsens if the relative shaft

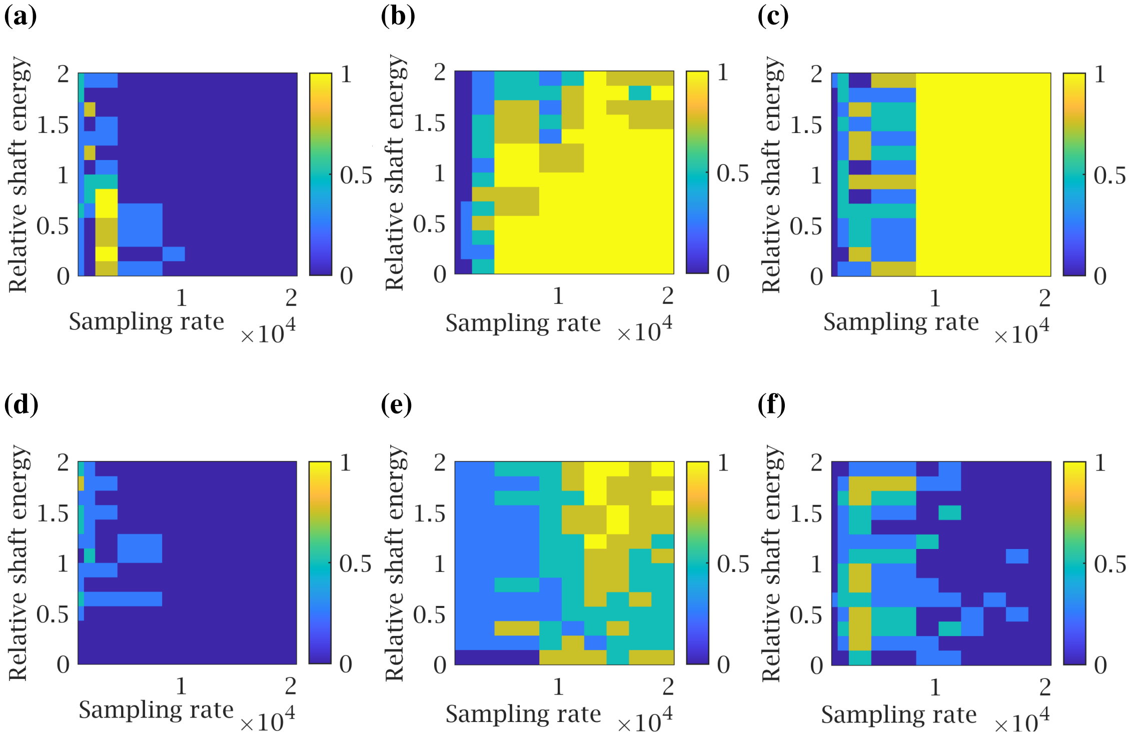

Figure 5.23: Flagged signal ratio distribution under low noise for the three approaches. Color indicates flag rate. Defected signals are considered in the top row, healthy signals in the bottom row: **(a,d)** Direct approach, **(b,e)** TFCE approach, **(c,f)** Envelope approach.

energy increases. Due to the fact that the shaft rotation signature needs to be identified as part of the TFCE approach [33], it could be intuitive to assume that higher shaft rotation energies lead to better detection. However, from Figure 5.23b it is obvious that the opposite is true. In sampling rate ranges between 6 kHz and 12 kHz, shaft energies higher than 1.5 lead to worse results. A possible reason for that is that the shaft rotation harmonics become more present in the spectrum if the shaft energy is higher. Due to the frequency resolution loss that is inherent with the transition from FFT to STFT and leakage effects, peaks can be smeared across a range of 10 Hz in the frequency domain. Therefore, the fourth shaft harmonic at 120 Hz may interfere with the BPFO base frequency at 128Hz and prevent the curve detection algorithm from finding a constant horizontal curve, which could then be detected by the TFCE algorithm. Additionally, as visible in Figure 5.23e, for sampling rates larger than 12 kHz, the rate of false positives increases for relative shaft rotation energies greater than 1. The reason for this was addressed previously, the harmonics of the shaft rotation are wrongly interpreted as BPFO harmonics.

For frequencies larger than $10\,\mathrm{kHz}$, the envelope approach yields very good results. While the flag rate with defected signals is consistently one, as visible in Figure 5.23c, the rate of false negatives as shown in Figure 5.23f only exceed 0.5 in three cases. In stark contrast, for sampling rates lower than $10\,\mathrm{kHz}$, both in defected and healthy signals the flag rates vary between 0.75 and 1. As elaborated above, the most likely reason for this is that the range of the bandpass filter with the kurtogram is not appropriate and thus the envelope modulation does not manage to extract the correct information. This can lead to both misdetection of noise as BPFO harmonics as well as to attenuation of the present BPFO peaks. The shaft energy does not seem to have any impact on the detection accuracy for the envelope approach. However, increased noise can deteriorate the performance of the envelope approach easily.

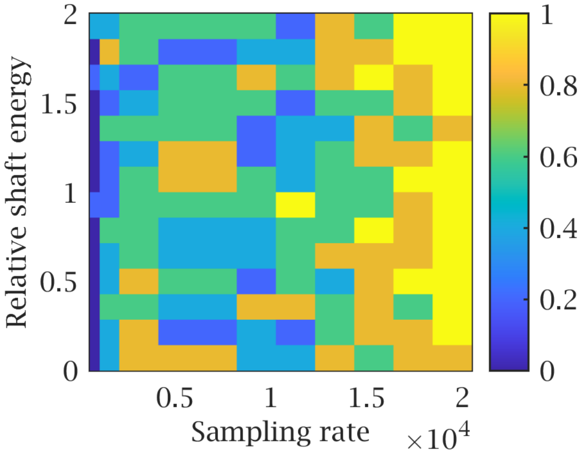

Figure 5.24: Flag rate for envelope approach for defected signals with medium noise.

In Figure 5.24, the envelope flag rates for defected signals with medium noise are presented. Reliable flagging of defected signals only begins at a sampling frequency of $20\,\mathrm{kHz}$, whereas for low noise reliable flagging starts at $10\,\mathrm{kHz}$, as apparent from Figure 5.23c. Due to the noise and possible resulting undesired harmonics as well as reduced harmonic certainty in the resulting envelope spectrum, flag rates vary throughout lower sampling rates.

Also for the TFCE approach, the sampling rate threshold for reliable detection increases with increased noise, from $6\,\mathrm{kHz}$ in the low noise case to $12\,\mathrm{kHz}$ in the medium noise case as pictured in Figure 5.25. Interestingly, the rate of false positives in this case as visible in Figure 5.25b decreases with increasing relative shaft energy, opposite to the low noise case discussed earlier. A possible reason for this is that increased noise masks the shaft rotation harmonics that can lead to false detections for low noise as detailed earlier.

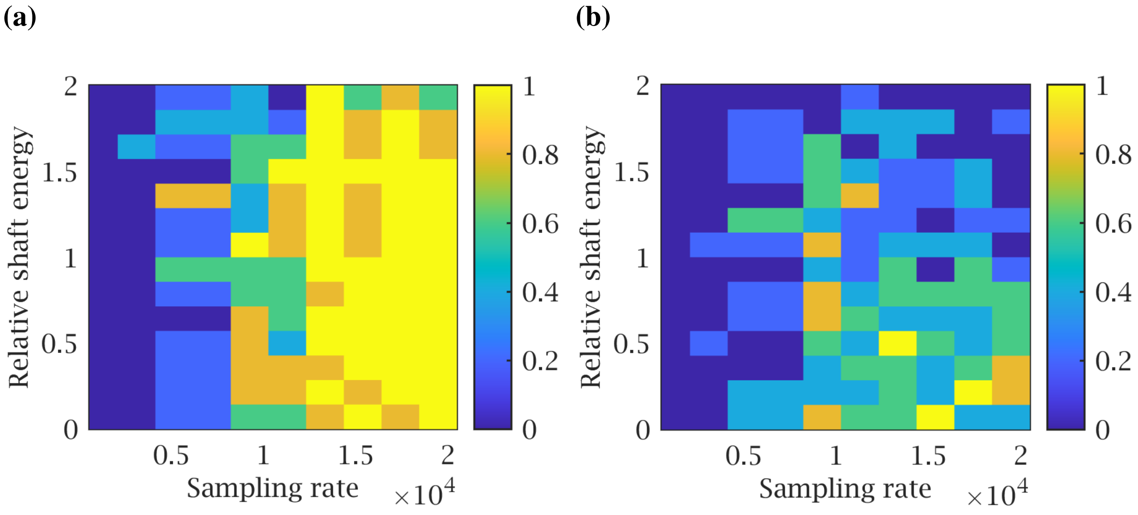

Figure 5.25: Flag rate for TFCE approach with medium noise: **(a)** defected signals, **(b)** healthy signals

5.3.3 Recommendations

Optimal use of the three defect detection approaches depends ultimately on expected characteristics of the signal, including noise and shaft energy, as well as the sampling rate employed. The classification measures as explained previously for low noise and sampling rates of $20.48\,\mathrm{kHz}$ and $4.096\,\mathrm{kHz}$ in a simulated signal are compared in Table 4.4. For the sampling rate of $20.48\,\mathrm{kHz}$, precision and F1 score of the direct peak finding approach cannot be computed since there were no defected signals detected. For a high sampling rate, the envelope approach scores highest in all three categories with a recall of $100\,\%$, precision of $97.4\,\%$ and F1-score of $98.68\,\%$. This is consistent with the notion that the envelope approach is one of the most common approaches used for bearing fault detection at sufficiently high sampling rates [3, 2]. However, for low sampling rates, the TFCE approach loses recall at $69.33\,\%$, while the F1 score stays relatively constant at around $70\,\%$. In contrast, the envelope demodulation algorithm and the direct peak finding approach only yield F1 scores of $15.91\,\%$ and $50.38\,\%$, respectively. This indicates that while the envelope demodulation is a very suitable approach for higher sampling rates, the TFCE approach shows more robust behavior over a wider range of sampling rates.

Table 5.3: Simulated performance metrics for the three approaches and a sampling rate of $20.48\,\mathrm{kHz}$ and low noise.

Sampling rate	Approach	Precision [%]	Recall [%]	F1-Score [%]
$20.48\,\mathrm{kHz}$	Peak finding	N/A	0	N/A
	Envelope	97.40	100.0	98.68
	TFCE	57.48	97.33	72.28
$4.096\,\mathrm{kHz}$	Peak finding	53.85	9.33	15.91
	Envelope	58.93	44.0	50.38
	TFCE	71.23	69.33	70.27

Table 5.4: Experimental data [39] performance metrics for a sampling rate of $20\,\mathrm{kHz}$.

Sampling rate	Approach	Precision [%]	Recall [%]	F1-Score [%]
$20\,\mathrm{kHz}$	Peak finding	100	4.0	7.69
	Envelope	90.38	94.0	92.16
	TFCE	98.99	98.0	98.49
$4\,\mathrm{kHz}$	Peak finding	100	8.0	14.81
	Envelope	56.14	32.0	40.76
	TFCE	42.42	28.0	33.73

This is further suggested by a comparison of experimental data classification scores, as given in Table 5.4. For the experimental data at high sampling rate, both the envelope approach and the TFCE approach perform well with F1-scores of $92.16\,\%$ and $98.49\,\%$, respectively. Again, for lower sampling rates, performance decreases for F1-scores of $40.76\,\%$ and $33.73\,\%$. It is interesting that while both approaches still score in similar F1 score ranges, the TFCE approach yielded slightly lower F1 score in this case.

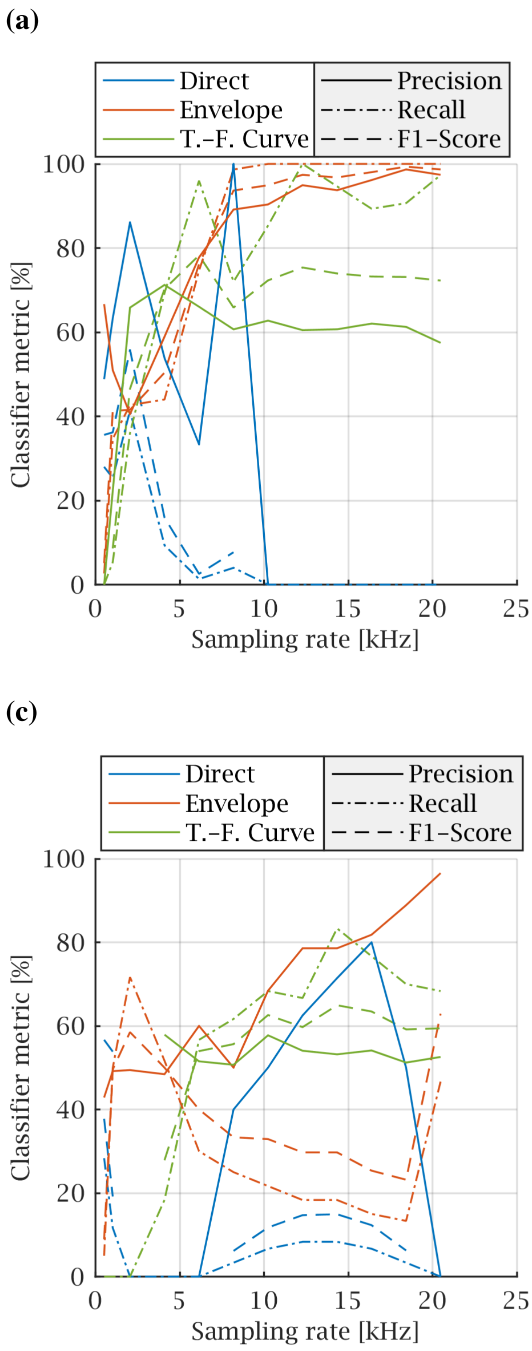

Figure 5.26: Simulation classifier metrics over sampling rate for various noise levels: **(a)** low noise, **(b)** medium noise, **(c)** high noise, **(d)** experimental data [39].

Figure 5.26 shows precision, recall and F1 score progression for low, medium and high noise simulated data, as well for the experimental data [39]. It is important to note that F1-score and precision values can be undefined, as mentioned with respect to Table 5.3.

This explains why some F1-score matrices have gaps as for example the F1 score curve for peak finding under high noise as depicted in Figure 5.26c for sampling rates between $1\,\mathrm{kHz}$ and $4\,\mathrm{kHz}$. It is immediately visible that the direct peak finding approach had the worst recall values, rarely exceeding $20\,\%$ across all pictured cases. However, especially for the experimental data [39], the peak finding algorithm showed high precision of up to $100\,\%$ starting at a relatively low sampling rate of $4\,\mathrm{kHz}$. It is also apparent that the TFCE approach was more robust w. r. t. noise when compared to the envelope approach. While the envelope approach constantly maintained all scores above $90\,\%$ for low noise and sampling rates higher than $10\,\mathrm{kHz}$, as pictured in Figure 5.26a for the higher noise cases and for the experimental data, the F1 score of the TFCE approach was consistently higher than that of the envelope approach.

These sampling rates graphs can be used as an indicator for which method is preferred, depending on a particular bearing system's noise level and sampling rate. For low-to-medium noise and high sampling rates, the envelope approach is advisable. If the sampling rate is at least $20\,\mathrm{kHz}$, then the envelope approach consistently presents precision, recall and F1 scores above $90\,\%$. This is confirms literature results, stating that the envelope demodulation approach is superior to other approaches like direct spectrum methods or the cepstrum method if a reasonably wide frequency range is known [3]. However, for lower sampling rates and some noise present, the performance of the envelope approach is inferior to that of the TFCE algorithm. Therefore, if only one algorithm can be used due to computation effort constraints, it is advisable to use the TFCE approach [33]. Unfortunately, for all three simulated noise cases, the recall and F1-score can not exceed 80%, still being the best algorithm considered. The direct peak finding approach has very low recall values throughout sampling rates. However, as apparent from the experimental data given in Figure 5.26d, the precision of the direct peak finding approach can be very good.

Therefore, the direct peak finding approach can be useful if applied as secondary identification method. In many industrial cases, multiple individual signals from the same sensor are regarded. To mitigate the influence of statistic variation in classifications due to noise, it can be useful to classify multiple signals from the same bearing, and then check the flag ratio of defected to undefected signals. This can either be done using just one algorithm, or using peak finding as secondary approach. If a high ratio of signals from one sensor are flagged by the TFCE approach and some of them are flagged by the peak finding approach, due to the high precision of the peak finding approach, the likelihood of identifying a defect is increased.

5.3.4 Generalization to Other Bearing Configurations

The experimental validations in the present work were carried out using rotation speeds of $30\,\mathrm{Hz}$ and BPFO frequencies of $128\,\mathrm{Hz}$ and $236\,\mathrm{Hz}$, with a modeled resonance frequency of $4\,\mathrm{kHz}$ for the simulated signal, while the bearing resonance frequency for the experimental data is unknown. In industrial application, these numbers can vary significantly, impacting the location of the signatures in the frequency domain. However, due to the fact that reasons for the various approach behaviors were found, some scaling rules to other frequencies can be established.

As discussed above, the limiting factor for the envelope demodulation approach is that the sampling rate must be high enough such that the bearing resonance frequency is well represented in the spectrum. If the bearing resonance frequency is known from CAD models or can be estimated [42], then if a sampling rate larger than 2.5 times this resonance frequency is feasible, envelope demodulation is recommended. The sampling rates for the other approaches depend roughly on the shaft and first few bearing harmonics. Thus, all sampling rate and frequency spectrum recommendations can be scaled with the location of these peaks in the spectrum. Hence it is advisable that the sampling rate is at all times high enough so that those can be present in the spectrum. For determining a minimal needed sampling rate, it is therefore necessary to determine an upper bound for possible BPFO frequency and then multiplying this frequency by eight to obtain the minimal recommended sampling rate for any approach.

CHAPTER 6

CONCLUSION

The purpose of this work was to understand how low sampling rates, unknown bearing parameters and rotational speeds effect bearing fault detection. Simulated as well as experimental signals were used to evaluate three different approaches in sampling rates ranging between $2\,\mathrm{kHz}$ and $20\,\mathrm{kHz}$. The direct peak finding approach was better for low sampling rates than for high sampling rates, but in general, detection rates are insufficient for standalone use. The envelope detection approach [3] performed very well for high sampling rates that include the bearing resonance frequency in the spectrum, however the detection rate decreases strongly with increased noise and decreased sampling rate. Although the TFCE approach can not consistently yield results as good as the envelope detection approach for low noise and high sampling rates, it shows the most robust behavior for the lower sampling rate cases.

To answer the first research question posed in Section 1.2, state-of-the-art detection methods exhibit significant precision and accuracy losses under reduced sampling rate constraints. However, if the sampling rates are large enough that some bearing harmonics are represented, detection is still possible. The BPFO detection method that is best suited for lower sampling rates and usual amounts of noise is TFCE, if possible enhanced by accumulation and thresholding of multiple signals and enhanced by the direct peak finding algorithm. However, for high sampling rates where the bearing resonance harmonic is certainly represented, especially in low-noise environments, envelope demodulation is more accurate and a decision can be made after regarding fewer signals.

6.1 Contributions

A novel peak finding algorithm that can be used by itself as well as as a second step for the envelope demodulation algorithm has been introduced. The way of identifying individual harmonics by their harmonic score as introduced in the present work has not been covered in literature. The behavior of established BPFO detection algorithms under low sampling rates had not been previously studied. The two established approaches [3, 33] as well as the direct peak finding approach were examined under decreasing sampling rates. The study showed that decreased sampling rate leads to decreased behavior of the TFCE and envelope approaches, whereby the TFCE approach behaves more robustly than the envelope approach.

6.2 Future Work

The present work mainly gave recommendations on the performance of the three algorithms as separate instances. As hinted previously, detection quality can be greatly improved by combining measurements from multiple signals as well as combining results from multiple approaches with differing precision and recall combinations. A direct extension of the work presented in this thesis would be quantifying these beneficial effects and exploring various ways of joining the different approaches to obtain a better ensemble approach. With increasing digitalization in manufacturing, computation on edge devices is very common. In large manufacturing areas, storing and later analyzing vibration signals from multiple sensors can easily become very costly. Potentially, these classification methods can be transferred to edge devices such as the Beaglebone Black. Since all considered approaches work as black-box models without a need of exact knowledge of bearing parameters, it could be beneficial to develop a portable edge device, which can be transported throughout multiple locations in a manufacturing hall, giving instant classification results based on these approaches. For such a device, the constraints postulated such as uncertain rotational speed and unknown bearing configuration are crucial to obtain an easy setup with low cost. It could be very cost-effective to use hardware designed for high sampling rates and then use a lowpass filter and downsampling as needed for the specific use-case. If the methods introduced in this thesis are employed with the intent of real-time analysis, the choice of optimal parameters introduced above could be re-visited. For real-time implementation, a higher importance should be placed on computational power as well as RAM usage during the optimization of parameters.

Appendices

APPENDIX A

DETECTION ALGORITHM IMPLEMENTATIONS.

A.1 Harmonic Peak Finding

The harmonic peak finding algorithm is presented here.

```
import numpy as np
import scipy.signal as ss
from typing import Tuple
from awesome_vibration_toolbox.data_prep.Measurement import Measurement

def detectBpfo(measurement: Measurement, numberOfCombinationTest = 18, nbHarmo = 4,
               minSizeHarmo = 0.01, freqRangeHarmo = 2, minSnr = 0.8, minBPFOFreq = 15,
               prominence_req=0.5, harmo_cert_threshold=0.1, FFT_type: str = ’density’) \
        -> Tuple[bool, str, dict]:
    """
    finding if a possible bpfo is in the spectrum.

    :param freqs: frequency 1D array
    :param numberOfCombinationTest: number of combinations we try, i.e.,
                                    how many  peaks we assume could potentially be
                                    bpfo fundamentals
    :param nbHarmo: number of harmonics of the fundamental we are able to find
    :param minSizeHarmo: minimum size of the harmonics of the dominant peak
    :param freqRangeHarmo: when looking for harmonics, taking the multiple of
                             dominant peak frequency, and looking above and under
                             this value for potential harmonic. Upper and lower
                             bound = k * freqDominantPeak +- freqRangeHarmo / 2
    :param minSnr: minimum signal to noise ratio. The signal is assumed to be the
                     bpfo signal only composed of the fundamental and harmonics and
                     the noise is all the remaining spectrum
    :param minBPFOFreq: Minimal possible BPFO frequency. Our machines run with at least
                        1200 rpm (according to Kevin), therefore the spindle frequency
                        should not be smaller than 20Hz.
                        The BPFO freqeuncy is always greater than the spindle frequency,
                        therefore 15Hz is a very conservative lower bound.

    :return:
    (bool) possibleBpfo: True if the signal may have a bpfo
    (dict) dictBpfo: containing all relevant information about possible bpfo if
                     flag possibleBpfo is True. Empty if False.
                        Possibly returned:
                        --> frequencyBpfo
                        --> rpm
                        --> bpfoIndex
                        --> snr
                        --> score
                        --> harmo_certainty
    """
    freqs, amps = measurement.get_FFT(FFT_type)
    possibleBpfo, dictBpfo = False, {’maxNumHarmo’: 0, ’maxScore’: 0,
                                     ’bestBaseIdx’: None}
    # going testing for all the largest peaks if we find a set {fundamental, harmonics}
    peakRank = 0
    if minBPFOFreq - freqRangeHarmo / 2 < freqs[4]:
        # This is the case where we can’t distinguish peaks because the resolution is
        # too small.
        # We require at least 4 samples between the peaks
        minBPFOFreq = freqs[4] + freqRangeHarmo / 2
    # print(’minBPFOfreq: ’, minBPFOFreq)
    while peakRank < numberOfCombinationTest :
        peakRank += 1
        indexPeak = findLargePeak(amps, rank = peakRank)
        if indexPeak is False:
            break
        if freqs[indexPeak] < minBPFOFreq:
            # This is the case that the found peak cannot be a bpfo peak because it is
            # too low in frequency.
            # Then we test one more peak.
            numberOfCombinationTest += 1
            if numberOfCombinationTest > len(freqs)/2:
                #  We can not find a BPFO because the minimal BPFO frequency is set too
                #  high
                raise ValueError("The specified minimal BPFO frequency is too high!")
            continue

        minSize = minSizeHarmo * np.amax(amps)

        # checking if we find enough harmonics
        nbHarmoFound, listIndHarmo, score, harmo_certainty = \
            findHarmoOfPeak(indexPeak, freqs, amps, minSize,
                            freqRangeHarmo = freqRangeHarmo,
                            prominence_req=prominence_req,
                            harmo_cert_threshold=harmo_cert_threshold)
        if nbHarmoFound > dictBpfo['maxNumHarmo']:
            dictBpfo['maxNumHarmo'] = nbHarmoFound
        if score > dictBpfo['maxScore']:
            dictBpfo['maxScore'] = score
            dictBpfo['bestBaseIdx'] = indexPeak
        # we also check the signal to noise ratio is large enough. If not then the
        # likelihood we raise a false positive is too high
        # The chosen BPFO family is the one with the highest sum of harmonic values
        # The peak value is neglected
        snr = SNR(amps, np.array(np.insert(listIndHarmo, 0, indexPeak), dtype = int))
        if nbHarmoFound >= nbHarmo and snr > minSnr:
            if possibleBpfo:  # this is the case where we have found a harmonic family
                if score <= dictBpfo["score"]:
                    # this is the case where our first test was better
                    continue
            possibleBpfo = True
            dictBpfo["frequencyBpfo"] = freqs[indexPeak]
            dictBpfo["harmonicsIndex"] = listIndHarmo
            try:
                dictBpfo["harmonicsFreqs"] = freqs[listIndHarmo]
            except IndexError:
                dictBpfo["harmonicsFreqs"] = []
            dictBpfo["rpm"] = freqs[indexPeak] * 60
            dictBpfo['bpfoIndex'] = indexPeak
            dictBpfo["snr"] = snr
            dictBpfo["score"] = score
            dictBpfo["harmo_certainty"] = harmo_certainty

    return(possibleBpfo, 'Harmonics', dictBpfo)


def findLargePeak(amplitudes, rank = 1, minBaseHeight=0.05) :
    """
    Detect if a peak is very dominant in the spectrum. Because, if bpfo there
    is, there should be a peak located at the bpfo frequency. Since we do not have
    this information we will look for the largest peak

    (float array 1 * n) amplitudes : amplitudes of the spectrum
    (int) rank : rank of peak in terms of size. If rank == 1, then we look for the
                 largest spectrum peak. If rank == 2, then we look for the second
                 largest etc.

    return
    (int) indexPeak : index of the peak in the amplitudes array

    """
    # size of peak we are looking for
    peak_indices, _ = ss.find_peaks(amplitudes)
    sorted_amps = sorted(amplitudes[peak_indices], reverse=True)
    try:
        size = sorted_amps[rank-1]
    except IndexError:
        # Signal does not have any peaks
        return (False)
    if size > minBaseHeight*np.amax(amplitudes):
        # getting index of it
        indexPeak = np.argmin(np.abs(amplitudes - size))
    else:
        indexPeak = False
    return (indexPeak)

def findHarmoOfPeak(indexBpfoPeak, freqs, amplitudes, minSizeHarmo, freqRangeHarmo = 6,
                    prominence_req=0.5, harmo_cert_threshold=0.1):
    """
    finding the harmonics of the specified peak
    (int) indexPeak : index of the dominant peak in the spectrum
    (float array 1 * n) freqs : frequency 1D array
    (float array 1 * n) amplitudes : amplitudes 1D array
    (float) minSizeHarmo : minimum size of the harmonics of the dominant peak
    (float) freqRangeHarmo : when looking for harmonics, taking the multiple of
                             dominant peak frequency, and looking above and under
                             this value for potential harmonic. Upper and lower
                             bound = k * freqDominantPeak +- freqRangeHarmo / 2

    returns
    (int) nbHarmo : number of harmonics found
```

```python
    (float 1 * n) listIndHarmo : index of found harmonics
    """

    # getting major peaks
    indexPeaks, dictio = ss.find_peaks(amplitudes, distance = 5, height = minSizeHarmo,
                                       prominence=minSizeHarmo*prominence_req)
    # counting the number of harmonics. If an harmonic is not found then ending
    # search
    nbHarmo, listIndHarmo, flag, freqDomPeak, inf = 0, [], True, freqs[indexBpfoPeak], 0

    while flag and inf < len(freqs):
        # looking for indexes between which we look for the harmonic
        workingIndex = \
            np.argwhere(( (nbHarmo + 2) * (freqDomPeak - freqRangeHarmo / 2) <= freqs ) *
                        (freqs <= (nbHarmo + 2) * (freqDomPeak + freqRangeHarmo / 2)))

        if len(workingIndex) > 0 :
            inf, sup = workingIndex[0], workingIndex[-1]
            candidates = indexPeaks[(inf <= indexPeaks) * (indexPeaks <= sup)]
            ampsCandidates = amplitudes[candidates]

            if len(candidates) > 0 :
                nbHarmo += 1

                #getting the index of the largest candidate peak
                listIndHarmo.append(np.int(
                    candidates[ampsCandidates == np.max(ampsCandidates)]))

            else : flag = False

        else : flag = False
            # here, our predicted frequency range is higher frequency than out spectrum.

    listIndHarmo = np.array(listIndHarmo)
    score, certainty = get_harmonic_score(freqs, amplitudes, listIndHarmo,
                                          freqs[indexBpfoPeak], freqRangeHarmo,
                                          harmo_cert_threshold=harmo_cert_threshold)
    # print('score: '+str(score) + ', harm: '+str(len(listIndHarmo)))
    if score == 0:
        nbHarmo = 0
    return(nbHarmo, listIndHarmo, score, certainty)
```

```

def SNR(amps, indexSignal) :
    signal = np.sum(amps[indexSignal] ** 2)

    noise = np.sum(amps ** 2) - signal

    snr = signal / noise
    return(snr)


def get_distance_to_next_higher_point(freqs, amps, peak_index):
    """
    Calculate the smallest distance to a higher amplitude in the spectrum.
    For a true harmonic signal, this distance should always be approximately
    the BPFO frequency, because the next higher peak should be left or right
    to the considered peak.
    :return:
    index, distance (float)
    """

    # find all indices of points that are higher than the considered peak
    high_indices = np.asarray(
        [index for index in range(len(amps)) if amps[index] > amps[peak_index]])
    if len(high_indices):
        min_index = min(high_indices, key=lambda i: abs(i-peak_index))
        return min_index, abs(freqs[peak_index] - freqs[min_index])
    else:
        return 0, float(’inf’)


def get_harmonic_score(freqs, amps, listIndHarmo, bpfo_freq, freqRangeHarmo=6,
                       harmo_cert_threshold=0.1):
    """
    Calculates the harmonic score, i.e. the sum of all peaks whose distance to the
    closest higher value is approximately the BPFO frequency
    :param amps: spectrum
    :param listIndHarmo: List of suspected harmonic peaks
    :param bpfo_freq: Suspected BPFO frequency
    :return: score, harmo_certainty
    """
    score = 0
```

```python
    num_scorers = 0
    for peak_index in listIndHarmo:
        _ , distance = get_distance_to_next_higher_point(freqs, amps, peak_index)
        if bpfo_freq - freqRangeHarmo/2 <= distance \
                and bpfo_freq + freqRangeHarmo/2 >= distance:
            num_scorers += 1
            score += amps[peak_index]
    if score > 0:
        harmo_certainty = num_scorers/len(listIndHarmo)
    else:
        harmo_certainty = 0
    if harmo_certainty <= harmo_cert_threshold:
        score = 0
    return score, harmo_certainty
```

For the peak finding algorithm, the objects of the class `Measurement` are needed. This class is defined below.

```python
"""
Measurement class file

This module specifies the Measurement class, which is a standardized class for all
analytics modules.

Developed by Team Kurfess/Saldana at Georgia Institute of Technology in 2018 -- 2020.

For inquiries, contact Fabia Bayer fbayer6@gatech.edu.
"""

import numpy as np
from typing import Tuple
import awesome_vibration_toolbox.data_prep.FFTHelper as FFT
import scipy.fftpack

class Measurement(object):
    """
    A class used for standardized storage of vibration measurement information.

    All attributes can only be accessed by their get() methods and can only be
    changed by the update methods.

    Measurement is characterized by an identification (ID).
    If no ID is given, a numeric ID increments automatically with updating
    the waveform.

    Methods:
    --------
     - get_sampling_rate(self)
     - get_waveform(self)
     - get_ID(self)
     - get_rpm(self)
     - get_FFT(self, type)

       returns the FFT frequencies and amplitudes.
       Allowed types:
         - ’acc’
         - ’vel’
         - ’density’
         - ’raw’
         - ’raw_vel’
     - update_waveform(self, new_waveform, new_ID)
     - update_bearing(self, alpha, n) todo not implemented yet


    Attributes:
    ----------
     - sampling_rate: float
     - waveform: np.ndarray
     - ID: object
     - rpm: float
     - freqs: np.ndarray
     - spectral_density: np.ndarray
     - acc_amps: np.ndarray
     - vel_freqs: np.ndarray
     - vel_amps: np.ndarray
     - raw_freqs: np.ndarray
     - raw_amps: np.ndarray
     - raw_vel_freqs: np.ndarray
     - raw_vels: np.ndarray
     - max_resolution: Maximal accepted frequency spacing between two spectrum samples
    """


    def __init__(self, sampling_rate: float, waveform: np.ndarray, ID: object = 0,
                 rpm: float = None):
        self.__sampling_rate = sampling_rate
        self.__waveform = waveform
        self.__ID = ID
        self.__rpm = rpm
        self.__freqs = None
        self.__vel_freqs = None
        self.__acc_amps = None
        self.__vel_amps = None
        self.__spectral_density = None
        self.__bearing_info = None
        self.__delta_freq = None
        self.__max_resolution = None
        self.__raw_freqs = None
        self.__raw_amps = None
        self.__raw_vel_freqs = None
        self.__raw_vels = None

    # FFT calculation and getter method
    def get_FFT(self, spectrum_type: str=’acc’, max_resolution: float = 0.2) \
            -> Tuple[np.ndarray, np.ndarray]:
        """
        FFT getter and calculation.

        Calculates all necessary FFT information when they are needed for the
        first time.

        Allowed inputs for spectrum_type:
         -- ’acc’:      Acceleration spectrum

         -- ’vel’:      Velocity spectrum
         -- ’density’:  Spectral density
         -- ’raw’:      Raw FFT without Welch’s method and any windowing (acceleration).
                        May yield clearer peaks, but more noise.
         -- ’raw_vel’:  Raw FFT without Welch’s method and any windowing, integrated to
                        velocity.
        :param spectrum_type: desired spectrum.
        :param max_resolution: Desired acceptable resolution (interval between
                               two samples) of the resulting spectrum.
                               Keep in mind the trade-off between de-noising
                               and resolution.
        :return: (np.ndarray, np.ndarray) freqs, amps - desired spectrum and frequencies
        """
        if spectrum_type not in ['acc', 'vel', 'density', 'raw', 'raw_vel']:
            raise ValueError('"'+str(spectrum_type)+'" is not an allowed FFT type.\n'+
                             'Allowed types: "acc", "vel", "density", "raw", "raw_vel".')

        elif 'raw' in spectrum_type:
            # compute the raw FFT if called for the first time
            if self.__raw_amps is None:
                cplx_fft = scipy.fftpack.fft(self.get_waveform())
                self.__raw_amps = np.abs(cplx_fft)[:int(np.ceil(len(cplx_fft)/2))]
                # Scaling for physical meaning
                self.__raw_freqs = np.array(range(len(self.__raw_amps)))\
                                   * self.get_sampling_rate()/2/len(self.__raw_amps)

            if 'vel' in spectrum_type:
                if self.__raw_vels is None:
                    # discard the two lowest entries because division yields
                    # too high values
                    self.__raw_vel_freqs = self.__raw_freqs[2:]
                    self.__raw_vels = np.divide(self.__raw_amps[2:], self.__raw_vel_freqs)
                return self.__raw_vel_freqs, self.__raw_vels

            else:
                return self.__raw_freqs, self.__raw_amps

        else:
            # compute the FFT if this is called the first time
            if self.__freqs is None or self.__max_resolution != max_resolution:
                self.__max_resolution = max_resolution
                self.__acc_amps = None
                self.__vel_amps = None
                self.__freqs, self.__spectral_density = \
                    FFT.generateFFT(self.__sampling_rate, self.__waveform,
                                    minResolution=max_resolution,
                                    pointsBetween=1, padded=True)
            if spectrum_type == 'density':
                return self.__freqs, self.__spectral_density


            # Compute the acceleration spectrum if acceleration or
            # velocity spectrum is needed
            if self.__acc_amps is None:
                self.__acc_amps = \
                    FFT.get_spectrum_from_density(self.__sampling_rate,
                                                  self.__spectral_density,
                                                  self.__waveform, normalize=True)
            if spectrum_type == 'acc':
                return self.__freqs, self.__acc_amps

            # Compute velocity spectrum if needed
            if self.__vel_amps is None:
                self.__vel_freqs, self.__vel_amps = \
                    FFT.get_integrated_spectrum(self.__freqs, self.__acc_amps,
                                                convertUnits=True)
            return self.__vel_freqs, self.__vel_amps


    def update_waveform(self, new_waveform: np.ndarray, new_ID = None,
                        new_samplingrate: float = None):
        """
        Update the waveform while keeping all other parameters constant.
        All FFT info is deleted and recalculated at the next use.
        Numerical ID is incremented if no new ID is given.

        :param new_waveform: Waveform to be updated
        :param new_ID: Updated ID. Increments automatically if no ID is given and
                       previous ID is numeric.
        :param new_samplingrate: In case the sampling rate changes, new sampling rate
                                 can be added (It is recommended to create a new
                                 Measurement object instead).
        """
        self.__waveform = new_waveform
        self.__acc_amps = None
        self.__vel_amps = None
        self.__freqs = None
        self.__vel_freqs = None
        self.__spectral_density = None
        if new_ID is None:
            if type(self.__ID) is int:
                self.__ID += 1
            else:
                raise ValueError("ID is not numeric and ID update can not be computed "
                                 + "automatically. Please specify a new ID.")
        else:
            self.__ID = new_ID

        if new_samplingrate is not None:
            self.__sampling_rate = new_samplingrate

    def update_bearing_info(self, alpha: float, number_balls: int):
        """
        Calculates bearing information and stores it in the attribute.
        todo.
        :param alpha:
        :param number_balls:
        :return:
        """
        pass

    # Trivial Getter Methods.
    def get_sampling_rate(self) -> float:
        return self.__sampling_rate

    def get_waveform(self) -> np.ndarray:
        return self.__waveform

    def get_ID(self):
        return self.__ID

    def get_rpm(self) -> float:
        return self.__rpm
```

A.2 Simulated Signal Generation

The algorithm for the generation of simulated signals is presented here.

```
import numpy as np
# import matplotlib.pyplot as plt
from typing import Union, List, Tuple
from awesome_vibration_toolbox.data_prep.Measurement import Measurement
import scipy.signal as ss
import math
from awesome_vibration_toolbox.data_prep.FFTHelper import generateFFT

class Signal_Generator_Object(object):
    """
     TODO documentation.....
    """
    def __init__(self, sampling_freqs: List[int], freq_res: float = 0.25,
                 fault_type: str = "bpfo", antialiasing: bool = True, **kwargs):
        """

        The base signal must be the least common multiple of all desired sampling rates
        and must also be bigger than 20000.
        """
        self.antialiasing = antialiasing

        # compute least common multiple of all given sampling freqs to be able to
        # downsample as desired. Taken from
        # https://stackoverflow.com/questions/37237954/
        # calculate-the-lcm-of-a-list-of-given-numbers-in-python
        self.samplingrate_high = sampling_freqs[0]
        for freq in sampling_freqs[1:]:
            self.samplingrate_high = \
                self.samplingrate_high * freq // math.gcd(self.samplingrate_high, freq)
        # ensure that the sampling rate is > 40000 to mitigate as much aliasing
        # as possible
        self.samplingrate_high = \
            self.samplingrate_high * (40000 // self.samplingrate_high + 1)
        N = self.samplingrate_high/2/freq_res

        if not N.is_integer():
            raise ValueError("N is "+str(N)+". "+
                            "Some debugging necessary or frequency resolution " +
                            "chosen stupidly :)")
        N = int(N)

        # we have now N, self.samplingrate_high, freq_resolution.
        # distinguish the cases
        if fault_type == "bpfo":
            bpfo_args = {’N’: N, ’samplingrate’: self.samplingrate_high}
            # analyze the kwargs
            try:
                bpfo_args[’bpfo_freq’] = kwargs[’bpfo_freq’]
            except KeyError:
                raise ValueError("No BPFO frequency specified although case "
                                 +fault_type+" was given.")
            for add_param in [’bearing_freq’, ’amplitude’, ’decay’]:
                # These are the optional parameters
                try:
                    bpfo_args[add_param] = kwargs[add_param]
                except KeyError:
                    pass

            # generate bpfo signal
            self.base_sig, _ = perfect_bpfo_signal(**bpfo_args)

        elif fault_type == "shaft":
            shaft_args = {’N’: N, ’samplingrate’: self.samplingrate_high}
            # analyze the kwargs
            try:
                shaft_args[’shaft_freq’] = kwargs[’shaft_freq’]
            except KeyError:
                raise ValueError("No shaft frequency specified although case "
                                 + fault_type + " was given.")
            for add_param in [’amplitude’, ’exponential_harms’, ’harm_max’]:
                # These are the optional parameters
                try:
                    shaft_args[add_param] = kwargs[add_param]
                except KeyError:
                    pass

            # generate shaft signal
            self.base_sig, _ = shaft_rotation_component(**shaft_args)

        elif fault_type == "gear":
            gear_args = {’N’: N, ’samplingrate’: self.samplingrate_high}
            # analyze the kwargs
            try:
                gear_args[’gear_freq’] = kwargs[’gear_freq’]
            except KeyError:
                raise ValueError("No gear frequency specified although case "
                                 + fault_type + " was given.")
            for add_param in [’num_teeth’, ’gear_res_freq’, ’lambda_g’, ’gearmesh_amps’]:
                # These are the optional parameters
                try:
                    gear_args[add_param] = kwargs[add_param]
                except KeyError:
                    pass

            # generate shaft signal
            self.base_sig, _ = gear_component(**gear_args)
        else:
            raise ValueError("Signal type "+fault_type+
                             "undefinded. Must be in [’bpfo’, ’shaft’, ’gear’].")

        self.downsampled_sigs = {}

    def get_signal(self, samplingrate, energy):
        """
        Return a signal with the right amplitude and downsampled to the
        right sampling rate.
        """

        if samplingrate not in self.downsampled_sigs.keys():
            mult_factor = self.samplingrate_high / samplingrate
            if not mult_factor.is_integer():
                raise ValueError("The sampling rate "+str(samplingrate) +
                                 " is not supported by base signal with sampling rate "
                                 +str(self.samplingrate_high))
            mult_factor = int(mult_factor)
            if self.antialiasing:
                base_sig = ss.decimate(self.base_sig, mult_factor, ftype=’fir’)
            else:
                base_sig = self.base_sig[::mult_factor]
            energy_base = np.sum(np.square(base_sig))
            self.downsampled_sigs[samplingrate] = (base_sig, energy_base)
        else:
            base_sig, energy_base = self.downsampled_sigs[samplingrate]
        res = base_sig / np.sqrt(energy_base) * np.sqrt(energy)
        # plt.plot(res)
        # plt.show()
        # print(’bla’)
        return res
```

```python


def initialize_mixed_signals(samplingrates: List[int], freq_resolution: float,
                             antialiasing: bool = True, bpfo_freq: float = 128,
                             shaft_freq: float = 30, gear_freq: float = 30):
    """
    Create the signal generator objects
    """

    bpfo_gen = Signal_Generator_Object(sampling_freqs=samplingrates,
                                       freq_res=freq_resolution,
                                       fault_type=’bpfo’, antialiasing=True,
                                       bpfo_freq=bpfo_freq)
    shaft_gen = Signal_Generator_Object(sampling_freqs=samplingrates,
                                        freq_res=freq_resolution,
                                        fault_type=’shaft’, antialiasing=True,
                                        shaft_freq=shaft_freq)
    gear_gen = Signal_Generator_Object(sampling_freqs=samplingrates,
                                       freq_res=freq_resolution,
                                       fault_type=’gear’, antialiasing=True,
                                       gear_freq=gear_freq)

    return [bpfo_gen, shaft_gen, gear_gen]


def get_mixed_signal_2(generator_objects: List[Signal_Generator_Object], shaft_en,
                       bpfo_en, gear_en, noise_en, samplingrate):
    """
    Return a signal sith the desired signal energies.
    """
    bpfo_sig = generator_objects[0].get_signal(samplingrate, bpfo_en)
    shaft_sig = generator_objects[1].get_signal(samplingrate, shaft_en)
    gear_sig = generator_objects[2].get_signal(samplingrate, gear_en)
    noise, noise_energy = noise_signal(N=len(bpfo_sig))
    noise = noise / noise_energy*noise_en

    signal = bpfo_sig + shaft_sig + gear_sig + noise

    measurement = Measurement(samplingrate, signal,
                              ID=str(shaft_en) + "-" + str(bpfo_en) + "-" +
                                 str(gear_en) + "-" + str(noise_en))
    return signal, measurement


def perfect_bpfo_signal(bpfo_freq: float = 128/2, bearing_freq: float = 4500,
                        samplingrate: float = 20000, N: int = 20000,
                        amplitude: float = 0.01, decay: float = 2000) \
        -> Tuple[np.ndarray, float]:
    """
    Return a perfect bpfo signal from the corresponding equation.

    :param bpfo_freq: desired BPPFO frequency.
    :param bearing_freq: Desired bearing eigen frequency
    :param samplingrate: Desired signal sampling rate
    :param amplitude: Desired maximal defect amplitude
    :param decay: desired exponential decay factor
    :return: np.ndarray of signal values.
    """

    period_bpfo = 1/bpfo_freq

    def cont_signal_fcn(t):
        """
        Calculate a periodically exponentially decaying signal at time t.
        :param t: time instant
        :return: Value of the signal at time t
        """
        slip = 0.98 + 0.04*np.random.rand()
        # slip = 1
        t_in_period = t % period_bpfo * slip
        carrier = np.cos(2*np.pi*bearing_freq*t_in_period)
        envelope = amplitude*np.exp(-t_in_period*decay)
        return carrier * envelope

    ts = np.array(range(N))/samplingrate
    signal = np.array([cont_signal_fcn(t) for t in ts])
    signal_energy = float(np.sum(np.square(signal)))

    return signal, signal_energy


def shaft_rotation_component (shaft_freq, samplingrate: float = 20000, N: int = 20000,
                              amplitude: Union[List[float], float] = 1,
                              exponential_harms: Union[None, float] = -1,
                              harm_max: int = 6) -> Tuple[np.ndarray, float]:
    """
    Create a periodic shaft rotation signal. Represented by a Taylor series
    (or at least the first few terms) at the shaft freq and its harmonics.
    :param shaft_freq: Shaft frequency. RPM/60.
    :param amplitude: Base parameter or list of parameters
    :param exponential_harms: If only one amplitude is given, exponential decay
                              of amplitudes is assumed. This parameter specifies
    :param harm_max: If harmonics are constructed exponentially,
                     it stops after this harmonic.
    :return:
    """
    # [28] proposes this shaft representation; they have amplitude_1 = 0.8 (Or 0.7)
    # and B_2 = 0.2, the rest 0

    if exponential_harms:
        try:
            # Making sure that we are considered with the right type
            amplitude = amplitude[0]
            raise ValueError("If exponential_harms is specified, " +
                             "amplitude must be a float.")
        except TypeError:
            amplitude = [amplitude * np.exp(k * exponential_harms)
                         for k in range(harm_max + 1)]

    phases = 2*np.pi*np.random.rand(*np.shape(amplitude))

    def cont_signal_fcn(t):
        """
        Calculate a periodically exponentially decaying signal at time t.
            :param t: time instant
            :return: Value of the signal at time t

        :param t: time instant
        :return: Value of the signal at time t
        """
        signal = 0
        for k, a in enumerate(amplitude):
            signal += a*np.cos(2*np.pi*(k+1)*shaft_freq*t + phases[k])
        return signal

    ts = np.array(range(N)) / samplingrate
    signal = np.array([cont_signal_fcn(t) for t in ts])
    signal_energy = float(np.sum(np.square(signal)))

    return signal, signal_energy


def gear_component(gear_freq: float, num_teeth: int = 12, gear_res_freq: float = 2500,
                   lambda_g: float = 0.05, samplingrate: float = 20000, N: int = 20000,
                   gearmesh_amps: Union[None, List[float]]=None) \
        -> Tuple[np.ndarray, float]:

    if gearmesh_amps is None:
        gearmesh_amps = [0.5, 5, 1]

    G = len(gearmesh_amps)

    def cont_signal_fcn(t):
        """
        Calculate a gearmesh signal at time t.
        Source: Bearing fault diagnosis under unknown variable speed via
        gear noise cancellation and rotational order
        sideband identification, Wang et al., 2015

        :param t: time instant
        :return: Value of the signal at time t
        """

        # Resonance response of gear
        signal = (1 + np.cos(2*np.pi*gear_freq*t))*np.cos(2*np.pi*gear_res_freq*t)
        for j in range(G):
            # Meshing and multiples
            signal += lambda_g * gearmesh_amps[j] \
                      * np.cos(2*np.pi*(j+1)*num_teeth*gear_freq*t)
        return signal
    ts = np.array(range(N))/samplingrate
    signal = np.array([cont_signal_fcn(t) for t in ts])
    signal_energy = float(np.sum(np.square(signal)))

```

```python
    return signal, signal_energy


def noise_signal(mu: float = 0, sigma: float = 1, N: int = 20000) \
        -> Tuple[np.ndarray, float]:
    """
    Add Gaussian white noise to a signal so that the desired SNR is obtained.
    :param mu: Mean of gaussian signal from which the noise is drawn
    :param sigma: standard deviation of the gaussian noise
    :param N: length of desired signal
    :return: The signal, enhanced with noise, and its energy
    """
    noise = np.random.normal(mu, sigma, N)
    noise_energy = float(np.sum(np.square(noise)))

    return noise, noise_energy


if __name__ == '__main__':
    # N = 200000
    # samplingrate = 2000000


    samplingrate = 81920
    freq_res = 0.25
    list_of_generators = \
        initialize_mixed_signals([512, 1024, 2048, 4096, 8192, 10240, 16384,
                                  20480, 40960, 81920],
                                 freq_resolution=freq_res)

    perfect = get_mixed_signal_2(list_of_generators, 0, 1, 0, 0,
                                 samplingrate=samplingrate)
```

APPENDIX B

PERFORMANCE METRIC RESULTS

B.1 Simulated Data

As introduced in Section 4, the simulated data is partitioned into different noise categories. The performance metric results for the considered sampling rates and noise categories are given in this section.

B.1.1 Low Noise

Table B.1: Simulated confusion matrices for a sampling rate of 4.096 kHz and low noise.

Flag raised	Defected signal	Healthy signal
peak finding: True	7	6
peak finding: False	68	68
Envelope: True	33	23
Envelope: False	42	51
TFCE: True	52	21
TFCE: False	23	53

Table B.2: Simulated performance metrics for the three approaches and a sampling rate of 4.096 kHz and low noise.

Approach	Precision [%]	Recall [%]	F1-Score [%]
peak finding	53.85	9.33	15.91
Envelope	58.93	44.0	50.38
TFCE	71.23	69.33	70.27

Table B.3: Simulated confusion matrices for a sampling rate of 6.144 kHz and low noise.

Flag raised	Defected signal	Healthy signal
peak finding: True	1	2
peak finding: False	74	72
Envelope: True	56	16
Envelope: False	19	58
TFCE: True	72	37
TFCE: False	3	37

Table B.4: Simulated performance metrics for the three approaches and a sampling rate of 6.144 kHz and low noise.

Approach	Precision [%]	Recall [%]	F1-Score [%]
peak finding	33.33	1.33	2.56
Envelope	77.78	74.67	76.19
TFCE	66.06	96.00	78.26

Table B.5: Simulated confusion matrices for a sampling rate of 10.420 kHz and low noise.

Flag raised	Defected signal	Healthy signal
peak finding: True	0	1
peak finding: False	75	73
Envelope: True	75	8
Envelope: False	0	66
TFCE: True	64	38
TFCE: False	11	36

Table B.6: Simulated performance metrics for the three approaches and a sampling rate of 10.240 kHz and low noise.

Approach	Precision [%]	Recall [%]	F1-Score [%]
peak finding	0	0	N/A
Envelope	90.36	100.0	94.94
TFCE	62.75	85.33	72.23

Table B.7: Simulated confusion matrices for a sampling rate of 20.480 kHz and low noise.

Flag raised	Defected signal	Healthy signal
peak finding: True	0	0
peak finding: False	75	74
Envelope: True	75	2
Envelope: False	0	72
TFCE: True	73	54
TFCE: False	2	20

Table B.8: Simulated performance metrics for the three approaches and a sampling rate of 20.480 kHz and low noise.

Approach	Precision [%]	Recall [%]	F1-Score [%]
peak finding	N/A	0	N/A
Envelope	97.40	100.0	98.68
TFCE	57.48	97.33	72.28

B.1.2 Medium Noise

Table B.9: Simulated confusion matrices for a sampling rate of 4.096 kHz and medium noise.

Flag raised	Defected signal	Healthy signal
peak finding: True	7	2
peak finding: False	68	73
Envelope: True	41	35
Envelope: False	34	40
TFCE: True	17	16
TFCE: False	58	59

Table B.10: Simulated performance metrics for the three approaches and a sampling rate of 4.096 kHz and medium noise.

Approach	Precision [%]	Recall [%]	F1-Score [%]
peak finding	77.78	9.33	16.67
Envelope	53.95	54.67	54.30
TFCE	51.52	22.67	31.48

Table B.11: Simulated confusion matrices for a sampling rate of 6.144 kHz and medium noise.

Flag raised	Defected signal	Healthy signal
peak finding: True	6	8
peak finding: False	69	67
Envelope: True	26	17
Envelope: False	49	58
TFCE: True	49	36
TFCE: False	26	39

Table B.12: Simulated performance metrics for the three approaches and a sampling rate of 6.144 kHz and medium noise.

Approach	Precision [%]	Recall [%]	F1-Score [%]
peak finding	42.86	8.00	13.48
Envelope	60.47	34.67	44.07
TFCE	57.65	65.33	61.25

Table B.13: Simulated confusion matrices for a sampling rate of 10.420 kHz and medium noise.

Flag raised	Defected signal	Healthy signal
peak finding: True	4	9
peak finding: False	71	66
Envelope: True	23	12
Envelope: False	52	63
TFCE: True	40	37
TFCE: False	35	38

Table B.14: Simulated performance metrics for the three approaches and a sampling rate of 10.240 kHz and medium noise.

Approach	Precision [%]	Recall [%]	F1-Score [%]
peak finding	30.77	5.33	9.09
Envelope	65.71	30.67	41.82
TFCE	51.95	53.33	52.63

Table B.15: Simulated confusion matrices for a sampling rate of 20.480 kHz and medium noise.

Flag raised	Defected signal	Healthy signal
peak finding: True	4	3
peak finding: False	71	72
Envelope: True	75	0
Envelope: False	0	75
TFCE: True	68	24
TFCE: False	7	51

Table B.16: Simulated performance metrics for the three approaches and a sampling rate of 20.480 kHz and medium noise.

Approach	Precision [%]	Recall [%]	F1-Score [%]
peak finding	57.14	5.33	9.76
Envelope	100.0	100.0	100.0
TFCE	73.91	90.76	81.44

B.1.3 High Noise

Table B.17: Simulated confusion matrices for a sampling rate of 4.096 kHz and high noise.

Flag raised	Defected signal	Healthy signal
peak finding: True	0	0
peak finding: False	60	60
Envelope: True	31	33
Envelope: False	29	27
TFCE: True	11	8
TFCE: False	49	52

Table B.18: Simulated performance metrics for the three approaches and a sampling rate of 4.096 kHz and high noise.

Approach	Precision [%]	Recall [%]	F1-Score [%]
peak finding	N/A	0	N/A
Envelope	48.44	51.67	50.0
TFCE	57.89	18.33	27.85

Table B.19: Simulated confusion matrices for a sampling rate of 6.144 kHz and high noise.

Flag raised	Defected signal	Healthy signal
peak finding: True	0	1
peak finding: False	60	59
Envelope: True	18	12
Envelope: False	42	48
TFCE: True	34	32
TFCE: False	26	28

Table B.20: Simulated performance metrics for the three approaches and a sampling rate of 6.144 kHz and high noise.

Approach	Precision [%]	Recall [%]	F1-Score [%]
peak finding	0	0	N/A
Envelope	60.00	30.00	40.00
TFCE	51.52	56.67	53.97

Table B.21: Simulated confusion matrices for a sampling rate of 10.420 kHz and high noise.

Flag raised	Defected signal	Healthy signal
peak finding: True	4	4
peak finding: False	56	56
Envelope: True	13	6
Envelope: False	47	54
TFCE: True	41	30
TFCE: False	19	30

Table B.22: Simulated performance metrics for the three approaches and a sampling rate of 10.240 kHz and high noise.

Approach	Precision [%]	Recall [%]	F1-Score [%]
peak finding	50.00	6.67	11.76
Envelope	68.42	21.67	32.91
TFCE	57.75	68.33	62.60

Table B.23: Simulated confusion matrices for a sampling rate of 20.480 kHz and high noise.

Flag raised	Defected signal	Healthy signal
peak finding: True	0	3
peak finding: False	60	57
Envelope: True	28	1
Envelope: False	32	59
TFCE: True	41	37
TFCE: False	19	23

Table B.24: Simulated performance metrics for the three approaches and a sampling rate of 20.480 kHz and high noise.

Approach	Precision [%]	Recall [%]	F1-Score [%]
peak finding	0	0	N/A
Envelope	96.55	46.67	62.92
TFCE	52.56	68.33	59.42

B.2 Experimental Data [39]

All datasets are curated by taking 100 signals each from the beginning of the measurement process (healthy) and 100 signals from near the end of the measurement process (defected).

Table B.25: Experimental data [39] confusion matrices for a sampling rate of 20 kHz.

Flag raised	Defected signal	Healthy signal
peak finding: True	4	0
peak finding: False	96	100
Envelope: True	94	10
Envelope: False	6	90
TFCE: True	98	1
TFCE: False	2	99

Table B.26: Experimental data [39] performance metrics for a sampling rate of 20 kHz.

Approach	Precision [%]	Recall [%]	F1-Score [%]
peak finding	100	4.0	7.69
Envelope	90.38	94.0	92.16
TFCE	98.99	98.0	98.49

Table B.27: Experimental data [39] confusion matrices for a sampling rate of 10 kHz.

Flag raised	Defected signal	Healthy signal
peak finding: True	6	0
peak finding: False	94	100
Envelope: True	5	29
Envelope: False	95	71
TFCE: True	91	12
TFCE: False	9	88

Table B.28: Experimental data [39] performance metrics for a sampling rate of 10 kHz.

Approach	Precision [%]	Recall [%]	F1-Score [%]
peak finding	100	6.0	11.32
Envelope	14.71	5.0	7.46
TFCE	88.35	91.0	89.66

Table B.29: Experimental data [39] confusion matrices for a sampling rate of 5 kHz.

Flag raised	Defected signal	Healthy signal
peak finding: True	8	0
peak finding: False	92	100
Envelope: True	19	25
Envelope: False	81	75
TFCE: True	72	10
TFCE: False	28	90

Table B.30: Experimental data [39] performance metrics for a sampling rate of 5 kHz.

Approach	Precision [%]	Recall [%]	F1-Score [%]
peak finding	100	8.0	14.81
Envelope	43.18	19.0	26.39
TFCE	87.80	72.0	79.12

Table B.31: Experimental data [39] confusion matrices for a sampling rate of 4 kHz.

Flag raised	Defected signal	Healthy signal
peak finding: True	8	0
peak finding: False	92	100
Envelope: True	32	25
Envelope: False	68	75
TFCE: True	99	7
TFCE: False	1	93

Table B.32: Experimental data [39] performance metrics for a sampling rate of 4 kHz.

Approach	Precision [%]	Recall [%]	F1-Score [%]
peak finding	100	8.0	14.81
Envelope	56.14	32.0	40.76
TFCE	42.42	28.0	96.12

Table B.33: Experimental data [39] confusion matrices for a sampling rate of 2 kHz.

Flag raised	Defected signal	Healthy signal
peak finding: True	0	7
peak finding: False	100	93
Envelope: True	28	38
Envelope: False	72	62
TFCE: True	41	0
TFCE: False	59	100

Table B.34: Experimental data [39] performance metrics for a sampling rate of 2 kHz.

Approach	Precision [%]	Recall [%]	F1-Score [%]
peak finding	0.0	0.0	N/A
Envelope	42.42	28.0	33.73
TFCE	100	41.0	58.16

APPENDIX C

PERFORMANCE WITH EXPERIMENTAL DATA UNDER VARIOUS SAMPLING RATES

In this section, graphs indicating the performance of the three approaches under various sampling rates are listed.

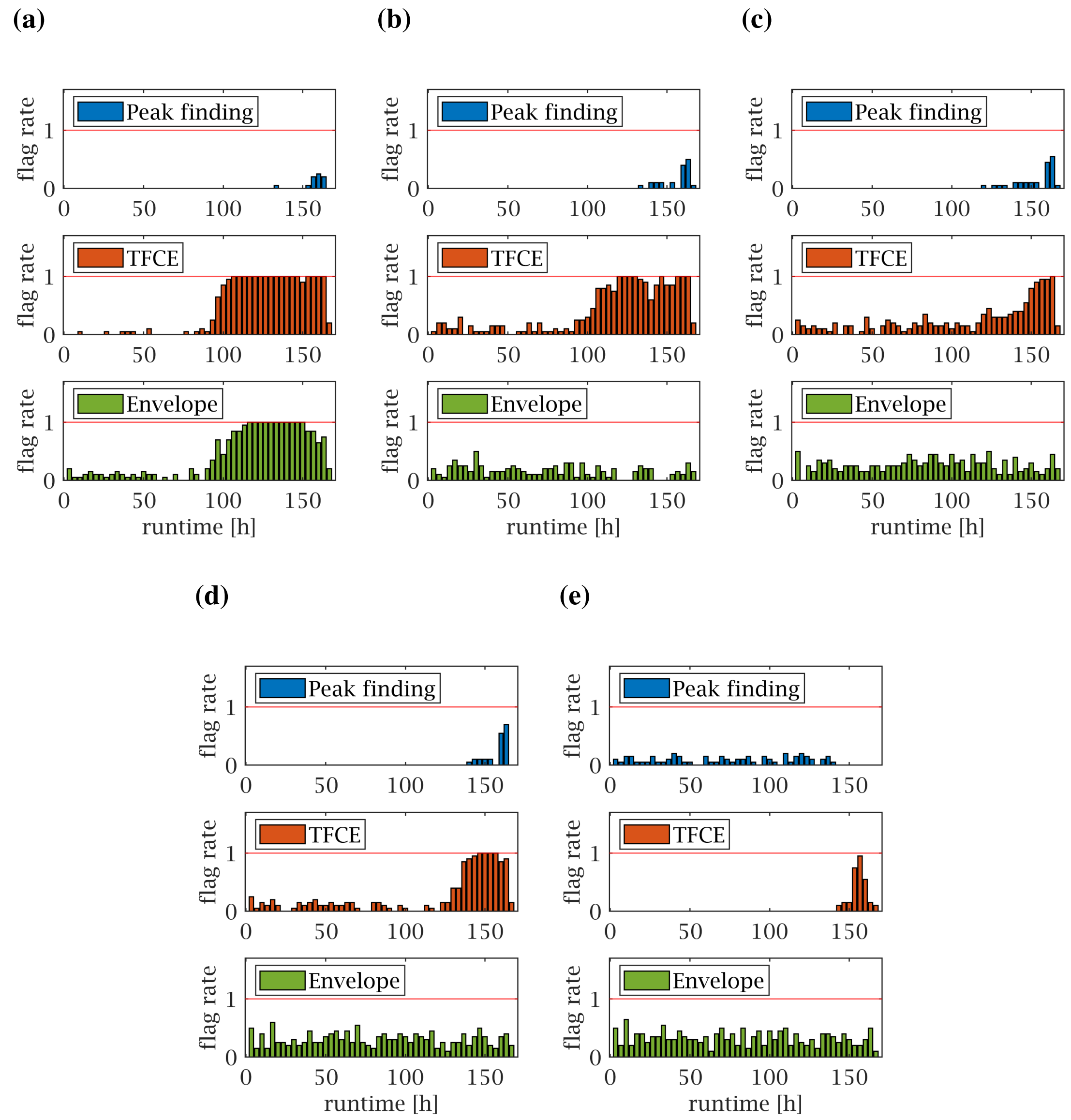

Figure C.1: Flag rate progression for experimental data [39] and various sampling rates: **(a)** 20 kHz, **(b)** 10 kHz, **(c)** 5 kHz, **(d)** 4 kHz, **(e)** 2 kHz.

REFERENCES

[1] J. Piotrowski, *Shaft Alignment Handbook*, 3rd ed. CRC Press, 2006.

[2] R. Ahmad and S. Kamaruddin, “An overview of time-based and condition-based maintenance in industrial application,” *Computers and Industrial Engineering*, 2012.

[3] R. B. Randall, *Vibration-based Condition Monitoring: Industrial, aerospace and automotive applications*. Wiley and Sons, 2011.

[4] A. K. Jardine, D. Lin, and D. Banjevic, “A review on machinery diagnostics and prognostics implementing condition-based maintenance,” *Mechanical systems and signal processing*, vol. 20, no. 7, pp. 1483–1510, 2006.

[5] H. P. Bloch and F. K. Geitner, “Practical machinery management for process plants. volume 2: Machinery failure analysis and troubleshooting,” Exxon Chemical Co., Baytown, TX, Tech. Rep., 1983.

[6] T. J. Geiger, “13 - machinery vibration monitoring sensors,” in *Sensor Technology Handbook*, J. S. Wilson, Ed., Burlington: Newnes, 2005, pp. 285 –306.

[7] W. Yang and P. J. Tavner, “Empirical mode decomposition, an adaptive approach for interpreting shaft vibratory signals of large rotating machinery,” *Journal of Sound and Vibration*, 2009.

[8] C. J. Li and J. Ma, “Wavelet decomposition of vibrations for detection of bearing-localized defects,” *NDT and E International*, vol. 30, no. 3, pp. 143 –149, 1997.

[9] J. Lee, F. Wu, W. Zhao, M. Ghaffari, L. Liao, and D. Siegel, “Prognostics and health management design for rotary machinery systems - reviews, methodology and applications,” *Mechanical Systems and Signal Processing*, 2014.

[10] G. D. Hongchao Wang Jin Chen, “Feature extraction of rolling bearing’s early weak fault based on eemd and tunable q-factor wavelet transform,” *Mechanical Systems and Signal Processing*, 2014.

[11] B. Künne, “Wälzlager,” in *Einführung in die Maschinenelemente: Gestaltung — Berechnung — Konstruktion*. Wiesbaden: Vieweg+Teubner Verlag, 2001, pp. 147–166.

[12] G. Gautier, R. Serra, and J.-M. Mencik, “Roller bearing monitoring by new subspace-based damage indicator,” *Shock and Vibration*, vol. 2015, Aug. 2015, Figure License: CC BY 3.0.

[13] M. P. Boyce, “19 - control systems and instrumentation,” in *Gas Turbine Engineering Handbook (Third Edition)*, M. P. Boyce, Ed., Third Edition, Burlington: Gulf Professional Publishing, 2006, pp. 664 –720.

[14] C. Aszkler, “5 - acceleration, shock and vibration sensors,” in *Sensor Technology Handbook*, J. S. Wilson, Ed., Burlington: Newnes, 2005, pp. 137 –159.

[15] S. Kosonocky and P. Xiao, “Analog-to-digial conversion architectures,” in *The Digital Signal Processing Handbook*, CRC Press, 2010.

[16] W. K. Jenkins, “Fourier methods for signal analysis and processing,” in *The Digital Signal Processing Handbook*, CRC Press, 2010.

[17] Wikimedia Commons, *Aliasing.svg*, https://commons.wikimedia.org/wiki/, File:Aliasing_mrtz.svg, Access date: 2020-01-17. License: CC BY-SA 2.5, 2004.

[18] O. Alkin, *Signals and Systems: A MATLAB Integrated Approach*. CRC press, 2015.

[19] H. Qiu, J. Lee, J. Lin, and G. Yu, “Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics,” *Journal of Sound and Vibration*, 2006.

[20] W. Jiang, S. K. Spurgeon, J. A. Twiddle, F. S. Schlindwein, Y. Feng, and S. Thanagasundram, “A wavelet cluster-based band-pass filtering and envelope demodulation approach with application to fault diagnosis in a dry vacuum pump,” *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 2007.

[21] M. Feldman, “Hilbert transform in vibration analysis,” *Mechanical Systems and Signal Processing*, 2010.

[22] T. Olson, *Applied Fourier Analysis From Signal Processing to Medical Imaging*. Springer New York, 2017.

[23] M. Feldman, *Hilbert Transform Applications in Mechanical Vibration*. John Wiley & Sons, 2011.

[24] J. Duoandikoetxea Zuazo, *Fourier analysis*, ser. Graduate Studies in Mathematics, v. 29. Providence, R.I.: American Mathematical Society, 2001, ISBN: 9781470411459.

[25] M. Swamynathan, *Mastering Machine Learning with Python in Six Steps: A Practical Implementation Guide to Predictive Data Analytics Using Python*. O'Reilly Media, 2019.

[26] A. C. Müller and S. Guido, *Introduction to machine learning with Python : a guide for data scientists*. O'Reilly Media, 2016, ISBN: 1-4493-6988-X.

[27] H. Martin and F. Honarvar, "Application of statistical moments to bearing failure detection," *Applied Acoustics*, vol. 44, no. 1, pp. 67 –77, 1995.

[28] J. Antoni and R. B. Randall, "The spectral kurtosis: Application to vibratory surveillance and diagnostics of rotating machines," *Mechanical Systems and Signal Processing*, 2006.

[29] D. E. Vakman, *Signals, oscillations, and waves: A modern approach*. Artech House, 1998.

[30] J. Antoni, "Fast computation of the kurtogram for the detection of transient faults," *Mechanical Systems and Signal Processing*, vol. 21, no. 1, pp. 108 –124, 2007.

[31] J. Antoni, *Fast kurtogram*, https://www.mathworks.com/matlabcentral/fileexchange/48912-fast-kurtogram, MATLAB Central File Exchange. Retrieved April 20, 2020. 2020.

[32] I. S. Bozchalooi and M. Liang, "Parameter-free bearing fault detection based on maximum likelihood estimation and differentiation," *Measurement Science and Technology*, 2009.

[33] H. Huang, N. Baddour, and M. Liang, "Bearing fault diagnosis under unknown time-varying rotational speed conditions via multiple time-frequency curve extraction," *Journal of Sound and Vibration*, 2018.

[34] ——, "Multiple time-frequency curve extraction matlab code and its application to automatic bearing fault diagnosis under time-varying speed conditions," *MethodsX*, vol. 6, pp. 1415 –1432, 2019.

[35] H. Huang and N. Baddour, "Bearing vibration data collected under time-varying rotational speed conditions," *Data in Brief*, vol. 21, pp. 1745 –1749, 2018.

[36] D. Iatsenko, P. V. E. McClintock, and A. Stefanovska, "Extraction of instantaneous frequencies from ridges in time–frequency representations of signals," *Signal Processing*, vol. 125, pp. 290 –303, 2016.

[37] M. Liebling, T.-F. Bernhard, A. H. Bachmann, L. Froehly, T. Lasser, and M. Unser, "Continuous wavelet transform ridge extraction for spectral interferometry imag-

ing," in *Coherence Domain Optical Methods and Optical Coherence Tomography in Biomedicine IX*, V. V. Tuchin, J. A. Izatt, and J. G. Fujimoto, Eds., International Society for Optics and Photonics, vol. 5690, SPIE, 2005, pp. 397 –402.

[38] NTN Global, *Ntn cylindrical roller bearings ultage series catalog*, https://www.ntnglobal.com/en/products/catalog/pdf/3037E.pdf, Access date: 2020-05-08. 2020.

[39] J. Lee, H. Qiu, G. Yu, J. Lin, and Texnord Technical Services, *Bearing data set*, IMS, University of Cincinnati. NASA Ames Prognostics Data Repository, CA. http://ti.arc.nasa.gov/project/prognostic-data-repository, 2007.

[40] C. Liu and K. Gryllias, "A semi-supervised support vector data description-based fault detection method for rolling element bearings based on cyclic spectral analysis," *Mechanical Systems and Signal Processing*, vol. 140, 2020.

[41] T. Wang, M. Liang, J. Li, W. Cheng, and C. Li, "Bearing fault diagnosis under unknown variable speed via gear noise cancellation and rotational order sideband identification," *Mechanical Systems and Signal Processing*, 2015.

[42] I. S. Bozchalooi and M. Liang, "A joint resonance frequency estimation and in-band noise reduction method for enhancing the detectability of bearing fault signals," *Mechanical Systems and Signal Processing*, vol. 22, no. 4, pp. 915 –933, 2008, Special Issue: Crack Effects in Rotordynamics.