Classification of Mechanical Fault-Excited Events based on Frequency

Arild Bergesen Husebø, Huynh Van Khang[0000−0002−0480−6859], Kjell Gunnar Robbersmyr[0000−0001−9578−7325], and Andreas Klausen[0000−0002−5411−3655]

Dept. of Engineering Sciences, University of Agder (UiA)
Jon Lilletuns vei 9, NO-4879, Grimstad, Norway
arild.bergesen.husebo@uia.no

Abstract. We propose a method for classifying periodic events generated at one or multiple frequencies on any one-dimensional space. This is very useful in problems where you need to find the type of event based on observations of location, e.g. in time. For each frequency, all events are mapped into periodic axes, which are represented independently of each other. Using an expectation-maximization algorithm, we can fit distributions to the events and classify them using maximum likelihood. The proposed method is applied to two mechanical faulty cases: a defect rolling-element bearing, and a gearbox with defect teeth. We show very good classification results in cases of multiple event types of similar frequency, multiple event types of different frequencies, and combinations of the two for artificially generated events.

Keywords: Clustering · Periodic Events · Condition Monitoring

1 Introduction

Faults and defects in mechanical components of rotating machinery are usually associated with characteristic forcing frequencies, which can be seen as components of the vibration measured from vibration sensors, e.g. accelerometers, mounted in strategic locations on the machine [2]. The characteristic frequencies can be computed analytically based on the component geometry and machine speed, e.g. the rate of which teeth on two connected gears mesh together, or the rate at which rolling elements in a bearing pass a point on the outer race. Diagnosis of the faults are usually performed by Fourier spectrum analysis of vibration signals or through more sophisticated methods like the envelope spectrum [9], by looking for peaks or changes at the characteristic frequencies. However, in cases of varying machine speed, the spectrum components might be smeared [13]. The usual solution to this problem is to perform order tracking [8, 10]. Order tracking may stretch the waveform in ways that make it less efficient for time-dependent-frequency (as opposed to speed-dependent-frequency) analysis like high-frequency resonance methods as in [14]. In such cases, the order tracking should be performed after signal filtering has been done, or in the case presented here, when the fault-excited events have been discretised as locations

in time. From a condition monitoring perspective, the biggest drawback of frequency analysis is the inability to distinguish between two or more faults that excite events at the same frequency.

Time-frequency analysis is a widely applied alternative that takes the time as well as the frequency domain into account. Its application areas contain cases, where we need to know at what time certain frequencies are most prominent, for example, if two or more faults excite events at the same rate. The most common forms of time-frequency analysis methods are through the short-time Fourier transform (STFT), the continuous wavelet transform (CWT), where the Morlet wavelet is commonly selected as the mother wavelet for mechanical condition monitoring, and less common methods like the Wigner distribution function (WDF) [5]. However, a lot of relevant time-domain information may still be lost in the process due to the transformation, and the resulting spectrograms still contain a lot of information irrelevant to the fault at hand.

By analysing the time-waveform of vibration signals, it is not only possible to see when a fault-excited event occurs, e.g. the train of impulses generated when a roller element in a bearing passes a defect on the outer race, but also gain a lot of information about how components interact with the defect area. This can in turn tell us about the shape of the defect [12, 6]. Knowing the shape, e.g. size of a defect can give us an idea about the remaining lifetime of that component. If one is able to isolate the parts of a waveform that contain fault-excited events, this information can be found based on expected patterns modeled for different defect shapes. The first step for isolating fault-excited events requires locating the event in time, being achieved through various signal processing methods, that will not be covered in this paper.

Different components degrade at different speeds, thus an intelligent system demands knowing the type of defect in question. This is rarely a problem if there is only one defect, but when there are fault-excited events from multiple sources in the same time-waveform, it becomes hard to decide which event belongs to which defect. To solve this issue, we propose an unsupervised classification method that can classify events based on their frequency and location in relation to other events. Figure 1 illustrates the task at hand. We want to decide the origin of our observed events.

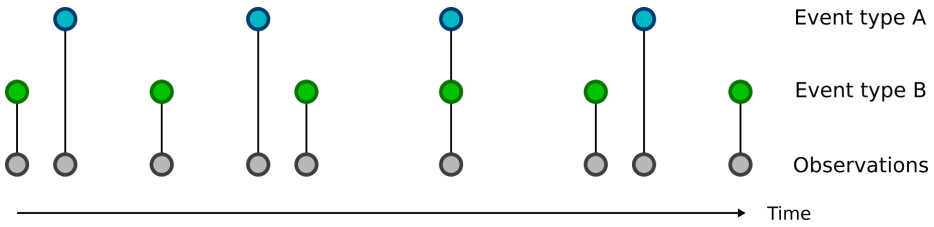

Fig. 1. Actual event types and observations

Classification is a huge topic within data science and depends on available data. Further, there are a plethora of algorithms to choose from. One of the bigger categories of classification algorithms is clustering algorithms, where the main goal is to group together data that likely belong together. Common clustering techniques include k-nearest neighbours (K-NN) [3] and k-means [7]. However, for our specific application, it is desired to learn the underlying distribution of data, thus we will apply an implementation of the expectation maximization (EM) algorithm [4]. The novelty of this paper is how the data is transformed onto periodic axes during classification such that each event is given a weight in every class and across one or more frequencies. More information about the implementation as well as its limitations are given in Section 2. Two example cases with artificial data are explained in Section 3, along with classification results and discussion.

2 Event Classification

This section uses period instead of frequency for convenience. Given a period $T = f^{-1}$, we can transform events to a periodic axis w.r.t. that period using

$$\boldsymbol{\phi}_T(x;T) = x \bmod T \tag{1}$$

Any event location falls into the interval $[0, T)$. This can be converted to radians and be represented as a point on a unit circle. For example, if $\boldsymbol{x}$ contains events generated with a period of T and a phase offset of ϕ, the transformed event locations $\boldsymbol{\phi}$ will all lie in a cluster on that circle. The location of the cluster on the perimeter would be at ϕ radians:

$$\boldsymbol{\phi}(x;T) = 2\pi\frac{x \bmod T}{T} \tag{2}$$

The importance of having correct and accurate values for T must be stressed. Even just a slight error from the true period can lead to huge difference as the total error at location x is

$$e_{\text{total}}(x;T) = e_{\text{period}} \cdot \left\lfloor \frac{x}{T} \right\rfloor \tag{3}$$

The number of classes, which generating period they belong to, must be decided. Let the following set contain all classes $\{c_1, c_2, ...c_M\}$. Each class is also associated with a period, denoted by T_c. In many practical examples, the number of classes and their associated periods are unknown. There are many ways to solve this issue, most commonly relying on some trial and error method. However, some optimizations can be made. If the number of unique generating frequencies are known, there must be at least one class to assign every frequency. If the number of event classes that exist for the same frequency is known, they should all be assigned the frequency in question. If, in addition to the previous points, the probability of a specific event occurring is known, the expected number of events that will belong to a class can be estimated based on the class frequency

and location of the last event. Multiple trial sets of classes can then be created and ordered after priority to optimize such a trial and error method. However, this is not in the scope of this paper, so we assume that the set of classes is known.

The classifier is a soft expectation-maximization algorithm that finds the components (or classes) of a mixture distribution based on data subject to a transformation dependent on the period associated with target class. Classes are represented by distributions, being part of a mixture distribution, where the probability volume must equal 1 at all times. Each class is assigned a parameter a_c that weighs the distribution according to its contribution towards the mixture, such that $\sum_{c=1}^{M} a_c = 1$.

Every class' underlying distribution is described by its respective probability density function $f_c(\phi; \boldsymbol{\theta}_c)$ and parameters $\boldsymbol{\theta}_c$. Given values for $\boldsymbol{\theta}_c$ and a_c, a membership weight for each event n in every class c, denoted as $w_{c,n}$, can be estimated:

$$w_{c,n} = \frac{a_c f_c(\phi_{c,n}|\boldsymbol{\theta}_c)}{\sum_{h=1}^{M} a_h f_c(\phi_{h,n}|\boldsymbol{\theta}_h)} \tag{4}$$

This is commonly referred to as the expectation step of an EM algorithm. As long as $\sum_{c=1}^{M} a_c = 1$, $\sum_{c=1}^{M} w_{c,n} = 1$ will hold true for any event n. An expected number of events belonging to a class can be represented using

$$N_c = \sum_{n=1}^{N} w_{c,n} \tag{5}$$

In this paper, we assume a normally distributed error on the location of every event and that each successive event location is independent of the previous one. In case of varying event period, some tracking technique must be applied. We consider the model described in equation 6 in the choice of distribution for our algorithm.

$$x(c, n) = nT_c + \phi_c + \epsilon_c \tag{6}$$

$x(c, n)$ is the location of event n of class c with $0 \leq \phi_c < T_c$ offset. T_c is the true period between events of this class and $\epsilon_c \sim \mathcal{N}(0, \sigma_c)$ is the location error with σ_c standard deviation.

The von Mises distribution is applied as it closely approximates a wrapped normal distribution. Its parameters, μ and κ, are parameters of location and dispersion, respectively. The probability density function is given by

$$f(\phi; \mu, \kappa) = \frac{\exp\{\kappa \cos(\phi - \mu)\}}{2\pi I_0(\kappa)} \tag{7}$$

where I_0 is the modified Bessel function of order zero [11].

Estimation of the parameters are done in what is commonly referred to as the maximization step. First, estimate the complex barycentre of the transformed data, weighted by the membership weight

$$r_c = \frac{1}{N_c} \sum_{n=1}^{N} \exp\{w_{c,n} \cdot j\phi_{\text{rad}}(x_{c,n}; T_c)\} \tag{8}$$

The complex argument gives the location parameter

$$\hat{\mu}_c = \mathrm{Arg}\,(r_c) \tag{9}$$

The κ cannot be directly estimated, thus the approximation presented in [1] is applied. It is described as an approximation for the parameter of the distribution on a hypersphere, i.e. von Mises-Fisher distribution, but is more accurate in lower dimensions like the one-dimensional case of this paper.

$$\hat{\kappa}_c \approx \frac{2|r_c| - |r_c|^3}{1 - |r_c|^2} \tag{10}$$

$|r_c|$ denotes the complex modulus of the barycentre. We refer to the estimation of distribution parameters as the function $\theta\,(\boldsymbol{\phi}_c, w_c)$.

Finally, new values for a_c must be estimated:

$$a_c = \frac{N_c}{N} \tag{11}$$

The mixture weights and location parameters are initialised randomly within their parameter space. The dispersion parameters are initialised to 1. Each iteration is repeated until the algorithm converges or an iteration limit is exceeded. Convergence criteria are not covered in this paper. An overview of the algorithm is given in Algorithm 1:

Algorithm 1: Classification algorithm

input : event locations $\boldsymbol{x}$
classes $\{c_1, c_2, ..., c_M\}$,
periods $T_c \forall c \in \{c_1, c_2, ..., c_M\}$
output: class contribution parameters $\alpha_c \forall c \in \{c_1, c_2, ..., c_M\}$
class distribution parameters $\boldsymbol{\theta}_c \forall c \in \{c_1, c_2, ..., c_M\}$
initialize $\alpha_c, \boldsymbol{\theta_c} \forall c \in \{c_1, c_2, ..., c_M\}$;
repeat
  foreach $c \in \{c_1, c_2, ..., c_M\}$ **do**
    for $n \leftarrow 0$ **to** N **do**
      $\boldsymbol{\phi}_c[n] \leftarrow \phi(x[n]; T_c)$;
      $\boldsymbol{w}_c[n] \leftarrow \frac{\alpha_c f_c(\phi_c[n]|\boldsymbol{\theta}_c)}{\sum_{h=1}^{M} \alpha_h f_c(\phi_c[n]|\boldsymbol{\theta}_h)}$;
    end
    $\boldsymbol{\theta}_c \leftarrow \theta\,(\boldsymbol{\phi}_c, w_c)$;
    $\alpha_c \leftarrow \frac{\sum_{n=1}^{N} \boldsymbol{w}_c[n]}{N}$;
  end
until *convergence*;

Some events of different frequencies may lie too close to each other to distinguish between solely based on this method. A solution to this problem is to

set an upper threshold on the entropy of the possible outcomes of an event, so that we can discard uncertain events. Recall that the membership weight $w_{c,n}$ assigns the probability of that event n belongings to class c. Thus, the entropy can be estimated using

$$H_n = -\sum_{c=1}^{M} w_{c,n} \cdot \log(w_{c,n}) \tag{12}$$

We use the natural logarithm to achieve the results displayed in this paper.

3 Example Cases

Two different example cases of how this method can be applied to mechanical condition monitoring are presented in this section. We will present artificial data from a defect rolling-element bearing and from gears with defect teeth. Knowing the source of events like mechanically fault-excited impulses is crucial to properly diagnosing the fault and estimating remaining lifetime. The step of detecting the events, which in reality can be performed using a variety of methods, is skipped in this paper.

3.1 Case A: Defect rolling-element bearing

A rolling element bearing has a defect in the outer race and in the inner race, both exciting impulses when rolling elements pass over. The outer race is fixed while the inner race is connected to the shaft which has a speed of 1000 RPM. Knowing the bearing geometry, dimensions and speed, the analytical frequencies of the impulses can be calculated for the considered events . The artificial data contain event locations x_n in time. They are generated according to Table 1 and based on the model given in Equation 6. Inner race events located at x_n are discarded when $\sin(2\pi\frac{1000}{60}x_n) > -0.5$ in order to simulate the fault moving out of the load zone. An additional detection rate which randomly discards $1 - P_{\text{detection}}$ of the remaining events in that class is applied. Data is collected for a 2 second window. Applying the classification algorithm to the data gives

Table 1. Rolling-element bearing defect event generation table

Defect (c)	Frequency	T_c	ϕ_c	σ_c	$P_{\text{detection}}$
Outer race	59.7586 Hz	0.01673 s	0.00527 s	0.01	90%
Inner race	90.2714 Hz	0.01108 s	0.00912 s	0.05	90%

the classifications shown in Figure 2. Note that the shown classifications are simply based on which component having an event with the highest probability.

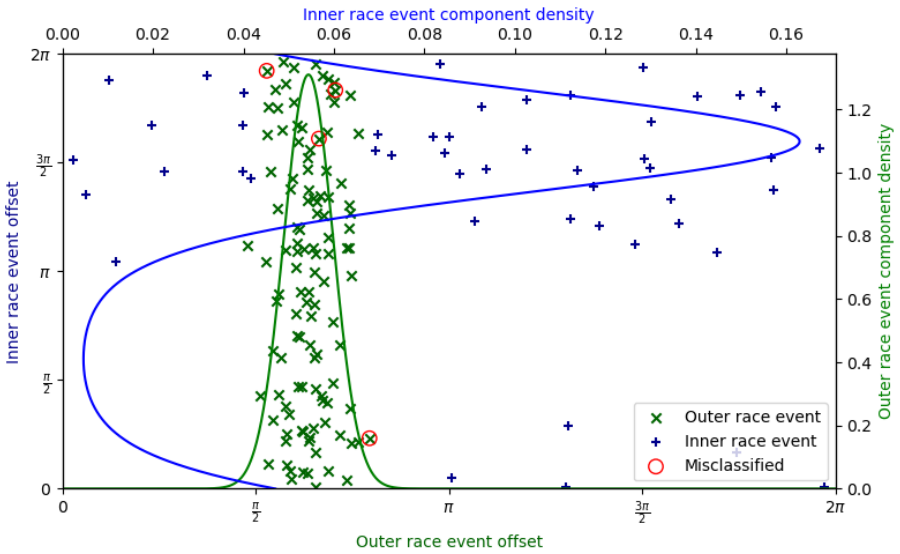

Fig. 2. Case A - Classifications and distributions

The transformed (Equation 2) time axes are plotted against each other with respect to their own period. The alternative axis shows the density of the mixture components resulting from the classification. The error from mean offset is much higher for the inner race event, which corresponds well with the table. Notice how events from the inner race "wraps around" the y-axis. This reflects the importance of using the von Mises distribution when working with high-variance error in event locations. There are a few misclassifications (marked by a red circle). They are located in the area where the events from the two classes intersect. These are locations where, in the time domain, events from different classes lie close to each other.

For our two-class case, the solution to equation $\alpha_1 f_1(\phi(x[n]; T_1)|\boldsymbol{\theta}_1) = \alpha_2 f_2(\phi(x[n]; T_2)|\boldsymbol{\theta}_2)$ is plotted to show the bounds of each class. This can be seen as different colored areas in Figure 3. The red area marks an upper entropy threshold of 0.3. Note that the events are marked based on their actual, true class.

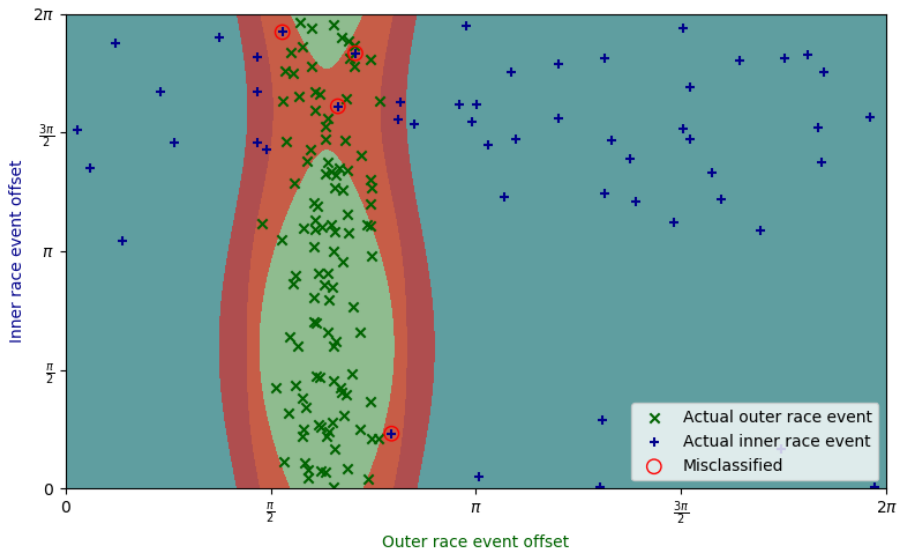

Fig. 3. Case A - Class bounds and entropy threshold

Due to the amount of events belonging to the outer race class compared to the inner race and the low dispersion of events within this class, the outer race class gets assigned most of the events that fall above the entropy threshold.

3.2 Case B: Defect gear teeth

An input shaft and an output shaft are connected by two gears. The input gear has 64 teeth while the output gear has 48 teeth. Two teeth on the input gear are worn enough to cause large amounts of backlash. One tooth on the output gear is missing. The rotational frequency of the input shaft is 7 Hz, thus the output shaft rate is $\frac{64}{48} \cdot 7\text{Hz} \approx 9.333\text{Hz}$. The event locations x are generated according to Table 2. Data is generated for a 5 second window.

Table 2. Gear tooth defect event generation table

Defect (c)	Frequency	T_c	ϕ_c	σ_c	$P_{\text{detection}}$
Input gear tooth 1	7.000 Hz	0.14286 s	0.01786 s (Tooth 8)	0.002	20%
Input gear tooth 2	7.000 Hz	0.14286 s	0.06697 s (Tooth 30)	0.002	20%
Output gear tooth	9.333 Hz	0.10714 s	0.08928 s (Tooth 40)	0.003	90%

Figure 4 shows the classifications and their component density distributions for the second case. Notice the cluster patterns forming along both axes. This is due to the synchronicity of the involved frequencies, i.e. 4:3 gear ratio. The true

input gear tooth components are similar, although the resulting dispersion parameter is higher for the first input gear tooth component. It is noted that some of the events belonging to this class are mistaken for output gear tooth events (red circles), affecting the parameter. However, increasing the data generation window reduces the error, making this issue seemingly more of a population size problem.

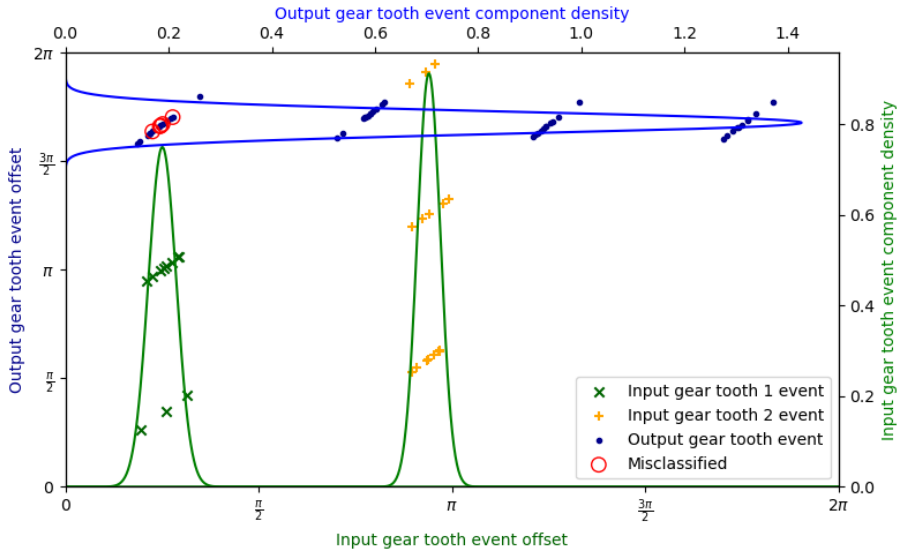

Fig. 4. Case B - Classifications and distributions

An interesting effect of this approach is that if the locations or just the phases of the shafts are known, the tooth of which a fault is located can be found.

Similar to case A, the solution to $\alpha_c f_c(\phi(x[n];T_c)|\boldsymbol{\theta}_c) = \max\left(\alpha_{c'} f_{c'}(\phi(x[n];T_{c'})|\boldsymbol{\theta}_{c'})\right), c' \in \{c_1, c_2, ..., c_M\}\backslash\{c_c\}$ can be plotted for every class to get the classification bounds as shown in Figure 5. The same upper entropy threshold of 0.3 is used.

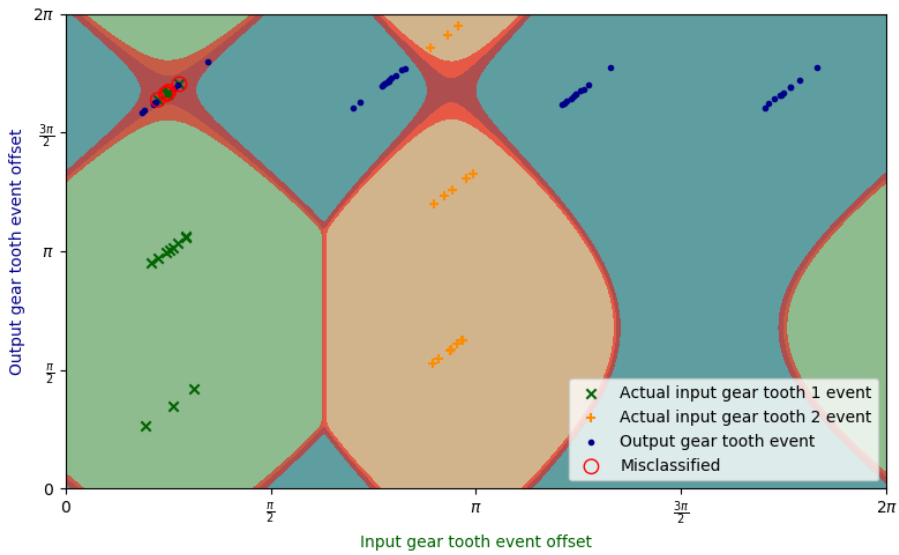

Fig. 5. Case B - Class bounds and entropy threshold

The figure shows that most classifications are unproblematic, except the cluster to the top left. This cluster lies close to the bounds between input gear tooth 1 class and the output gear tooth class. As previously mentioned, these events are impossible to distinguish based solely on their location with respect to each other.

It is worth noting that the algorithm does not guarantee convergence to a global maximum. The authors have noted cases, where two clusters that are located close to each other. might merge into one component, resulting in a very low weight for the second component. This in turn may lead to computational errors. In some cases, it might be required to perform multiple executions of the algorithm with different initial parameters. Other such cases may need more or new data.

4 Conclusion

A novel method to classify events based on their location and frequency of occurrence is presented in this study. The technique can be used for other applications that seek to achieve event classifications, but this paper applies it to mechanical fault-excited events from a bearing and a gearbox, showing very good results for artificial data. The clustering method uses an EM algorithm that first transforms data with respect to the frequency assigned a class. The learned mixture model consists of von Mises distributed components to account for the periodic offsets of events. The algorithm is not guaranteed to find the global maximum,

and therefore problems might occur depending on the data and/or initial conditions. Another drawback is the need to specify number of classes and their associated frequencies beforehand, which is common in many classification algorithms. Finally, it is essential with correct and accurate frequencies for the method to work.

References

1. Banerjee, A., Dhillon, I., Ghosh, J., Sra, S.: Clustering on the unit hypersphere using von mises-fisher distributions. Journal of Machine Learning Research **6**, 1345–1382 (09 2005)
2. Choudhary, A., Goyal, D., Shimi, S.L., Akula, A.: Condition monitoring and fault diagnosis of induction motors: A review. Archives of Computational Methods in Engineering **26**(4), 1221–1238 (Sep 2019). https://doi.org/10.1007/s11831-018-9286-z, https://doi.org/10.1007/s11831-018-9286-z
3. Cunningham, P., Delany, S.J.: k-nearest neighbour classifiers: 2nd edition (with python examples). CoRR **abs/2004.04523** (2020), https://arxiv.org/abs/2004.04523
4. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society. Series B (Methodological) **39**(1), 1–38 (1977), http://www.jstor.org/stable/2984875
5. Feng, Z., Liang, M., Chu, F.: Recent advances in time–frequency analysis methods for machinery fault diagnosis: A review with application examples. Mechanical Systems and Signal Processing **38**(1), 165–205 (2013). https://doi.org/https://doi.org/10.1016/j.ymssp.2013.01.017, https://www.sciencedirect.com/science/article/pii/S088832701300071X, condition monitoring of machines in non-stationary operations.
6. Ismail, M., Klausen, A.: Multiple defect size estimation of rolling bearings using autonomous diagnosis and vibrational jerk. In: 7th World Conference on Structural Control and Monitoring (7WCSCM). China (07 2018)
7. Jin, X., Han, J.: K-Means Clustering, pp. 563–564. Springer US, Boston, MA (2010). https://doi.org/10.1007/978-0-387-30164-8_425, https://doi.org/10.1007/978-0-387-30164-8_425
8. Klausen, A., Khang, H.V., Robbersmyr, K.G.: Multi-band identification for enhancing bearing fault detection in variable speed conditions. Mechanical Systems and Signal Processing **139**, 106422 (2020). https://doi.org/https://doi.org/10.1016/j.ymssp.2019.106422, https://www.sciencedirect.com/science/article/pii/S0888327019306430
9. Klausen, A., Robbersmyr, K.G., Karimi, H.R.: Autonomous bearing fault diagnosis method based on envelope spectrum. IFAC-PapersOnLine **50**(1), 13378–13383 (2017). https://doi.org/https://doi.org/10.1016/j.ifacol.2017.08.2262, https://www.sciencedirect.com/science/article/pii/S2405896317330550, 20th IFAC World Congress
10. Li, Y., Ding, K., He, G., Jiao, X.: Non-stationary vibration feature extraction method based on sparse decomposition and order tracking for gearbox fault diagnosis. Measurement **124**, 453–469 (2018). https://doi.org/https://doi.org/10.1016/j.measurement.2018.04.063, https://www.sciencedirect.com/science/article/pii/S0263224118303440
11. Mardia, K.: Directional statistics (2000)

12. Sawalhi, N., Randall, R.: Vibration response of spalled rolling element bearings: Observations, simulations and signal processing techniques to track the spall size. Mechanical Systems and Signal Processing **25**(3), 846–870 (2011)
13. Villa, L.F., Reñones, A., Perán, J.R., de Miguel, L.J.: Angular resampling for vibration analysis in wind turbines under non-linear speed fluctuation. Mechanical Systems and Signal Processing **25**(6), 2157–2168 (2011). https://doi.org/https://doi.org/10.1016/j.ymssp.2011.01.022, https://www.sciencedirect.com/science/article/pii/S0888327011000677, interdisciplinary Aspects of Vehicle Dynamics
14. WANG, W.: Early detection of gear tooth cracking using the resonance demodulation technique. Mechanical Systems and Signal Processing **15**(5), 887–903 (2001). https://doi.org/https://doi.org/10.1006/mssp.2001.1416, https://www.sciencedirect.com/science/article/pii/S0888327001914165