

Politecnico di Torino

Master's degree in Mechanical Engineering

Frequency Domain Order Tracking and Synchronous Average for Rotating Machinery Diagnostics



**Politecnico
di Torino**

Supervisors:

Prof. Alessandro Fasana

Daga Alessandro Paolo

Candidate:

Bekhzod Abdullaev

June 2022

Abstract

When we deal with rotating machinery, in order to monitor the characteristic and quality of the rotating parts, common method is to sample the accelerometer signals using discrete time axis with constant time interval. However, in most of the cases, rotational speed of the machine is nonstationary. Therefore, the characteristic frequencies of interest (speed of the shafts, gears and bearings) are not constant. So, signals sampled evenly in time cannot be used directly for condition monitoring.

In order to deal with these problems we need algorithms that are robust to changes in rotational speed. In this regard, Order Tracking and Synchronous Average are well-established algorithms.

Main idea of the Order Tracking is to find the specific sampling coordinates that result in signals sampled at constant angular increments, no more at constant time increments. OT also allows further averaging (Synchronous Average) to easily separate deterministic, periodic component and non-deterministic one. This separation for example allows for bearing characteristic signals to be distinguished from the gear contribution, fostering diagnostics.

In this paper, a Novel Frequency Domain algorithm is proposed and its performance is compared with Benchmark algorithms as well.

Contents

1.	Introduction.....	4
1.1	What is Rotating Machinery diagnostics.....	4
1.2	Spectrum and how to use it for diagnostics.....	4
1.3	Problems when speed is not constant.....	5
2.	Frequency Domain.....	7
2.1	How to compute Fourier Transform.....	7
2.2	FT for discrete signals.....	8
2.3	Zero padding.....	10
3.	Order Tracking.....	12
3.1	Conventional Order Tracking.....	12
3.2	Computed Order Tracking.....	13
4.	Fourier Interpolation.....	19
4.1	Implementation of Fourier Interpolation.....	19
4.2	Problem of Fourier Interpolation.....	21
5.	Time Synchronous Average.....	22
5.1	Definition and implementation of TSA.....	22
5.2	Comb filter effect of the TSA.....	26
5.3	Effect of the number of periods used to synchronise.....	27
6.	Synchronous Average in Frequency Domain.....	29
6.1	Introduction of DFT.....	29
6.2	Root Mean Square Error.....	31
7.	Discrete Fourier Transform.....	36
7.1	Implementation on real data.....	36
7.1.1	PHM.....	36
7.1.2	Safran Contest.....	43
8.	Conclusion.....	47
	Reference.....	48

1. Introduction

1.1 What is Rotating Machinery diagnostics

Being the crucial part of the Rotating Machinery, failure of the bearings is one of the main reason of the machine downtime and even breakdown. Therefore, early prediction of these failures can prevent from major damages to rotating machinery and also reduce the price of repair. The idea is to average the periods to separate the periodic part of the signal from non-periodic one (noise). To monitor the condition of the Rotating Machinery, various techniques are proposed nowadays and Order Tracking and Synchronous Average are well-established ones among them.

1.2 Spectrum and how to use it for diagnostics

On the other hand, signals (periodic and also aperiodic) can be represented in the frequency domain and it represents the how their power or energy is spread to different frequencies. This allocation of power to frequency axis is called the *spectrum of the signal*. Spectral representation of the signal can be discrete or continuous depending on if the signal is periodic or aperiodic. Because, power of the periodic signal is concentrated at frequencies multiples of a so-called fundamental frequency, directly related to the period of the signal. However, aperiodic signal represents the spectrum that is continuous function of frequency. As vibration signals consist of deterministic part that comes from gears and non-deterministic part that may come from bearing damages (or from random noise) in the Rotating Machinery, their spectrum can illustrate the multiples of fundamental frequency along with other valuable information about possible bearing damages and anomalies in the system. Therefore, Spectrum representation in the frequency domain can give more information that is interesting for Condition Monitoring than the time-based representation.

1.3 Problems when speed is not constant

When the working conditions are not stationary and the rotational speed varies in time, the period of each rotation differs. It means, as long as the sampling frequency is constant, number of samples for periods are not constant. Therefore, phase of the sampled points does not match with the phase of points sampled in other cycles.

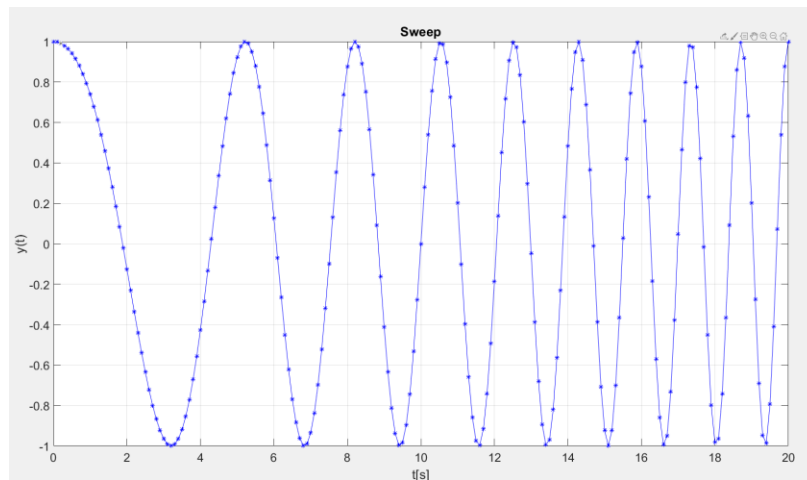


Figure 1. Sweep signal

In figure 1, Sweep signal is tested with sampling frequency $f_s=10\text{Hz}$ for 20s. Frequency content of the signal varies from 0.1 Hz to 0.8 Hz in this time.

```
fs=10; %Hz or sps
T=20; %s
tt=[0:1/fs:T];
nn=1:length(tt);
yy = chirp(tt,0.1,T,0.8);
plot(tt,yy,'b+-'),xlabel('t[s]'),ylabel('y(t)'), title('Sweep')
```

From the figure 1, it is visible that number of samples for each period is different and also they are situated in different phase angles.

This leads to couple of problems.

First, performing the Spectrogram analysis on the overall length of the signal gives the useless spectra (figure 2).

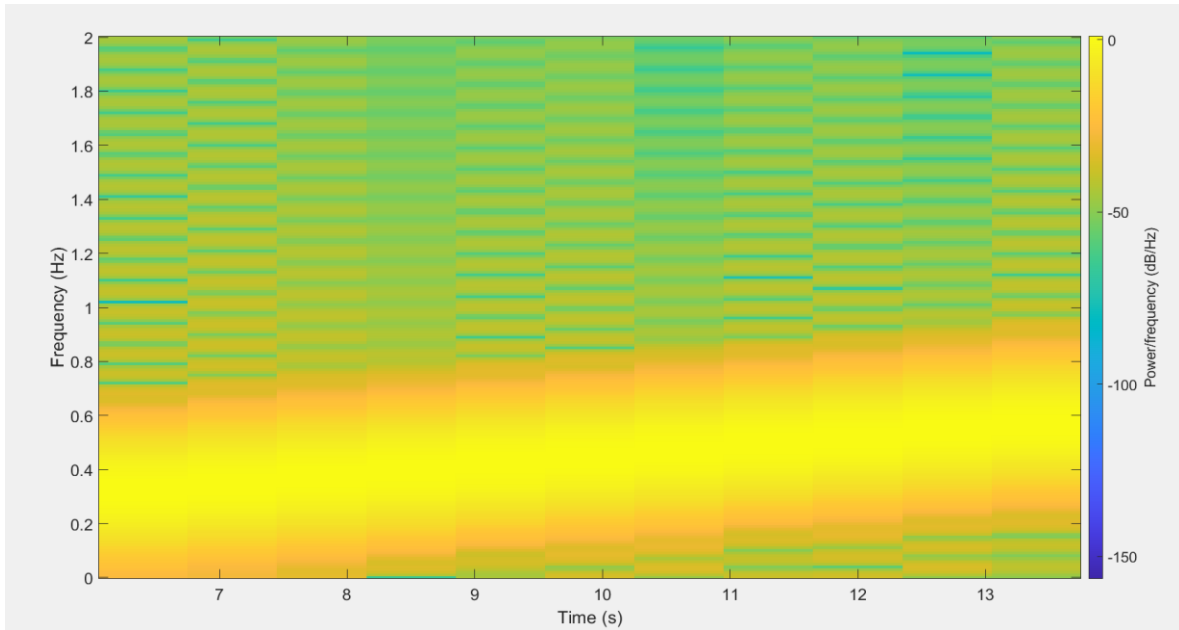


Figure 2. Spectrogram of sweep signal

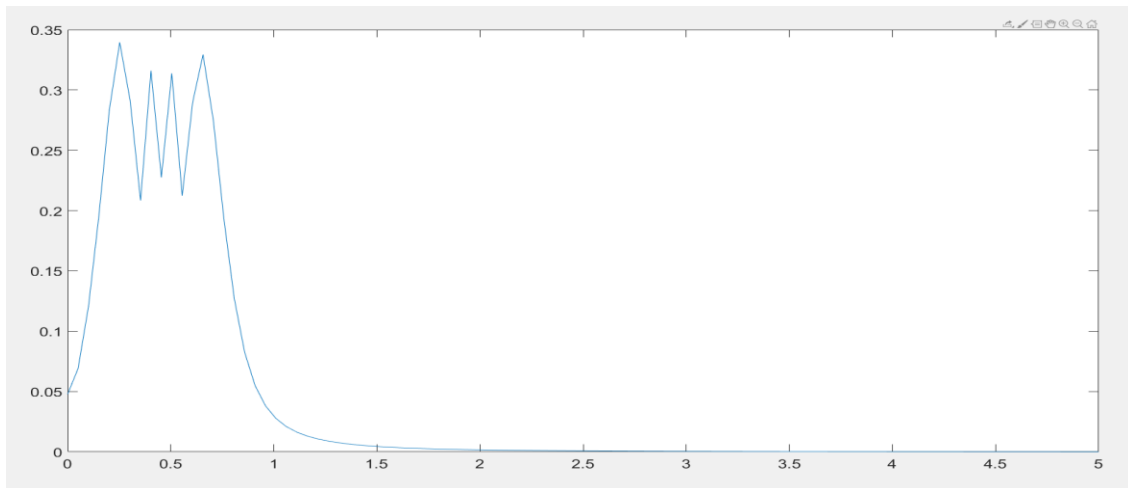


Figure 3. FFT of the Sweep signal

Second, it is not possible to get one average signal from the cycles that contain different number of samples and at various phase positions.

2. Frequency Domain

2.1 How to compute Fourier Transform

To draw frequency domain representation of the signal, knowledge about Fourier series and Fourier Transform is needed. Fourier says that "Every signal can be represented as a sum of harmonic signals".

$$x(t) = a_0 + \sum_{k=1}^{\infty} (a_k \cos(k\Omega_0 t) + b_k \sin(k\Omega_0 t)) \quad (1)$$

Where Ω_0 is the fundamental frequency and $k\Omega_0 = \Omega_k$ k^{th} harmonic

$$a_0 = \text{mean value} = \frac{1}{T} \int_T^{t+T} x(t) dt$$

$$a_k = \frac{2}{T} \int_0^T x(t) * \cos(k\Omega_0 t) dt$$

$$b_k = \frac{2}{T} \int_0^T x(t) * \sin(k\Omega_0 t) dt$$

$$x(t) = a_0 + \sum_{k=1}^{\infty} (a_k \cos(k\Omega_0 t) + b_k \sin(k\Omega_0 t)) = a_0 + \sum_{k=1}^{\infty} c_k (\sin(k\Omega_0 t + \varphi)) \quad (2)$$

Where c_k is the Amplitude of the k^{th} harmonic and φ_k is the phase

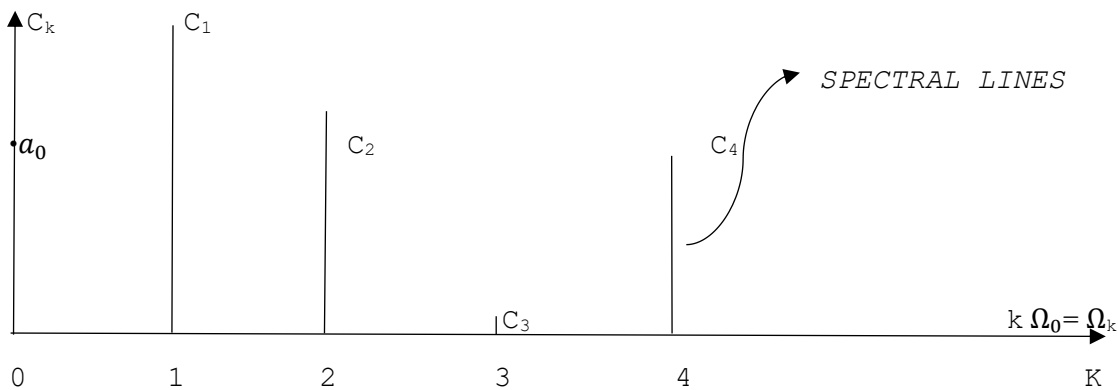


Figure 4. Spectral lines.

The ensemble of the spectral lines for all the frequencies forms the *SPECTRUM*. It can be easily proved that the spectral lines for opposite sign of k are conjugate of each other that means their module is equal and so they are symmetric with respect to origin. Therefore, negative values of k are not usually plotted in the spectrum representation to show meaningful frequency axis.

What if a signal is aperiodic? In this case, signal is assumed to be periodic with period $T \rightarrow \infty$.

$$X(\Omega) = \int_{-\infty}^{\infty} x(t)e^{-i\Omega t} dt = \text{Fourier Transform of } x(t) = \text{Spectrum of } x(t) \quad (3)$$

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\Omega)e^{i\Omega t} d\Omega = \text{Inverse Fourier Transform} \quad (4)$$

2.2 FT for discrete signals

When digital acquisition is performed, acquired signal is a discrete signal instead of continuous signal. In this case:

$$X(k) = \frac{1}{2N} \sum_{n=1}^{N-1} x_n * e^{-ik2\pi n/N} \quad (5)$$

It is called Discrete Fourier Transform (DFT)

How does acquisition parameters effect the Fourier Transform?

Parameters of data acquisition are Δt and fs . k in the formula is frequency index that means $\Omega_k = \Omega_0 k$.

Frequency resolution: $\Delta\Omega = \Omega_{k+1} - \Omega_k = \Omega_0$

$$\Delta\Omega = \frac{2\pi}{T} = 2\pi\Delta f \quad (6)$$

$$\Delta f = \frac{1}{T} = \frac{1}{N\Delta t} = \frac{fs}{N} \quad (7)$$

Here it is visible that when Fourier Transform of discrete signal is calculated, samples are assumed to be acquired at N constant angle increments of the revolution and frequency content of the transform ranges from $-fs/2$ to $fs/2$.

In matlab Discrete Fourier Transform is performed by *fft* function, which stands for Fast Fourier Transform.

$$X = \text{fft}(x); \quad (8)$$

Dividing frequency axis into evenly spaced N (number of samples in acquisition) intervals sometimes may cause some problems. For instance, simple sine wave with 0.47 Hz frequency is tested with $fs=10\text{Hz}$ sampling rate.

```
fs = 10; % sampling freq [Hz]
t = 0:1/fs:5; % time interval
y = sin(2*pi*0.47*t); % test signal
plot(t,y,'.-');
Y = fft(y);
f123=linspace(-fs/2,fs/2,length(Y));
figure; plot(f123,fftshift(abs(Y)),'b.-');
```

The result is shown in figure 5. It is visible that pick in the frequency axis is not exactly at 0.47 Hz. It is because of the spacing of the frequency axis by *fft*. New approach that is motivated by this problem is presented later in this paper.

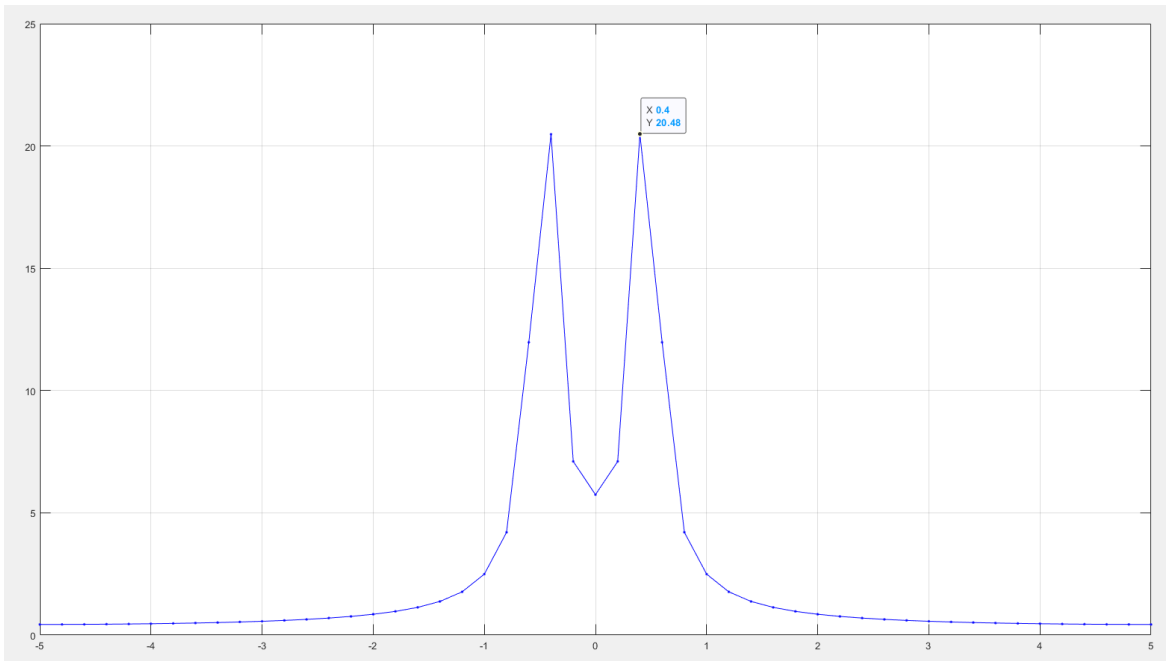


Figure 5. FFT of the test signal

2.3 Zero padding

Problem mentioned in chapter 2.2 can be solved by using `fft(y,N)` function of the Matlab which is used in order to increase the number of samples to take a Fourier Transform so as to reach a better frequency axis resolution. As indicated in eq.7 as much the number of samples used in Fourier Transform means as smaller the frequency steps are. In this case, if higher N than the actual length of the signal is used, matlab takes the signal samples and adds zeros at the end until N number is reached. That is way this technique is called as *ZERO PADDING*. Here it is shown that by using N 5 times the length of the signal resolution of the frequency axis refined a lot so the new peak appears at $f=0.47$ Hz (fig 6).

Attention should be paid to correctly represent the frequency axis and to the amplitude of the spectrum.

First, when matlab calculates the FFT of the signal it uses length of the signal to divide the frequency axis in this way:

$$[0:fs/N:fs-fs/N] \quad (9)$$

Not from $-fs/2$ to $fs/2$ as wrongly calculated above.

Second, after taking FFT, always need to be divided to the length of the signal before zero padding in order to get correct amplitude of the spectrum equal to the amplitude of the raw signal, not to the complete length of the signal with zeros.

```
fs = 10; % sampling freq [Hz]
t = 0:1/fs:5; % time interval
y = sin(2*pi*0.47*t); % test signal
plot(t,y,'r.-');
N=5*length(y);
Y = fft(y,N);
% f123=linspace(-fs/(2),fs/(2),length(Y));
f123=[0:N-1]*fs/N;
figure; plot(f123(1:N/2),2*abs(Y(1:N/2))/length(y),'b.-');
grid on, hold on
```

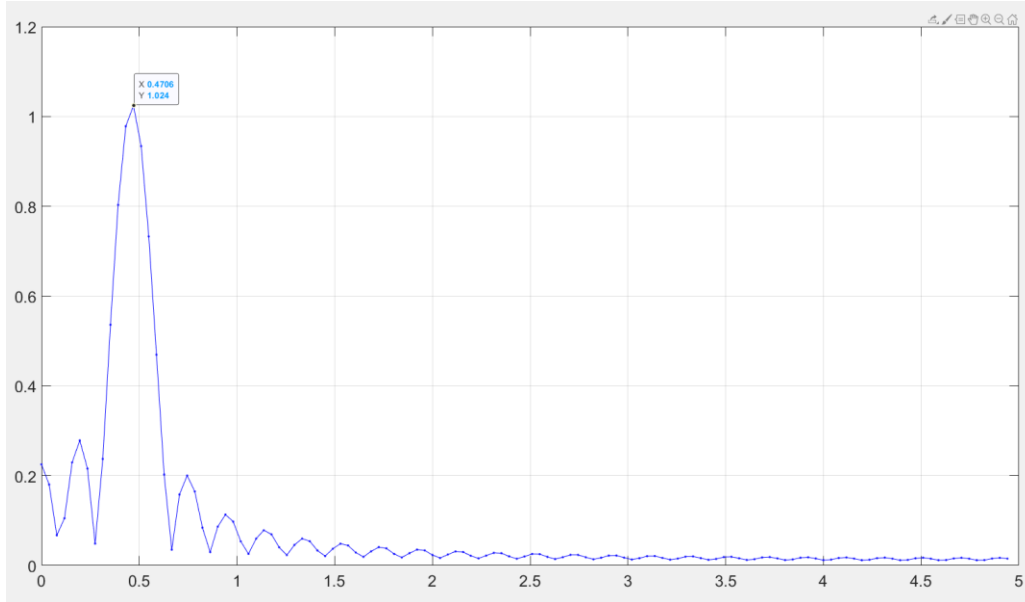


Figure 6. FFT with zero padding

It is also visible that there are some sidelobes in the spectrum. It is because of added zeros to the signal which are not really there.

As added values are all zeros, they do not give extra information about the spectrum, but only uses more point to better interpolate the signal before (figure 7). It is clearly visible in the figure where two representations are combined. This affect will be explained later in the chapter 5.2.

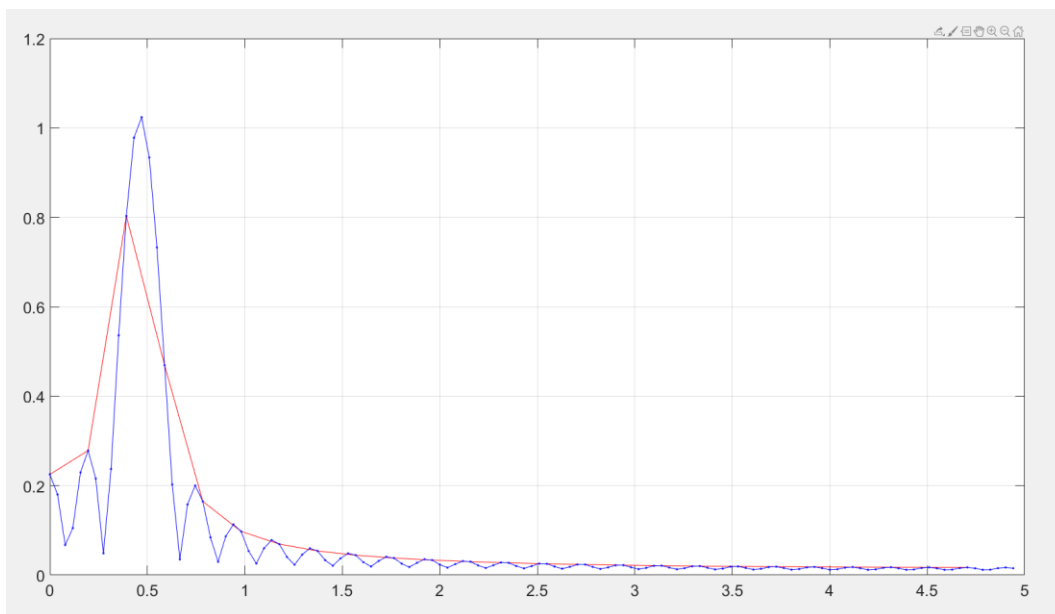


Figure 7. Comparison between with and without zero padding

3. Order Tracking

In vibration analysis of rotating machinery, frequency axis that represent the orders of the running speed rather than the absolute frequencies is often preferred which is called Order Tracking (OT) method. OT is very powerful and most commonly used method for condition monitoring of rotating machines, because it can easily identify vibrations related to rotational speed such as shaft defects and bearing wear. In order to use this method, the signal must be acquired at constant increments of shaft rotation. (Fyfe, K. R., & Munck, E. (1996). *Analysis of computed order tracking*)

3.1 Conventional Order Tracking

Conventional OT uses special analogue instrumentation to directly acquire a data at proportional rate to shaft frequency. Usually it consist of a ratio synthesiser and an anti-aliasing tracking filter. It may also include a frequency counter to monitor the shaft speed.

The ratio synthesiser generates a signal proportional to the shaft speed of the machine. This output is used to control the sampling rate and the cut-off frequency of the analogue tracking filter, a low-pass filter with an adjustable cut-off frequency. Once data acquisition process performed at constant $\Delta\theta$ is finished, a Fast Fourier Transform (FFT) is calculated resulting in an order spectrum.

Its usage is reduced because of the associated cost and complexity of the components. Conventional Order Tracking is also prone to errors when working condition is nonstationary.

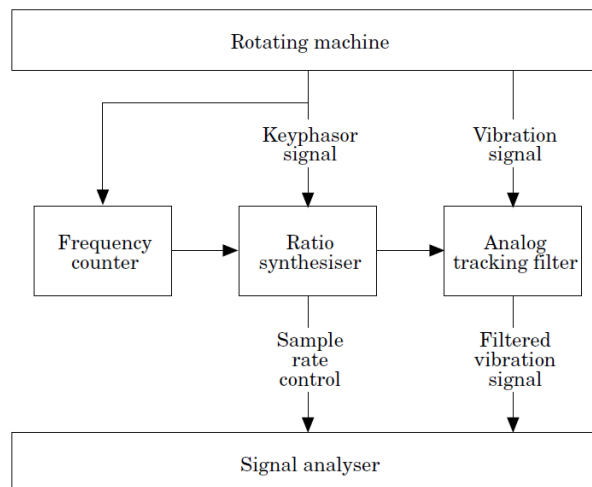


Figure 8. Equipment used for Conventional Order Tracking

3.2 Computed Order Tracking

Computed Order Tracking (COT) method that is digital alternative of the conventional one is introduced because of its some advantages over the former one. Main advantage of the Computed Order Tracking is that does not require the samples to be sampled at constant increments of the shaft rotation angle, but it uses samples taken at constant Δt time and keyphasor pulses to define desired sampling points at constant $\Delta\theta$.

That is reason why it is called Computed Order Tracking (COT) because it uses numerical calculation to extract speed-normalised data from the vibration signal that has been obtained through traditional data acquisition.

Keyphasor signal works like trigger and gives desired number pulses per revolution. Once the conventional data acquisition is done, it is resampled by software that uses tacho (keyphasor) signal to generate vibration amplitudes at constant $\Delta\theta$ angle increments.

Let's take as an example the sweep cosine signal represented in figure 1. It has been sampled at constant time increments and by the software it should be resampled at constant increments of shaft angle. To this regard, we have to find new time vector at which resampled data should be acquired depending on the keyphasor pulses. Two estimations are made during resampling procedure. The first one is while the calculation of sample time and putting it correctly on the time (independent) axis. It should be very precise, because it defines the quality of the interpolation, without precise calculation of resampling points, even with good interpolation technique, accuracy will not get better. The second one is involved in the resamples on the amplitude (dependent) axis which is again interpolation process.

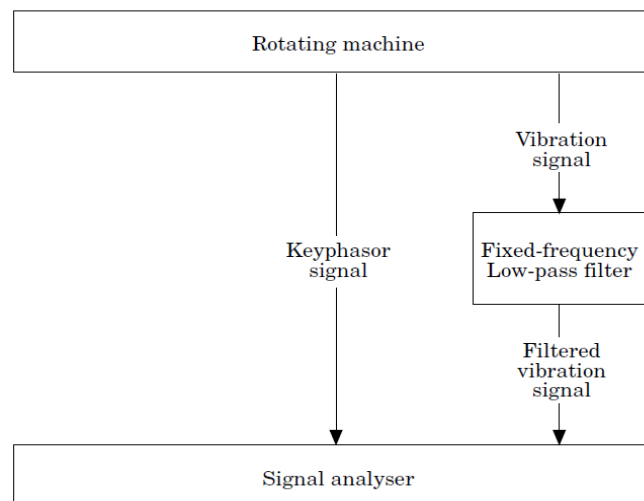


Figure 9. Equipment used for Computed Order Tracking

In order to calculate the resampling times, shaft speed is assumed to change at a constant rate. Therefore, rotational angle θ can be defined as equation time.

$$\theta(t) = b_0 + b_1 t + b_2 t^2 \quad (10)$$

The unknown coefficients b_0 , b_1 , and b_2 are defined by fitting three consecutive keyphasor pulse times (t_1 , t_2 and t_3), which occur at known shaft angles because the number of triggers per revolution is known. For example, if we have keyphasor pulse once per 2π angle (one trigger on the shaft) it means these times (t_1 , t_2 and t_3) belongs to:

$$\theta(t_1) = 0$$

$$\theta(t_2) = 2\pi$$

$$\theta(t_3) = 4\pi \quad (11)$$

Times t_1 , t_2 , and t_3 are known from the keyphasor pulse times. If these times are not known, for instance if there are multiple triggers per revolution and exact zero crossing times are not available, it is possible to identify these zero crossing times by interpolating two sample points that are one before crossing the time axis and the other one is after. If the sampling frequency is high enough, which means there enough samples per revolution, linear interpolation method may be enough. Linear interpolation is proven to work well with at least eight samples per cycle. With lower values problems during interpolation may arise and a different fitting may be needed. Note that relative times are preferred to reduce the error during the calculation.

```

function [kptempi]= keyphasorDEF(y,t,ind)

nn=2; % one zero every 2

s=sign(y);
tt=filter([1 1],1,s); % kind of square wave with a point in 0
pos=find(tt==0);

punti=pos(1:nn:end);

kptempi=t(punti); %preallocation

if ind==0
% Refinement: considering the computed zeros as a starting point we are
% taking one previous and following points, interpolating them with a
% linear function and finding out the roots of that function.
for k=1:length(punti)
    ttt=t(punti(k)-1:punti(k)+1);
    yyy=y(punti(k)-1:punti(k)+1);

    % 1: linear interpolation should be sufficient
    radice=roots(polyfit(ttt,yyy,1)); % degree 1: linear
    % rad=find(radice > ttt(1) & radice < ttt(3)); % in case degree>1;
    % kptempi(k)=radice(rad); %it takes the exact root, even if
    % samples will be out of initial time axis
    kptempi(k)=radice;
end
end

```

Once we have zero crossing times, by substituting them into equation (1)

$$\begin{pmatrix} 0 \\ \Delta\Phi \\ 2\Delta\Phi \end{pmatrix} = \begin{bmatrix} 1 & t_1 & t_1^2 \\ 1 & t_2 & t_2^2 \\ 1 & t_3 & t_3^2 \end{bmatrix} \begin{Bmatrix} b_0 \\ b_1 \\ b_2 \end{Bmatrix} \quad (12)$$

These equations are solved for the unknown coefficients and then with these known coefficients equation (10) can be solved for generic time that belongs to constant $\Delta\theta$ angle increments of interest. It results:

$$t = \frac{1}{2b_2} [\sqrt{4b_2(\theta - b_0) + b_1^2} - b_1] \quad (13)$$

By this solution time related to any phase angle can be found.

```
function [t_new]= resampleDEF(tempi,phi,K)
t_new=[];
for m=1:length(tempi)-2

    clear k tmp
    if m==1
        k=(0:3*K/2)/K; %initial
    elseif m==length(tempi)-2
        k=(K/2+1:2*K)/K; %final
    else
        k=(K/2+1:3*K/2)/K;
    end

    % (b) coefficients computation

    tt=tempi(m:m+2)-tempi(m); % relative times
    A=[1 1 1; tt; tt.^2]';
    noto=[0 phi 2*phi]';
    b=A\noto; % vector of 3 coefficients
           % (we approximate theta(t) with a second order polynomial)
    clear tt;

    % new local time axis

    if abs(b(3))<1e-5
        tmp=tempi(m)+(k.*phi-b(1))/b(2);
    else
        tmp=tempi(m)+1/(2*b(3))*(sqrt(4*b(3)*(k.*phi-b(1))+b(2)^2)-b(2));
    end
    t_new=[t_new tmp];
end
```

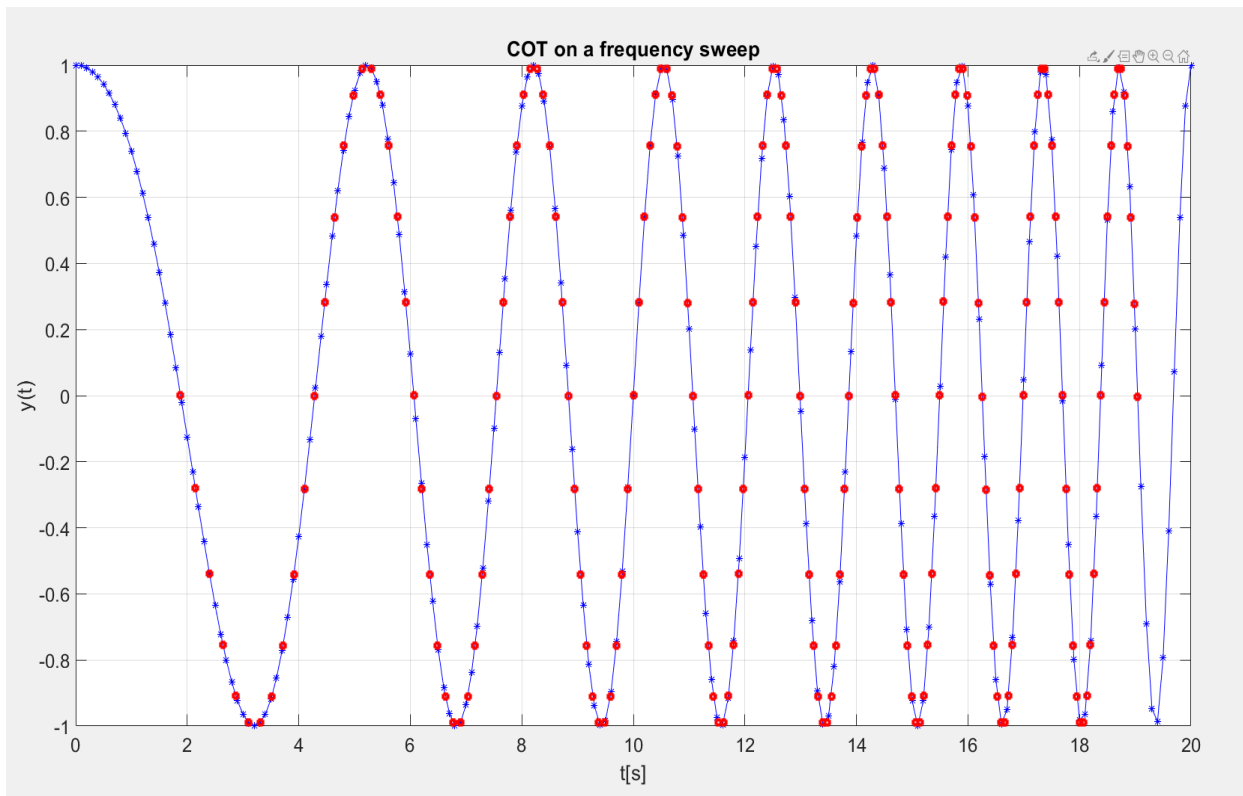


Figure 10. COT resampling

In figure 10, the signal is represented after resampling with $K=22$ data per revolution. It should be noted that the resampled data (red points) placed at the same increments of each wave regardless of where the traditional time-based samples were acquired. In order to find the amplitude values of resampling times raw signal is used to interpolate at intermediate points by *interp1* function in matlab.

```
tneww= resampleDEF(kptempiw2,phi,K);  
sgnlw = interp1(tt,yy,tneww2,'spline');
```

Once the resampling process is done, resampled data then Fourier Transformed by FFT to obtain order domain representation of the signal (figure 11).

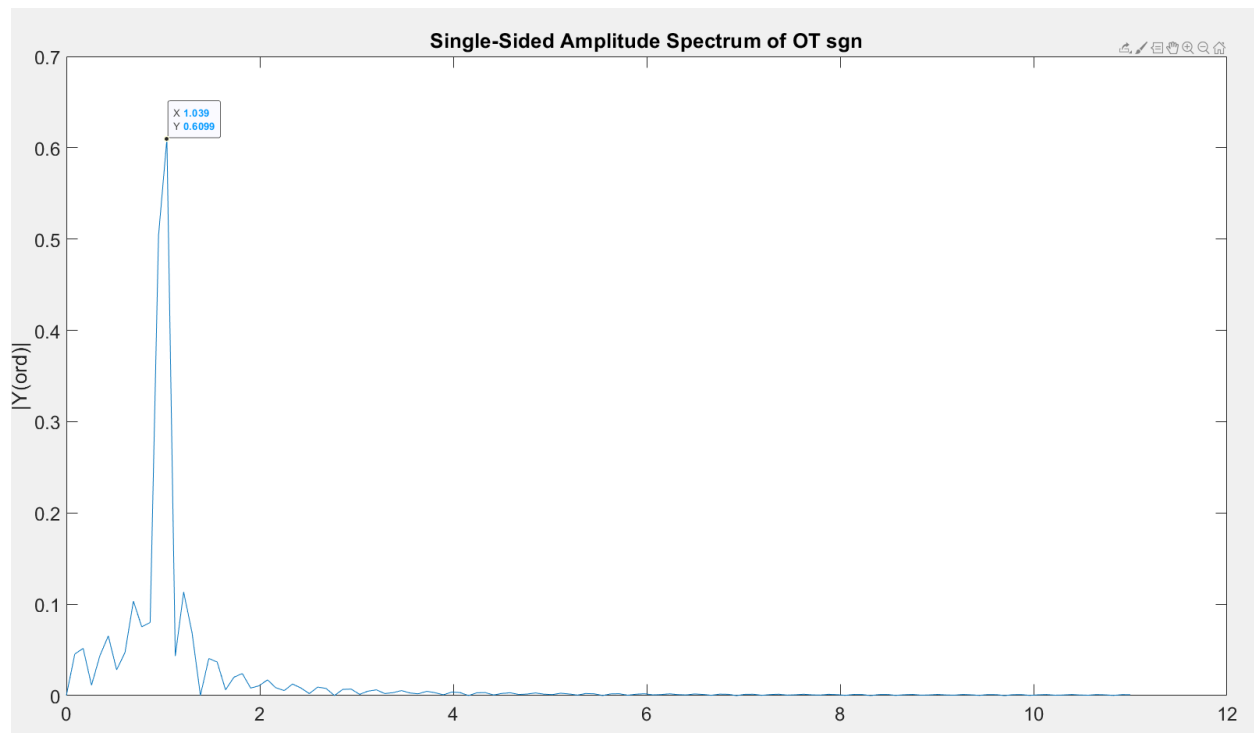


Figure 11. Single-sided spectrum of OT signal

Now the spectrogram of the signal is centred at around one (figure 12), meaning that regardless of what speed change, the peak is at order 1 which is equal to shaft speed.

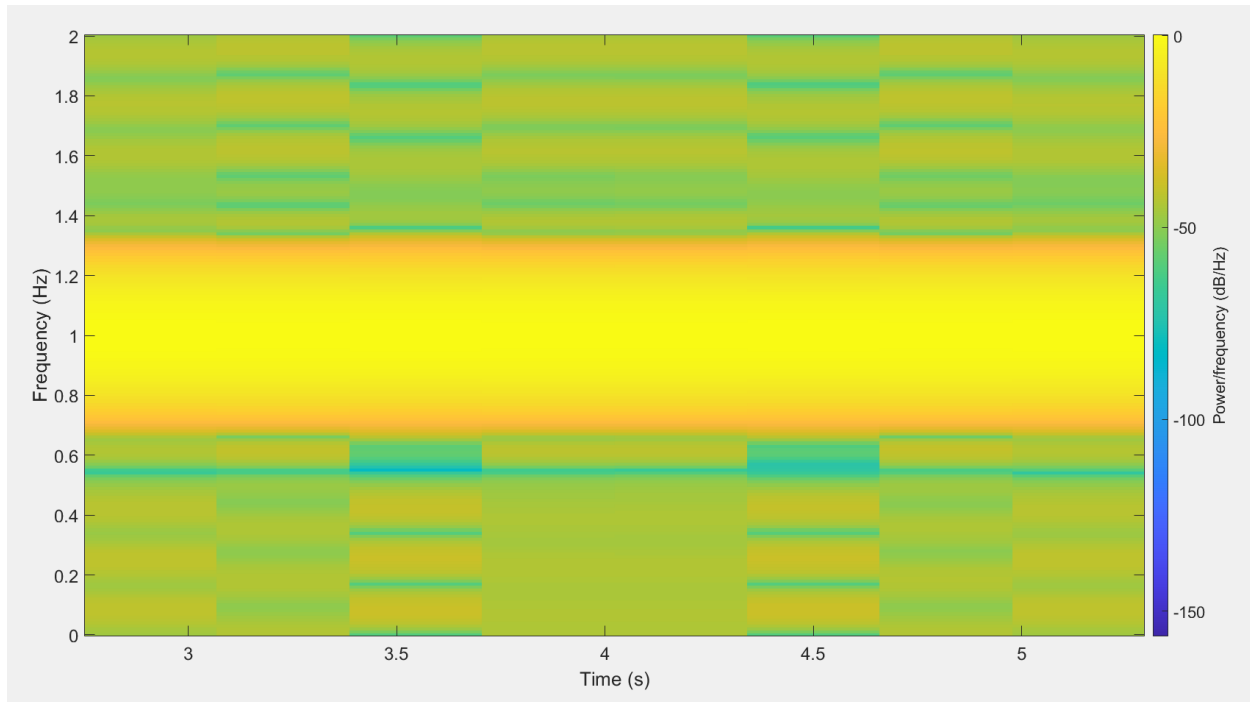


Figure 12. Spectrogram of signal after COT

It should be noted that accuracy of the interpolation method is as important as the accuracy of resampling times. Because interpolation method directly effects the amplitude of the resampled data, so the accuracy of the representation.

4. Fourier Interpolation

4.1 Implementation of Fourier Interpolation

As already mentioned in chapter 2, there are two estimations in COT process which are firstly to find exact positions of the resampling and the secondly to estimate the correct value corresponding to those positions. In addition, the zero crossing times are found by interpolating the tacho signal at certain areas. In order to check if it is possible reduce these estimations and get more precise evaluation of values that belongs to constant shaft angle increments, new approach of resampling is also tested which is called Fourier Interpolation.

To use this method, again, signal must be cut into revolutions. Therefore, tacho signal is still needed and keyphasor timing is still there. After finding the zero crossing times and signal is cut into cycles, every cycle then re-interpolated using the raw signal and interpft function of the matlab.

As an example, the sweep signal from chapter 1.3 is tasted with N=35 sampling points.

```
%% Fourier interpolation
figure;
N = 35;
for i=1:length(kptempiw2)-1
    t1=tt(tt>=kptempiw2(i) & tt<kptempiw2(i+1));
    l=length(t1);
    y1=yy(tt>=kptempiw2(i) & tt<kptempiw2(i+1));
    plot(t1,y1,'b.-')
    hold on

    y = interpft(y1,N);
    dy=length(t1)/fs/N;
    x2 =t1(1):dy:t1(end);
    m(i)=length(x2);
    if(i>1)
        if(m(i)<m(i-1))
            N=N+1;
            y = interpft(y1,N);
            dy=length(t1)/fs/N;
            x2 =t1(1):dy:t1(end);
            m(i)=length(x2);
        end
        if(m(i)>m(i-1))
            N=N-1;
            y = interpft(y1,N);
            dy=length(t1)/fs/N;
            x2 =t1(1):dy:t1(end);
            m(i)=length(x2);
        end
    end
    y = y(1:length(x2));
    plot(x2,y,'r.-')
    grid on,hold on
end
```

In figure 13 result of the Fourier interpolation is shown with red colour. It is noticeable that result is very smooth and interpolation much better compared to raw signal, which is linearly interpolated and represented with blue colour.

In the early stage, superiority of the Fourier interpolation is not much visible. It is because at first angular speed of the signal was not high enough to see the problems regarding to interpolation. But later on, as speed get high enough, some errors are visible when there are not enough sampling points per revolution and advantage of Fourier interpolation is clearly visible.

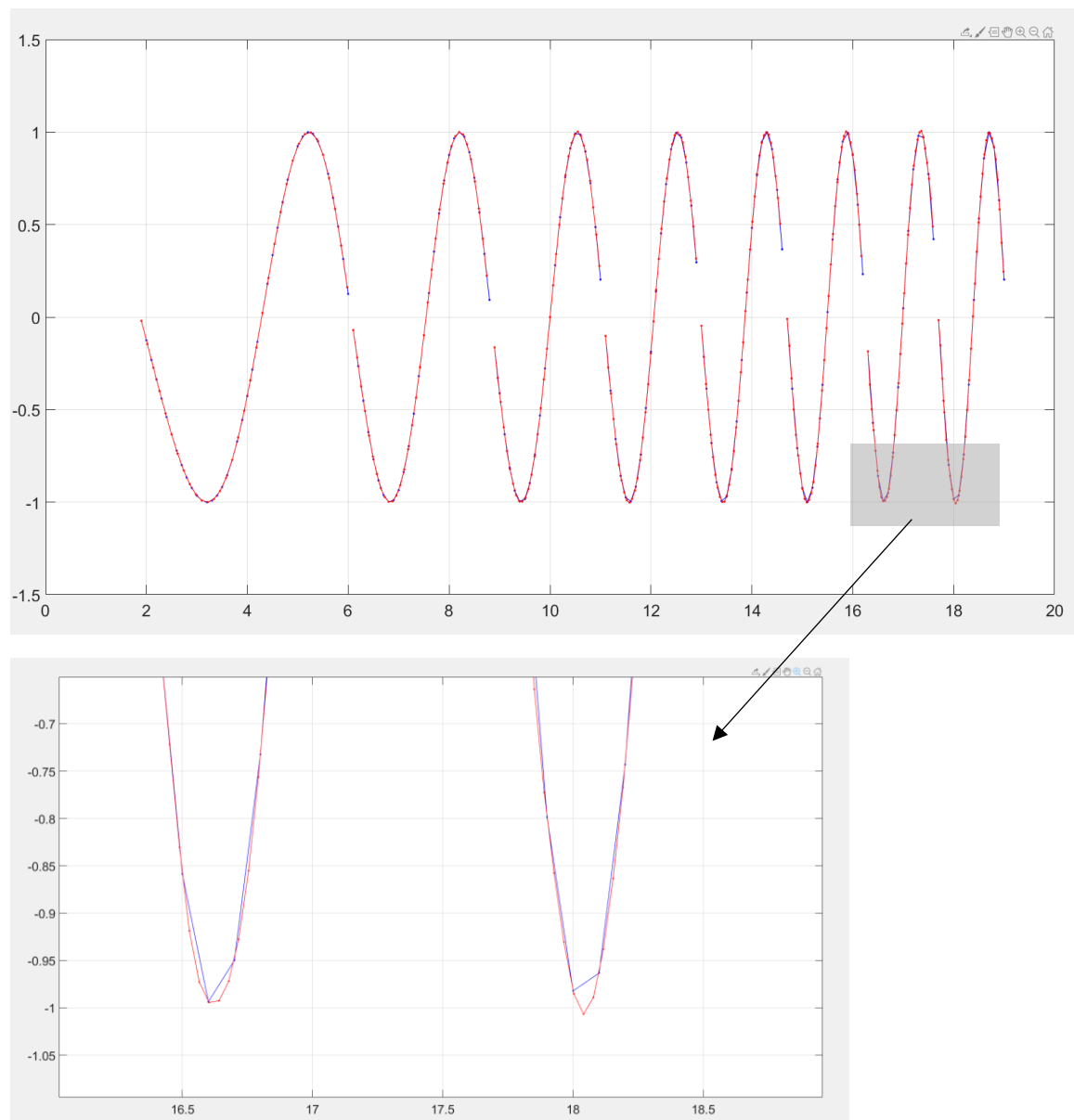


Figure 13. Interpolation comparison

4.2 Problem of Fourier Interpolation

`y = interpft(X,N)` interpolates the Fourier transform of the function values in `X` to produce `N` equally spaced points. So, we can sample each revolution with the same number of samples. As long as we have exactly the same number of samples per cycle, it can be continued with TSA to get better estimation of the true values of corresponding shaft increments. However, from the figure, it is possible to see there are still some problems that should be solved first. From the figure, one can tell that for each cycle, starting point is the first raw sample of the signal. When we cut raw data into revolutions, we do not usually have first samples at the same shaft angle. Therefore, it is not possible to average the periods to obtain better result, even if we have exactly the same number of samples per cycle. To solve the problem further improvement of the approach is test by moving the phase of the resampled points. We tried to move the phase of every first sample of the cycles to zero crossing point.

To this regard, once we cut the raw data into cycles, for each cycle post processing has been done to change the phase of the samples included in the cycle. In this process samples are first Fourier transformed

$$Y = \text{fft}(y); \quad (14)$$

$$f = [0:\text{length}(y)-1] * fs / \text{length}(y); \quad (15)$$

Then every point of the Fourier transform is phase shifted to the value that is equal to the difference between the angles at keyphasor zero crossing time and at the first sampling time of the cycle.

$$Y' = Y .* \exp(-(-1)^{0.5} * dt .* (f * 2 * \pi)); \quad (16)$$

$$dt = t(1) - kptempiw2(i); \quad (17)$$

Where `i` is the index of the cycle. Pay attention not to denote complex number by `i`, system confuses it with cycle index and big problems in the result arise.

Then, by inverse transform it is possible to take phase shifted values.

$$y' = \text{ifft}(Y'); \quad (18)$$

Finally, obtained y' signal is used to implement the Fourier Interpolation instead of raw signal.

The result is shown in figure 14. It is visible that samples in each cycle is moved back towards zero point. However, it is also clearly visible that they are not perfectly started from keyphasor zero crossing time.

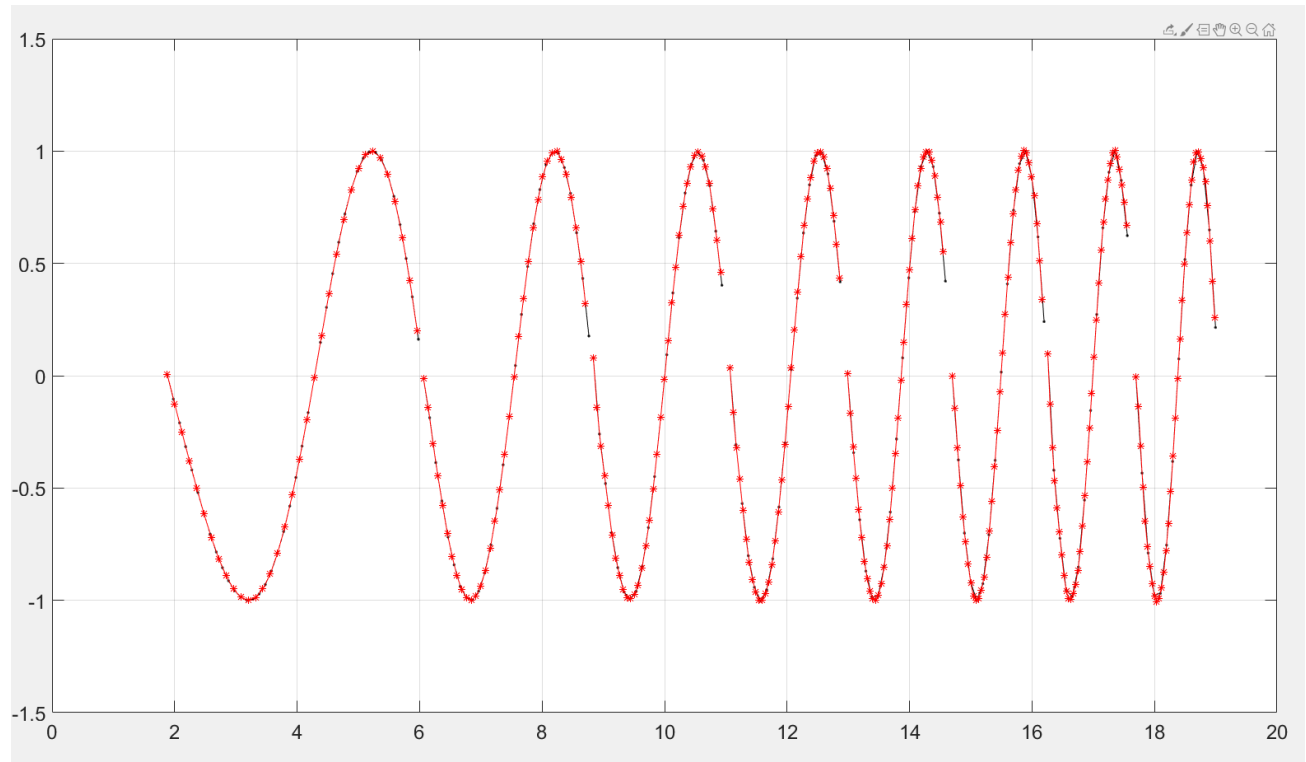


Figure 14. Fourier interpolation result after phase shift

5. Time Synchronous Average

5.1 Definition and implementation of TSA

Once the COT process is finished, as we have the resampled signal at constant increments of the shaft angle, it is easy to move on further improving the resampled data by dividing it into cycles and then averaging them resampled increment of shaft angle. It is used to distinguish the deterministic, periodic part of the signal from a non-deterministic noise. Because signals are considered to be the sum of periodic component and non-periodic noises. However, non-periodic part of the signal has a mean value, which is equal to zero. Therefore, by

averaging the values of the signal at the same angular increments leads to remove all non-deterministic part of the signal.

$$y(t) = \frac{1}{N} \sum_{n=0}^{N-1} y(t + nT) \quad (19)$$

This formula illustrates the main idea of the Time Synchronous Averaging (TSA). Where T is the main shaft period, N is the number of periods in the whole signal that shows the number of revolutions during the acquisition time.

Once we obtain the periodic part of the signal by TSA, by subtracting the deterministic part from the raw signal, we can easily determine the non-periodic (residual) part of the signal.

```
%function [x_per,x_ran]= TSA(y,K)
%
% Time Synchronous Averaging TSA
% given the signal vector y,
% given the number of samples per cycle K
%
% x_per will be the periodic part of the signal
% x_ran will be the random part
%
% Do note, it holds only for Order Tracked signals
% (sampling at constant angular increments)

function [x_per,x_ran]= TSA(y,K)

x_tsa=zeros(1,K);
KK=floor(length(y)/K);
for k=0:KK-1
    x_tsa=x_tsa+y(1+k*K:(k+1)*K);
end
x_tsa=x_tsa/KK; %average at every angle increment (phi/K)
x_per=x_tsa; % Periodic part
for k=1:KK-1
    x_per=[x_per x_tsa];
end
x_ran=y(1:KK*K)-x_per; % Random part
```

This matlab function is built to calculate the Time Synchronous Average of the given y signal which is resampled at K number of points per revolution. Those K resampling points should be at the same shaft angle increments to be able to average them so they should belong to Order Tracking process. Here is a simple test is shown to illustrate the working principle and quality of the TSA.

In order to do this sine signal with small noise is generated and sampled at constant time interval. Because of the speed of the signal

is constant, phase angles of the samples corresponding to each period is the same and TSA is applicable.

```
% test of TSA
clear all
close all

set(0,'DefaultAxesFontSize',18);

fs=40; %Hz or sps
T=5; %s
t=[0:1/fs:T-1/fs];
y=sin(2*pi*t)+0.03*randn(size(t));
plot(t,y,'*-');
hold on
[Sh_per,Sh_ran]= TSA(y,fs);
plot(t,Sh_per,'.-')
```

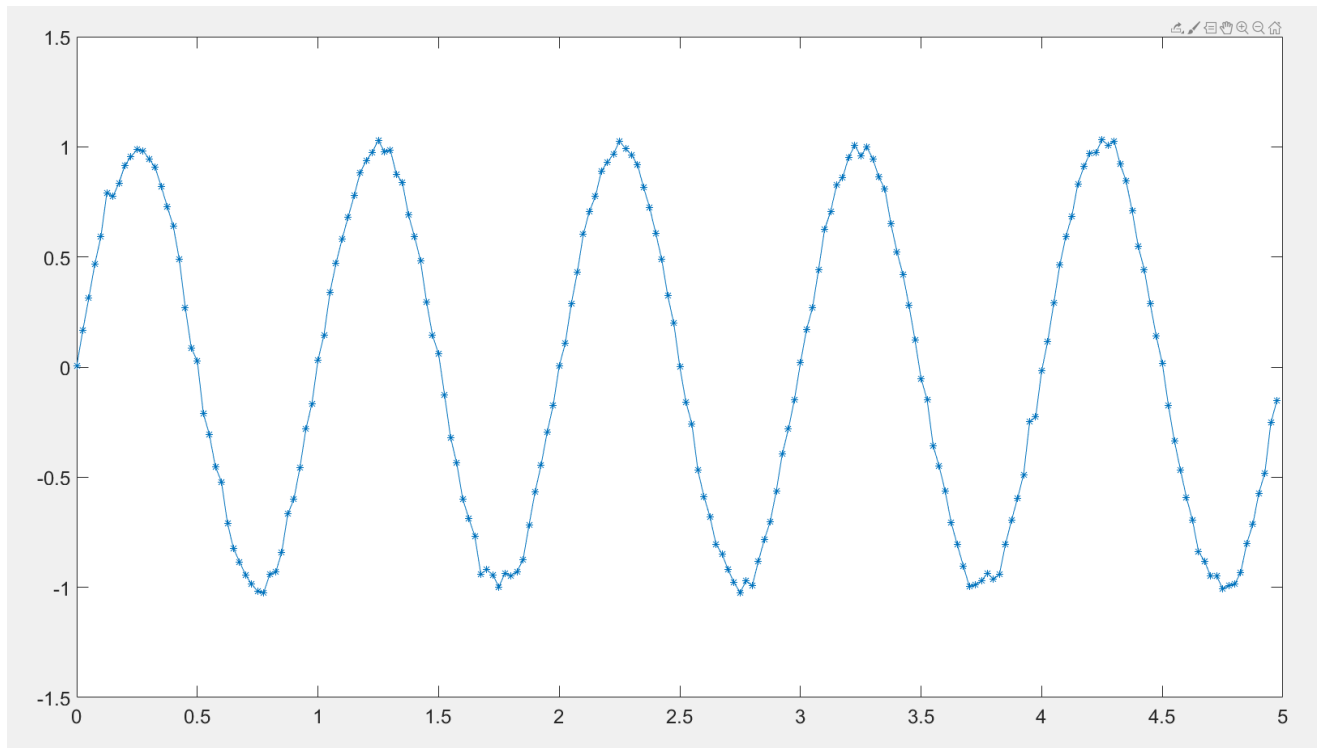


Figure 15. Before TSA

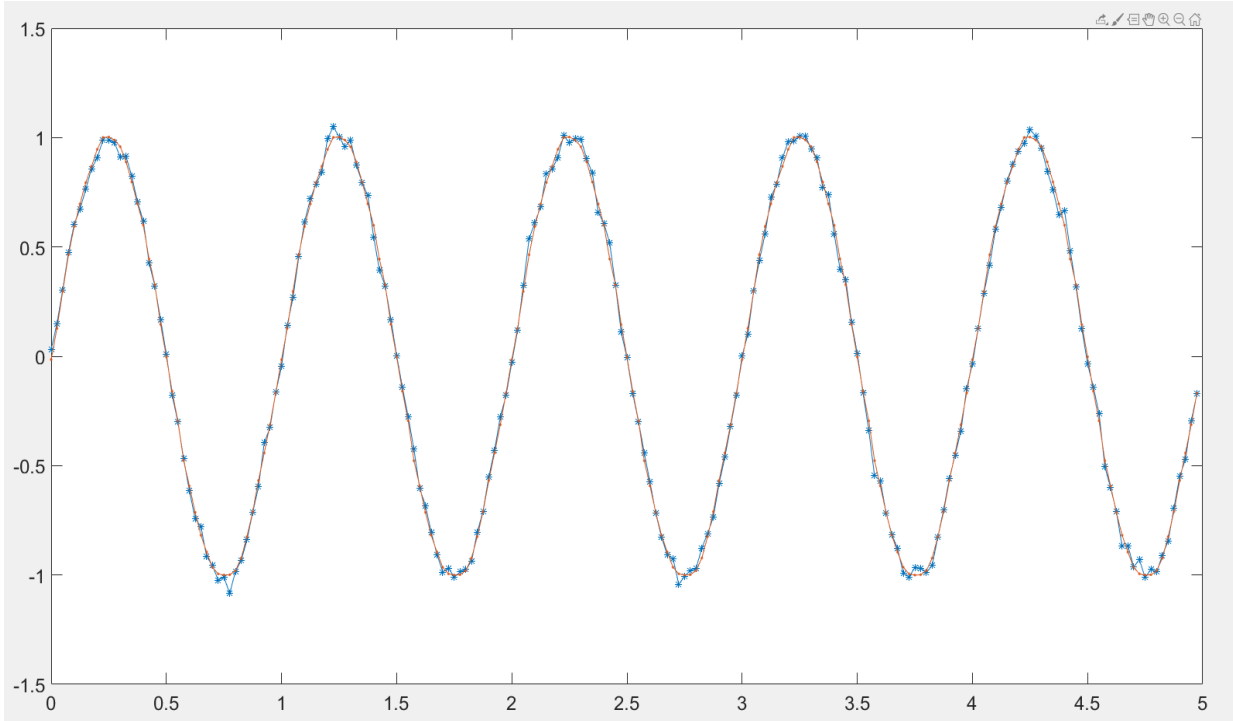


Figure 16. After TSA.

Here it is visible the effect of the TSA. Red line belongs to signal after the TSA, which is periodic part of the signal.

5.2 Comb filter effect of the TSA

As the samples of the raw signal separated by the exact period are averaged, any periodic component synchronous with the shaft speed, remains unchanged and any other becomes attenuated and converge asymptotically towards zero.

The frequency response of equation 19 can be computed via the z -transform as

$$Y(z) = \frac{1}{N} \sum_{r=1}^{N-1} (1 + z^{-rM})X(z)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{N} \frac{1 - z^{-MN}}{1 - z^{-M}} \quad (20)$$

Where M is the number of samples per period and N the number of sections averaged.

The frequency response then being

$$\left| H\left(\frac{f}{f_p}\right) \right| = H(z) \Big|_{z=\exp(-j\omega\Delta t)} = \frac{1}{N} \frac{\sin(\pi N f / f_p)}{\sin(\pi f / f_p)} \quad (21)$$

With $f_p=1/(M\Delta t)$ the frequency of the extracted periodic component.

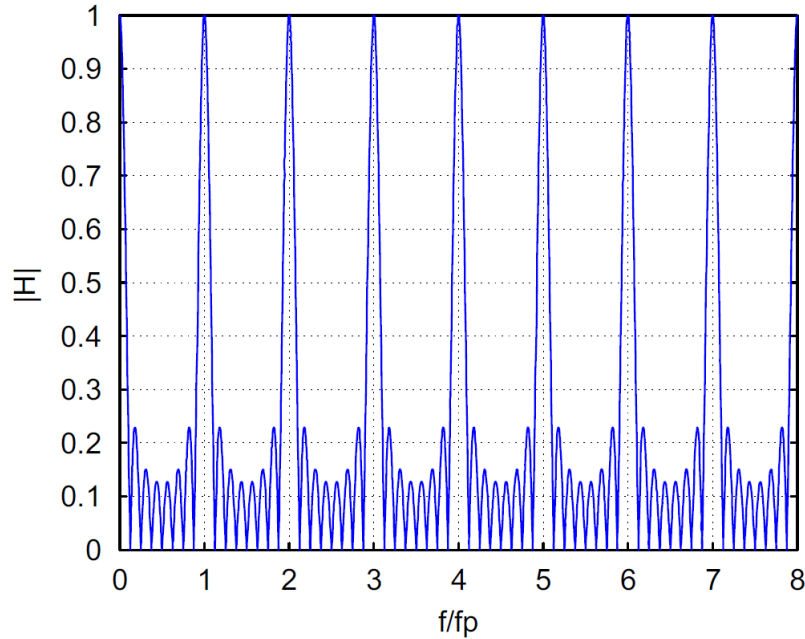


Figure 17. FRF of the SA

It is visible in the figure 17 that SA has “comb” like effect in the frequency domain. Main lobes are centered at the integer multiples of the shaft speed f_p . Therefore, this method is ideal for extracting the fundamental and as well as all harmonics of the signal while reducing the non-synchronous frequency components. Increasing N , the number of sections averaged, results in narrower main lobes (improved filtering of the periodic data) and increasing the secondary lobes. (S. Braun. *The synchronous (time domain) average revisited*)

5.3 Effect of the number of periods used to synchronise

During the TSA procedure, influence of the speed variation is lost as number of samples per cycle in the resampled signal is fixed. That is why, in the frequency domain, Fourier Transform of the TSA represents the order spectrum of the given signal. Zero padding as mentioned in chapter 2, can sharpen the spectrum by increasing the number of points to interpolate in the frequency axis. However, it may not always result to converge to true values of the orders.

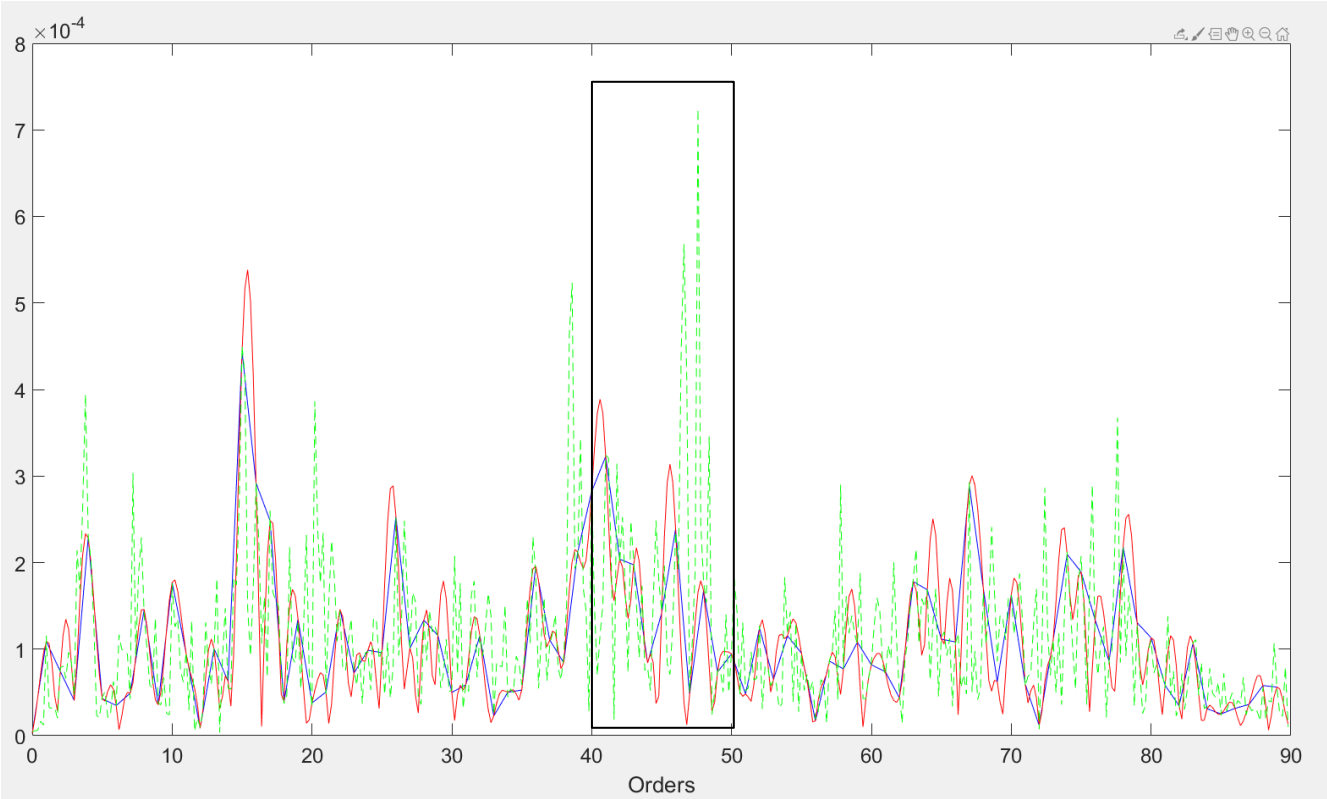


Figure 18.

Figure 18 represents the order spectrum of the signal acquired from a civil aircraft that will be briefly described in section 7.2.

Blue line belongs to spectrum of the signal after TSA without zero padding meaning that the length of frequency axis is equal to the length of the signal. Red one is with zero padding, increasing the length in frequency domain 5 times of the original signal and green dashed line shows the spectrum with the same length as red one but instead of increasing it by zero padding, 5 cycles are used to TSA.

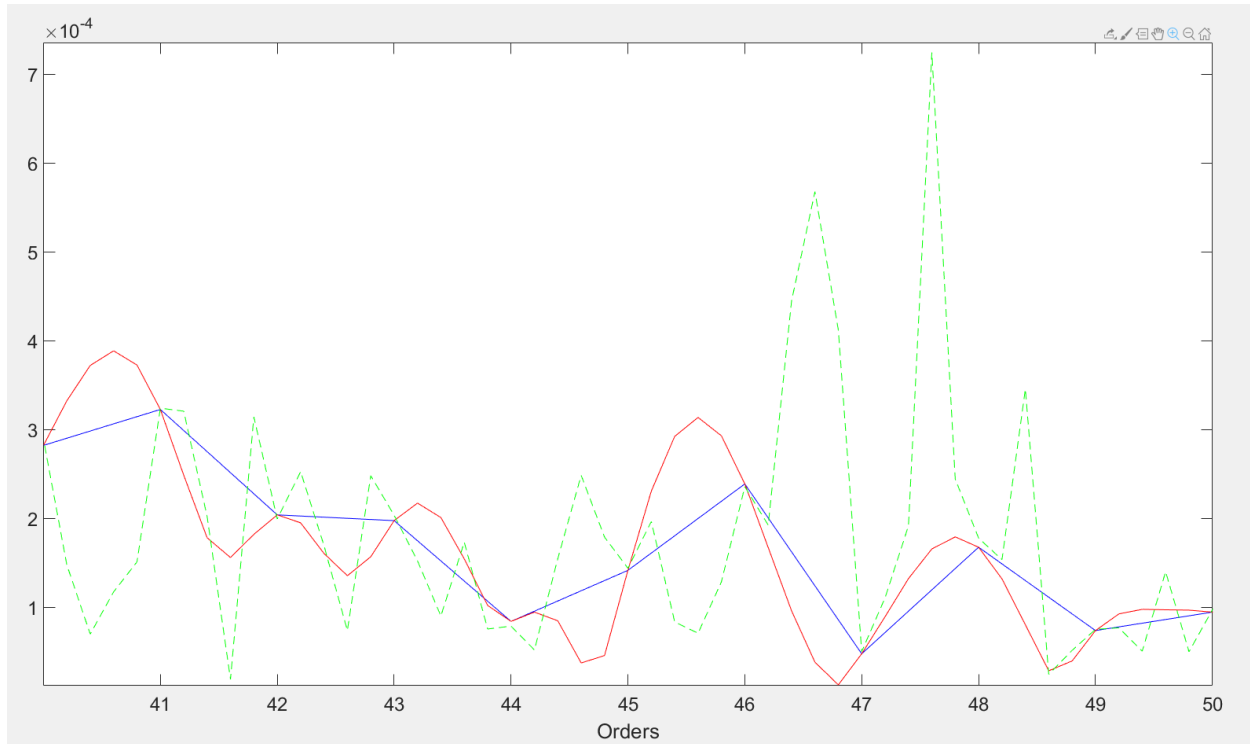


Figure 19.

In figure 19, it is possible to see that they are all equal where the blue one is calculated. Once the length is elongated by zero padding, it only uses those internal points to better interpolate the values of orders. On the other hand, increasing the number of cycles to synchronous averaging considers real data to calculate the spectrum at those internal points. In this way, if needed, desired frequency resolution can be achieved with meaningful data.

However, in the case of fast speed variations, problem may arise if too many cycles are chosen for TSA. This compromise between number of cycles and error will be discussed in chapter 6.2.

6. Synchronous Average in Frequency Domain

Synchronous Averaging the periods of the signal in time domain gives an idea of what if it is possible to represent revolutions of the signal in the same frequency axis and then synchronously average them to obtain the frequency domain of the signal. Would it give a better result from traditional approach (Resampling+TSA+FFT)?

Having the same frequency axis means that the frequency range of the representation should be the same and the spacing of the frequencies should also be equal for all the periods. First part of the requirements are met by acquiring the raw signal at constant time intervals, so that the constant sampling rate bounds the range of the frequency axis. Because, when the signal Fourier transformed to obtain frequency representation, FFT sets the range of the result to $[0:\Delta f:fs-df]$, where Δf equal to fs/N . But, the second requirement is not always met, because the number of samples per revolution changes by changing the rotational speed of the shaft. Therefore, from the equation 3, step of the frequency axis Δf is not the same for all periods.

6.1 Introduction of DFT

With the motivation to solve the problems stated in chapter 2.2, new approach is built to test the representation of the signal in frequency domain. A Novel Frequency Domain algorithm is used to build this approach.

Process:

In order to implement this approach, only raw data acquired by traditional time based method is used in this approach. Keyphaser signal is used only to cut the signal into revolutions and then raw samples included in each every revolution is sent to new function called `my_fft` to calculate the Fourier Transform of the period. The idea is to plot the spectrum of the sampled signal by defining its discrete Fourier Transform as

$$FT_x(\Omega) = \sum_{j=1}^N X(t_j) \exp(-i\Omega t_j) \quad (24)$$

Unlike to `fft`, this more general approach takes into account the timing of the samples. In traditional `fft`, sampled are always assumed to be evenly spaced and also the frequency axis is sampled with constant increments. Frequency axis is also independent here to sample how we want, because every single sample frequency is independent from others. Therefore, it is possible to choose the frequency axis directly.

For instance, solution to the problem mentioned in chapter 2 can be seen in figure 20. Red spectrum is obtain by new approach and blue one is the result of traditional fft. In this example, length of the acquired data was 51 samples. Therefore, frequency range of interest [0:fs] was divided into 50 intervals by traditional fft. However, red one is obtained by dividing the frequency axis into 80 intervals thanks to new degree of freedom to choose the frequency. So, the problem related to Δf spacing of the Discrete Fourier Transform is solved.

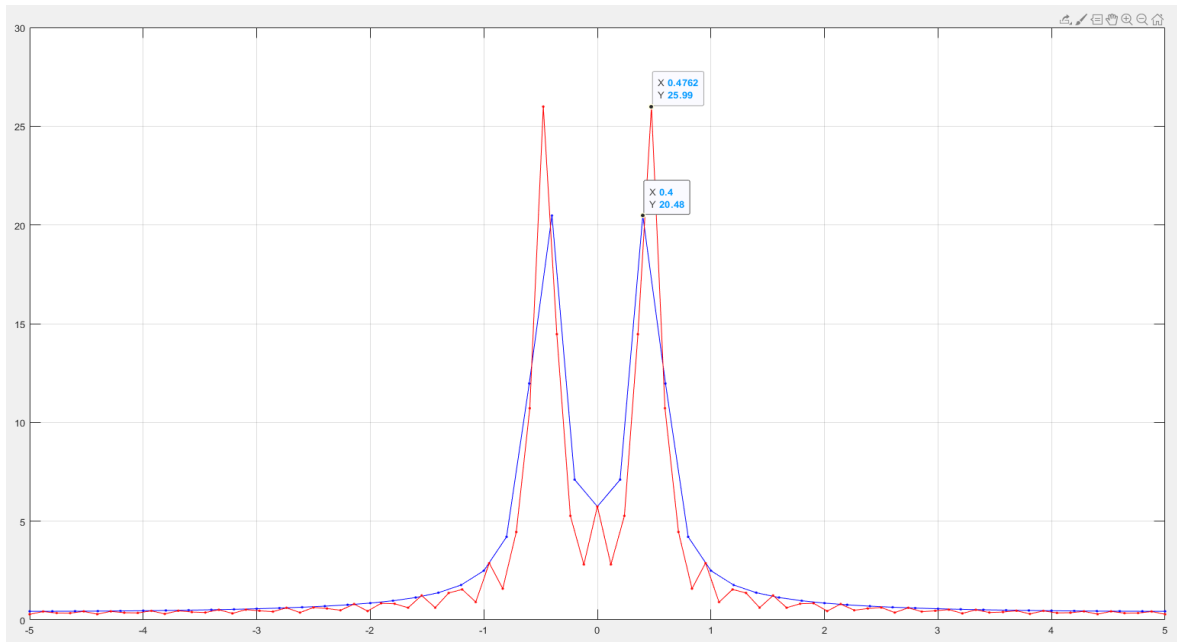


Figure 20. Comparison between *fft* and *Dft*

```

%function [Fc]= my_FFT(y,t,fs,N)
%this function is used to plot the spectrum of the given signal at
%customly sampled freq axes
%y is the signal samples
%t is the sampling time of the signal
%fs sampling freq
%N how many samples do we want to see in the freq axes of the spectrum plot

function [Fc,FT]= my_FFT2(y,t,fs,N)
Fc=[0:N-1]*fs/N;
FT = zeros(1,length(Fc));
for w=1:length(Fc)
    E=exp(-(-1)^0.5*t*(Fc(w)*2*pi));
    FT(w)=y*E;
end
end

```

As this approach gives new degree of freedom to choose the frequency axis independently from the signal itself, the idea of using Synchronous Average in Frequency Domain arises. Because, if we can calculate Discrete Fourier Transform in this way, then we can choose one frequency vector of our interest and find DFT for every revolution of the shaft in this range and then by synchronously averaging them it is possible to monitor the condition of the shaft.

6.2 Root Mean Square Error

In order to check the quality of the approach as well as the effect of variables like sampling rate, speed variation, number of samples per cycle in resampled data and number of cycles used to synchronous averaging, Root Mean Square Error (RMS-E) is introduced into procedure. The RMS (Root Mean Square) value of the signal is the normalized second statistical moment of the signal, meaning that the square root of the arithmetic mean of the squared values of the signal:

$$RMS = \sqrt{\frac{\sum_{i=1}^N [x(i)]^2}{N}} \quad (22)$$

RMS-E is the RMS value of the difference vector of the signal with respect to reference one.

$$RMS - E = \frac{RMS(det - estimated)}{RMS(det)} \quad (23)$$

RMS-E

```

% test of COT
fs=10000; %Hz or sps
T=20; %s
t=[0:1/fs:T];
yn = 0.1*randn(size(tt));
s=[1:0.2:3];
%% Zero crossing time
for ii=1:length(s)
yp=chirp(tt,10,T,10*s(ii));
y= yn+yp;
% plot(tt,y,'b*-'),xlabel('t[s]'),ylabel('y(t)'), title('Sweep')
kptempiw2= keyphasorDEF(yp,tt,0);
% Traditional Approach (Resampling+TSA+FFT)
K=150;
phi = 2*pi;
tneww2= resampleDEF(kptempiw2,phi,K);
sgn_per = interp1(tt,yp,tneww2,'spline');
% TSA
nn=1;
[Sh_per]= TSA(sgn_per,K,nn);
% FFT
% NFFT=2^nextpow2(length(Sh_per));
NFFT=length(Sh_per);
XPper=fft(Sh_per,NFFT)/length(Sh_per);
SGNP=2*abs(XPper(1:NFFT/2));
%
sgn_new = interp1(tt,y,tneww2,'spline');
[Sh_new]= TSA(sgn_new,K,nn);
% FFT
% NFFT=2^nextpow2(length(Sh_per));
Xper=fft(Sh_new,NFFT)/length(Sh_new);
SGN=2*abs(Xper(1:NFFT/2));
f_SGNper=[0:NFFT-1]*K/NFFT;
% Normalized approach
l = length(kptempiw2)-1;
A_DFT = zeros(l,NFFT/2);
n2=floor(nn/2);
% tic
for j=1:l/nn
i=nn*(j-1)+1;
dt= kptempiw2(i+1+n2)-kptempiw2(i+n2);
y11=y(tt>=kptempiw2(i) & tt<kptempiw2(i+nn));
t1=(tt(tt>=kptempiw2(i) & tt<kptempiw2(i+nn))-kptempiw2(i))/dt;
m(j)=length(y11);
[Fc, FT]= my_FFT2(y11,t1',K,NFFT);
A_DFT = A_DFT+FT(1:NFFT/2)/length(t1);
end
A_DFT = 2*abs(A_DFT/j);
%RMS error of FFT
Df=SGNP-SGN;
Ef=Df*Df'/(SGNP*SGNP');
ef(ii)=100*sqrt(Ef);
% RMS error of DFT
Dd=SGNP-A_DFT;
Ed=Dd*Dd'/(SGNP*SGNP');
ed(ii)=100*sqrt(Ed);
end
figure;
plot(s,ef,'b');
hold on
plot(s,ed,'r');

```


Effect of Sampling Rate

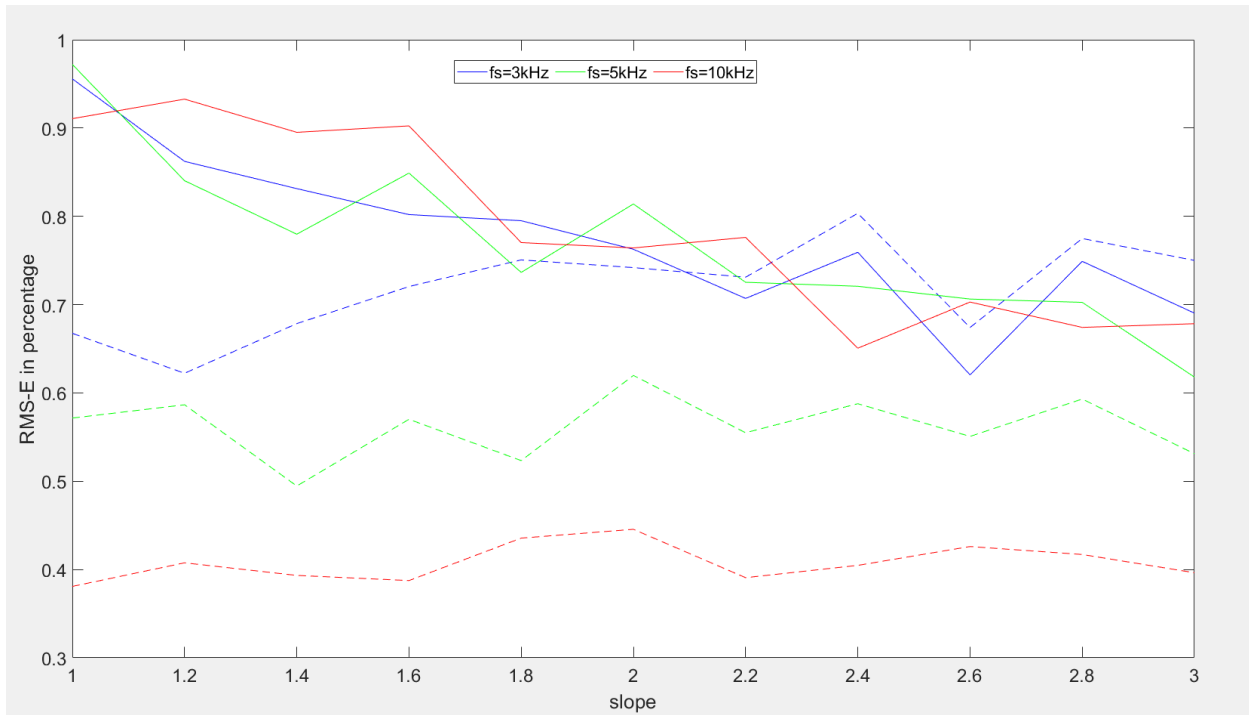


Figure 21. RMS-E at different sampling rates. Solid lines are traditional approach and dashed lines are Synchronous Averaging in frequency domain

From figure 21, it is possible to see different behaviour from two approaches.

Firstly, traditional approach shows no effect on sampling rate with almost the same trend as fs increases (minor differences are due to the added random noise). However, error in new approach gets lower by increasing the sampling rate.

Secondly, as slope increasing, trend in traditional approach is decreasing meaning that higher speed variation reduces the error in FFT after TSA procedure while proposed method shows about constant error with respect to slope.

Nevertheless, it is noteworthy that all errors are less than 1%, even less than 0.5% in the case of Synchronous averaging in frequency domain.

Effect of Number of samples in Resampling

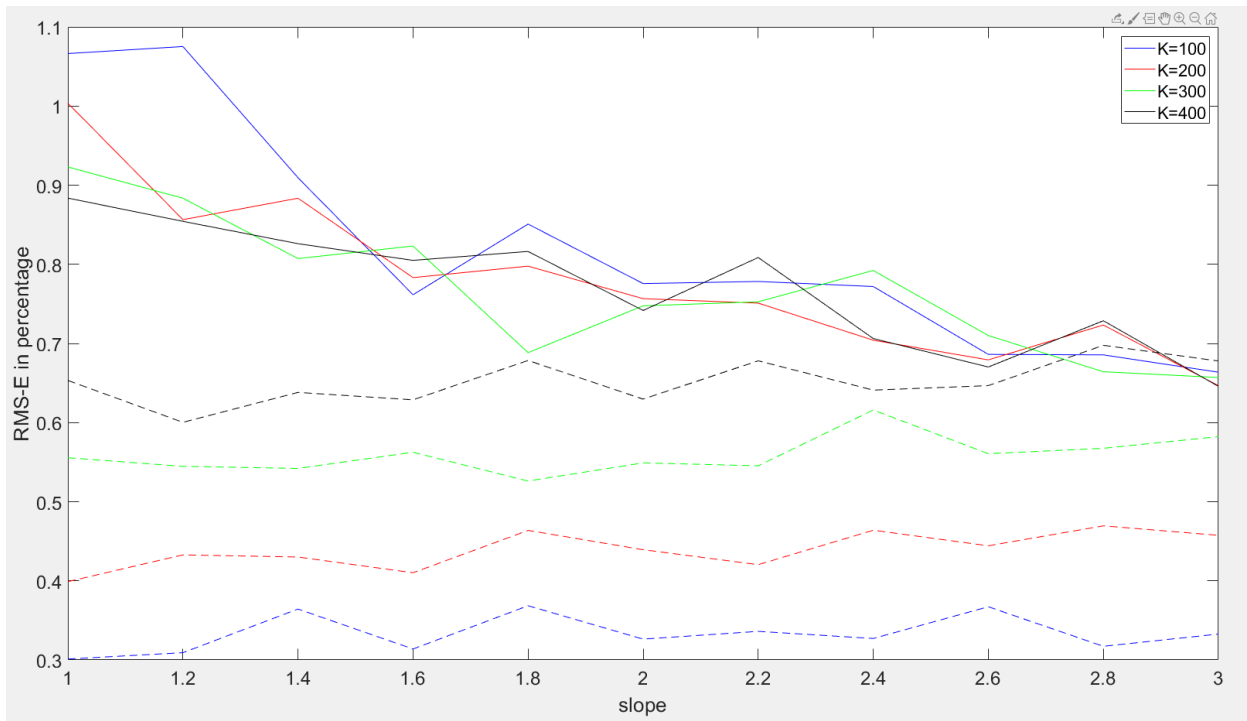


Figure 22. RMS-E at different K (number of samples per cycle used to resample). Solid lines are traditional approach and dashed lines are Synchronous Averaging in frequency domain

Also in figure 22, decreasing trend of traditional approach is noticeable. Like the sampling rate, K (number of samples per cycle used to resample) does not have much influence on the traditional method.

On the other hand, increasing the K increased the error in the new approach. It is believed to be because the synchronizing in frequency domain is performed to plot equivalent spectrum of traditional approach and in traditional approach, K decides until which order the spectrum is built. That is why at higher orders, new approach may show higher error. Other than that, K does not have any direct influence on Synchronous Averaging in frequency domain. So, if degree of freedom coming with approach is used to limit the frequency range and by increasing the sampling rate very low error can be achieved.

Also in this case error is less than 1% in traditional approach and less than 0.5% in new approach.

Effect of Number of cycles used to Synchronous Averaging

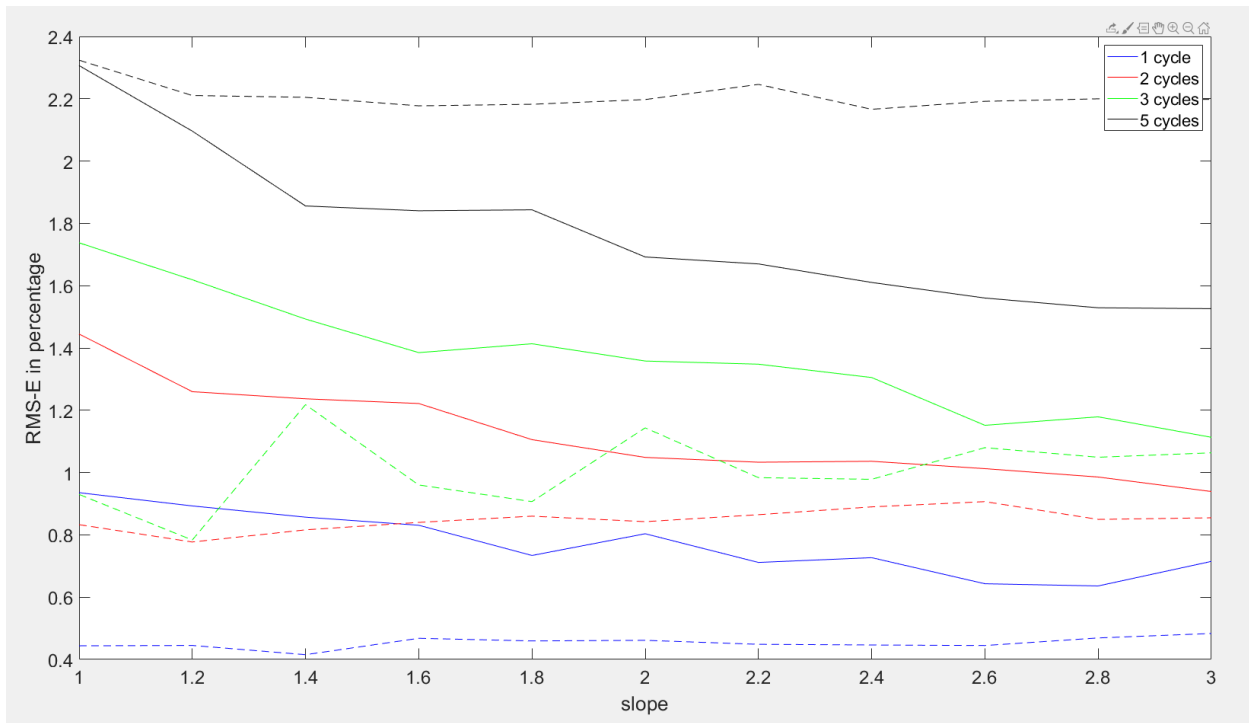


Figure 23. RMS-E at different n (number of cycles used to SA)

In this case, both methods (solid lines belongs to traditional approach and dashed ones are SA of DFT) show the same behaviour by increasing with n . It is because in traditional approach speed is assumed to be constant within a cycle and if many cycles are used for each synchronization, speed variation within a calculation may be too much and it leads to a higher errors. It is also the case for SA of DFT.

This shows some limits to increase number of cycles to SA. Up to 3 cycles, both methods show acceptable errors (less than 1% in new approach and about 1.5% in traditional approach). However, if it is increased more, error becomes very high.

7. Discrete Fourier Transform

7.1 Implementation on real data

In order to check further the performance of this approach, it is tested on couple of real cases.

7.1.1 PHM

Acquisition system

- Channel 1 is the input side Accelerometer
- Channel 2 is the output side Accelerometer
- Channel 3 is the Tachometer Signal: 10 pulse per revolution

Sample Rate: 66,666.67 Samples per Second (200 KHz/3). Data were collected at 30 Hz shaft speed.

						IS	Input Shaft	:IS	Input Side			
						ID	Idler Shaft	:OS	Ouput Side			
	Part					OS	Output Shaft					
	Gear			Bearing							Shaft	
Case	32T	96T	48T	80T	IS:IS	ID:IS	OS:IS	IS:OS	ID:OS	OS:OS	Input	Output
Spur 1	Good	Good	Good	Good	Good	Good	Good	Good	Good	Good	Good	Good
Spur 2	Chipped	Good	Eccentric	Good	Good	Good	Good	Good	Good	Good	Good	Good
Spur 3	Good	Good	Eccentric	Good	Good	Good	Good	Good	Good	Good	Good	Good
Spur 4	Good	Good	Eccentric	Broken	Ball	Good	Good	Good	Good	Good	Good	Good
Spur 5	Chipped	Good	Eccentric	Broken	Inner	Ball	Outer	Good	Good	Good	Good	Good
Spur 6	Good	Good	Good	Broken	Inner	Ball	Outer	Good	Good	Good	Imbalance	Good
Spur 7	Good	Good	Good	Good	Inner	Good	Good	Good	Good	Good	Good	Keyway Sheared
Spur 8	Good	Good	Good	Good	Good	Ball	Outer	Good	Good	Good	Imbalance	Good

Provided data has been obtained for several conditions of gears and bearings. Tests in the following is done with case one and two where there is a chipped tooth on the gear of the input shaft. Therefore, in the frequency domain representation,

In order to check the performance of the approach, both ways are built to compare side-by-side.

First, traditional method is built. Procedure includes obtaining keyphasor signal by zero crossing method and then followed by resampling technique so as to resample the signal in constant shaft angle increments, then performing TSA and finally FFT to obtain order domain representation.

Then, equivalent My_FFT approach is built. It can be applied in two ways, simply in frequency domain or in order domain. In order to get the frequency domain representation, first of all, the length of the frequency axis and the spacing Δf should be chosen before the process. Because, it should be constant for all the periods. Then, raw data is

cut into revolutions and sent to `my_fft` function to process the FT of the signal by new approach. `my_fft` function uses four inputs as shown.

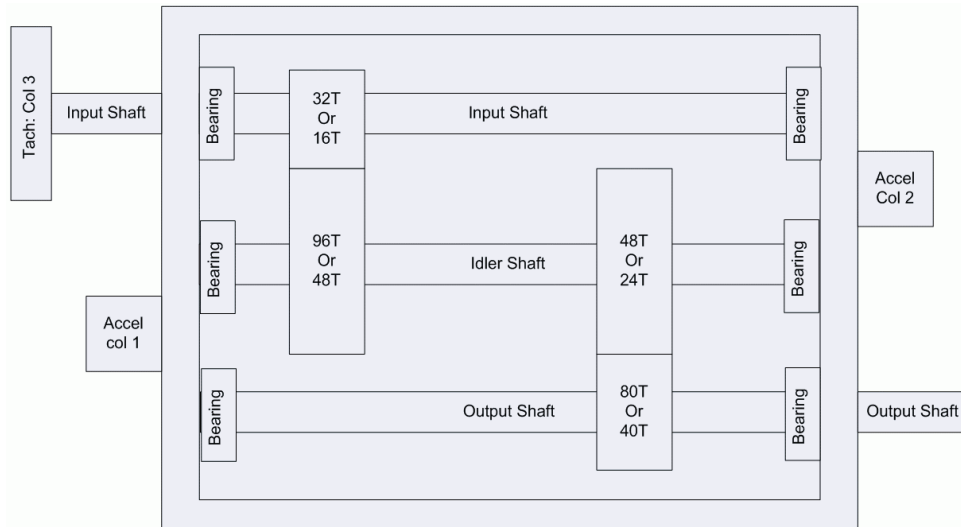


Figure 24. PHM09 schematics of the apparatus.

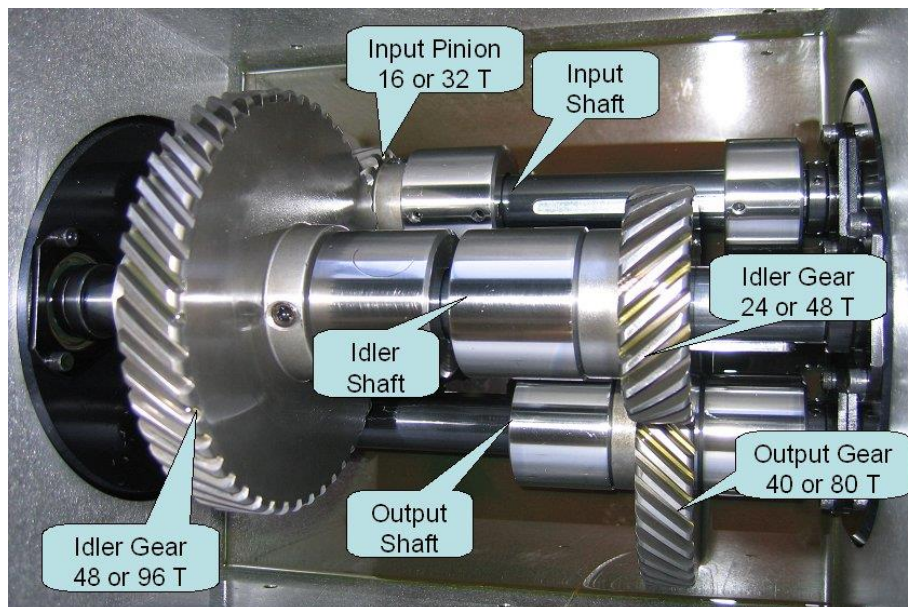


Figure 25. Detailed gearbox picture

Inputs for `my_fft` are:

- 1st: raw samples of the revolutions to be processed
- 2nd: times at which those samples are acquires
- 3rd: frequency range of interest
- 4th: N number that decides the Δf steps

Outputs of the my_fft function are [Fc] that is the frequency range of on which the test is performed and [FT] that is the Discrete Fourier Transform of given raw samples.

Once [FT] is calculated for each period, they are then averaged to find truly periodic part of the Spectrum and to reduce the random part related to the random noise or bearings. So, the Synchronous Averaging in Frequency Domain is performed.

```

%% Zero crossing time
kptempiw2= keyphasorDEF(yT,t,0);
% plot(kptempiw2,zeros(size(kp tempiw2)), 'ro');
%% Traditional Approach (Resampling+TSA+FFT)
tic
% Resampling
K=180;
phi = 2*pi;
tneww2= resampleDEF(kptempiw2,phi,K);
sgn_new = interp1(t,y2,tneww2, 'spline');
% figure;plot(t,y2, 'b*-');hold on;
% plot(tneww2,sgn_new, 'rs', 'linewidth', 3)
% TSA
nn=1;
[Sh_per]= TSA(sgn_new,K,nn);
% FFT
% NFFT=2^nextpow2(length(Sh_per));
NFFT=length(Sh_per);
Xper=fft(Sh_per,NFFT)/length(Sh_per);
SGN1per=2*abs(Xper(1:NFFT/2));
f_SGNper=[0:NFFT-1]*K/NFFT;
%
figure;plot(f_SGNper(1:NFFT/2),SGN1per, 'b');
title('FFT vs DFT','FontSize',font)
xlabel('Orders','FontSize',font)
hold on
toc

%% Normalized approach
% figure;
l = length(kptempiw2)-1;
NFFT =8*2^10;
AFT = zeros(1,NFFT);
tic
for j=1:l/nn
    i=nn*(j-1)+1;
    dt= kptempiw2(i+1)-kptempiw2(i);
    y11=y2(t>=kptempiw2(i) & t<kptempiw2(i+nn));
    t1=(t(t>=kptempiw2(i) & t<kptempiw2(i+nn))-kptempiw2(i))/dt;
    m(j)=length(y11);
    [Fc, FT]= my_FFT2(y11,t1',fs/220,NFFT);
    AFT = AFT+FT;
end
AFT = AFT/j/length(t1);
% figure;
plot(Fc,2*abs(AFT), 'r');
toc

```

In order to obtain order domain representation by this approach, signal first needs to be normalised. To this regard, length of the period dt is defined by the interval between two consecutive keyphasor times. Then, sampling times are normalised by dividing by dt . Therefore, each period is set to one. So the speed normalised representation is obtained

There are some factors that play a main role in the quality of the representation. For instance, number of cycles used per each DFT calculation. It means that, signal is not separated into single cycles but multiple cycles are used for each calculation. When the number of cycles per iteration is increased, result is obtained in better resolution. However, as it is mentioned in chapter 6.2, as we increase the number of cycles per SA, error also increases. Therefore, only some iterations (1, 2, 3 cycles per SA) are represented and compared in the following figures in order to be in acceptable range

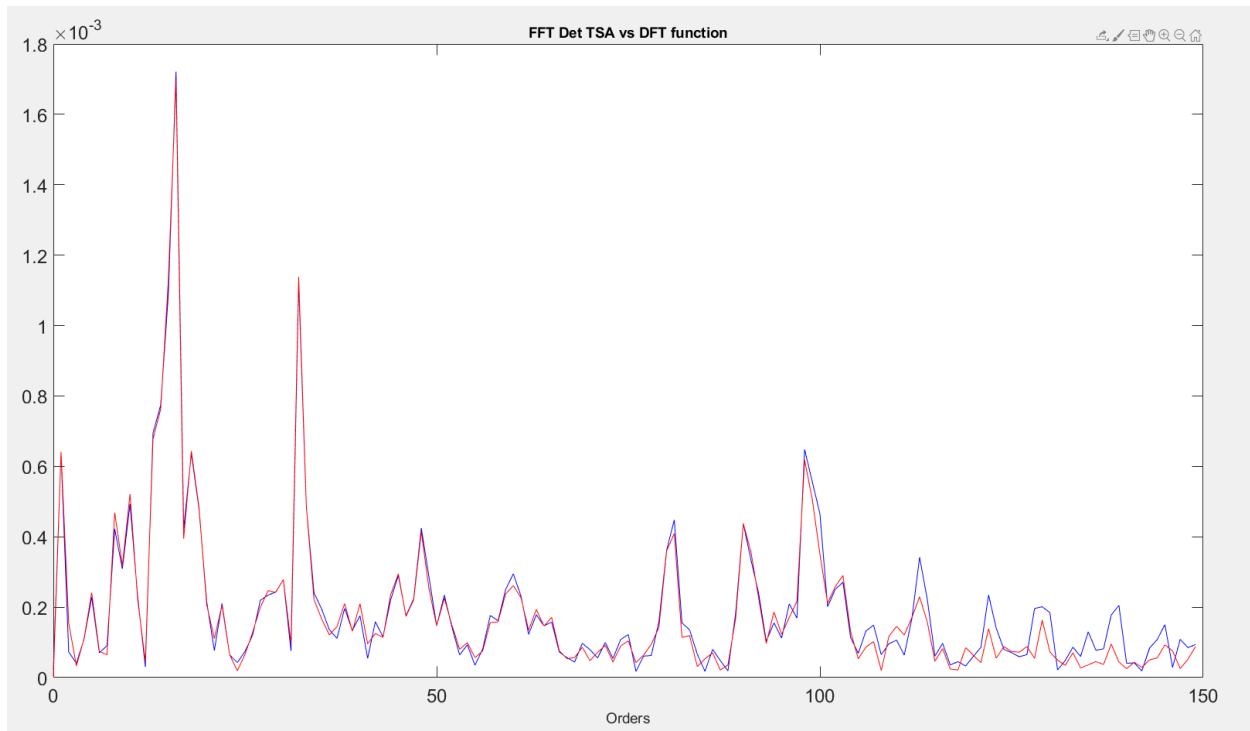


Figure 26. One cycle is used to average.

Figure 26 represents the spectrum of acquired data with Traditional approach (blue line) and SA of DFTs (red line) with only one cycle is used to average. It is visible that traditional fft after TSA method and SA after DFT method show almost identical results in order domain with coarse steps. It is because only one cycle length is used.

$$f=[0:N-1]*fs/N=[0:df:fs-df] \quad \text{where } df=fs/N$$

if we consider 1 period at a time to Fourier Transform then N is equal to number of samples per cycle which means:

$$f_s = N/T = N \cdot f'$$

where $f' = 1/T$ is shaft speed. Then:

$$df = f_s/N = f'$$

It shows that frequency resolution represents orders of the shaft speed. Moreover, in most of the cases, it is what is needed from Rotating Machinery Diagnostics. Because if there is a fault at gear tooth, the peak appears at Gear-Mesh frequency which is equal to $f_m = z \cdot f'$ (z is number of teeth of the gear). It appears at gear teeth number in order domain. It is also visible in figure 27, where the step plot of previous case is represented in order to better see the frequency resolution. Spur 2 (chipped tooth) case was chosen to apply the algorithm and noticeable peak appeared at Gear-Mesh frequency.

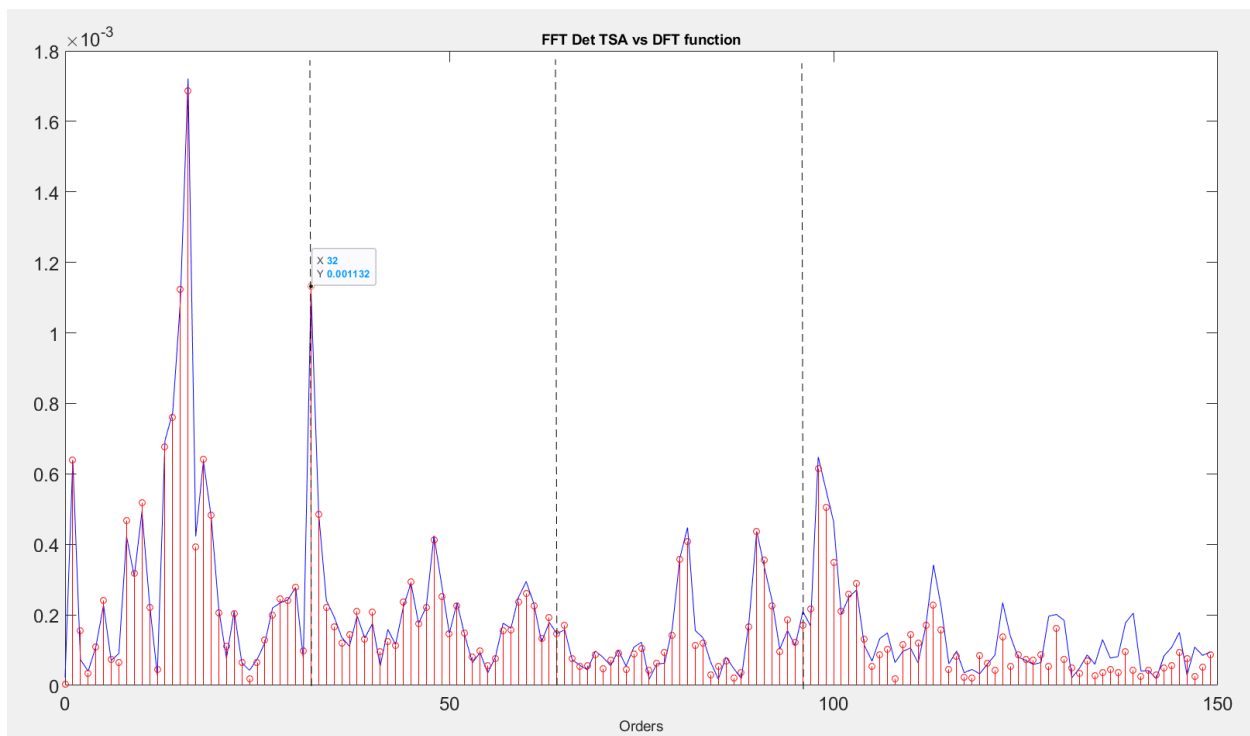


Figure 27. Stem plot with gear Mesh.

In the next figures spectrum where 2 and 3 cycles per SA used is represented. It is easy to notice that peaks are much sharper thanks to increased number of periods lead to use smaller steps in order spectrum and calculate internal points as well.

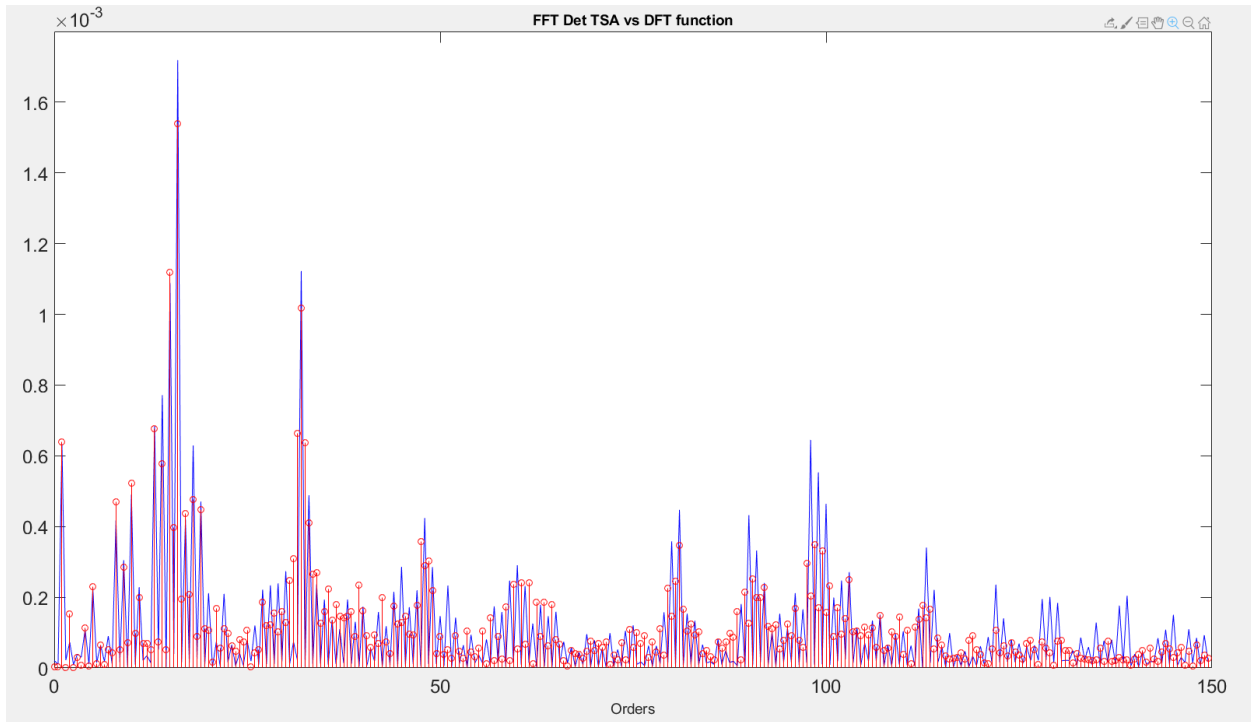


Figure 28. 2 cycles are used to average.

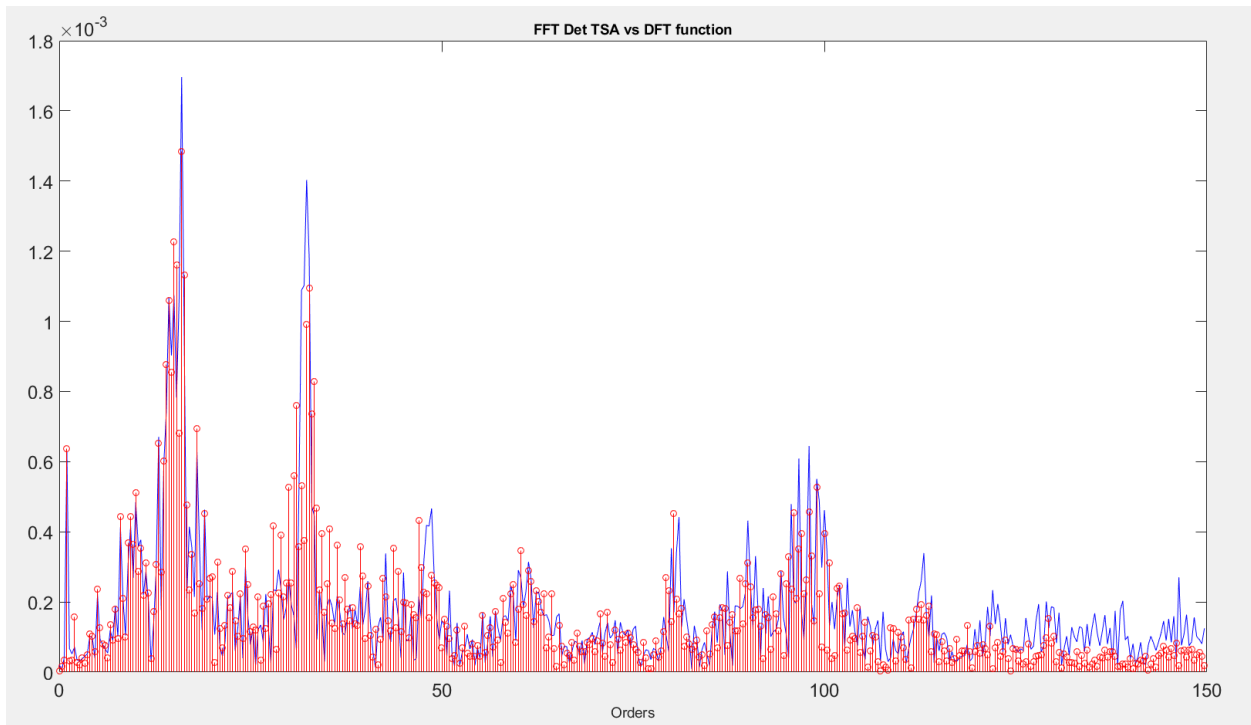


Figure 29. 3 cycles are used to average.

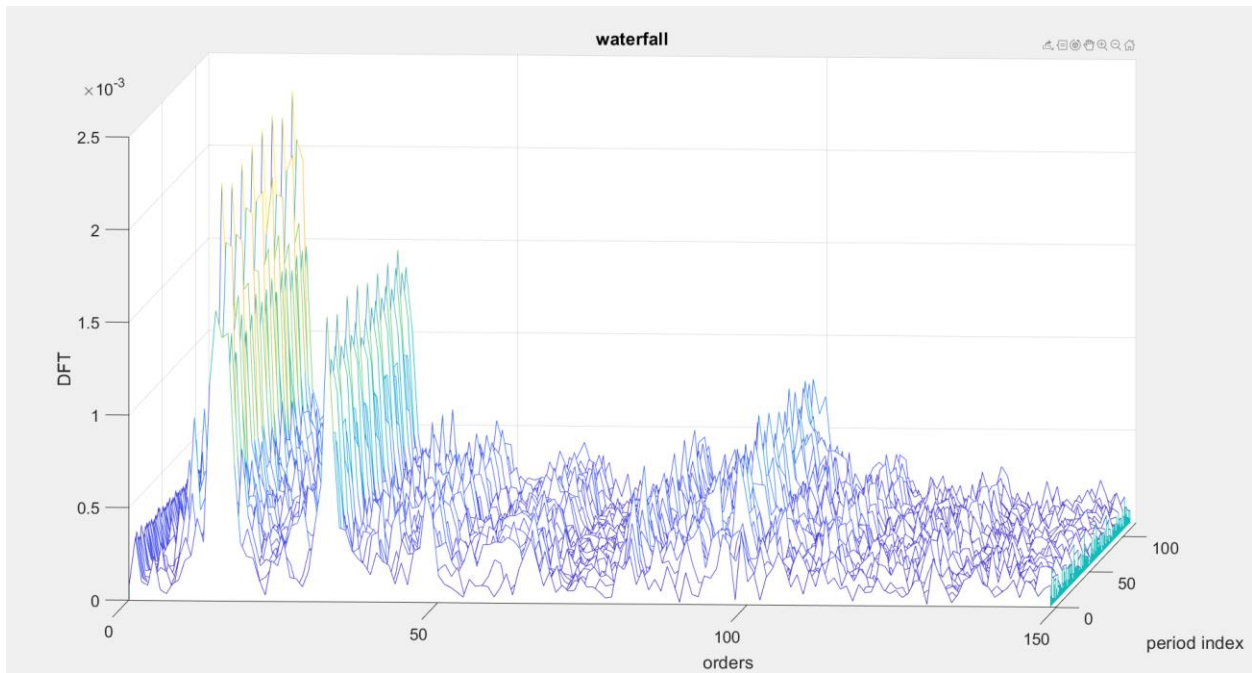


Figure 30. Waterfall plot of periods

It is also interesting to see Waterfall plot of the signal where the spectrum of each period is represented. Figure 26 spectrum plot is generated by averaging spectrums of each periods. Therefore, the procedure is called Synchronous Averaging in Frequency domain.

7.1.2 Safran Contest

In the second case, formula is tested on the data that is provided in the Safran contest organized during the International Conference Surveillance 8, October 20-21, 2015, at the Roanne Institute of Technology, France. Vibration data is acquired from a civil aircraft that is in a transient operating mode. Figure 22 shows a general overview of the engine with the damaged bearings and sensors locations. Also for this case, the procedure is the same with one applied to PHM contest.

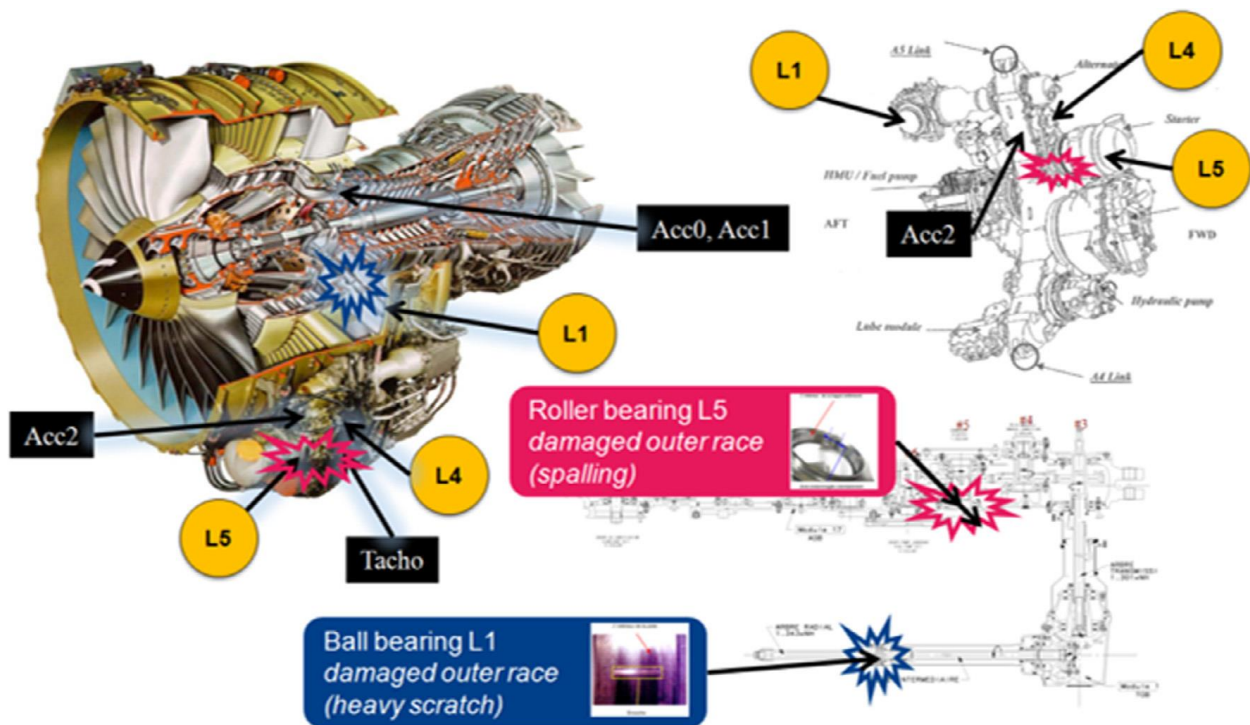


Figure 31. General overview of the engine and the accessory gearbox.

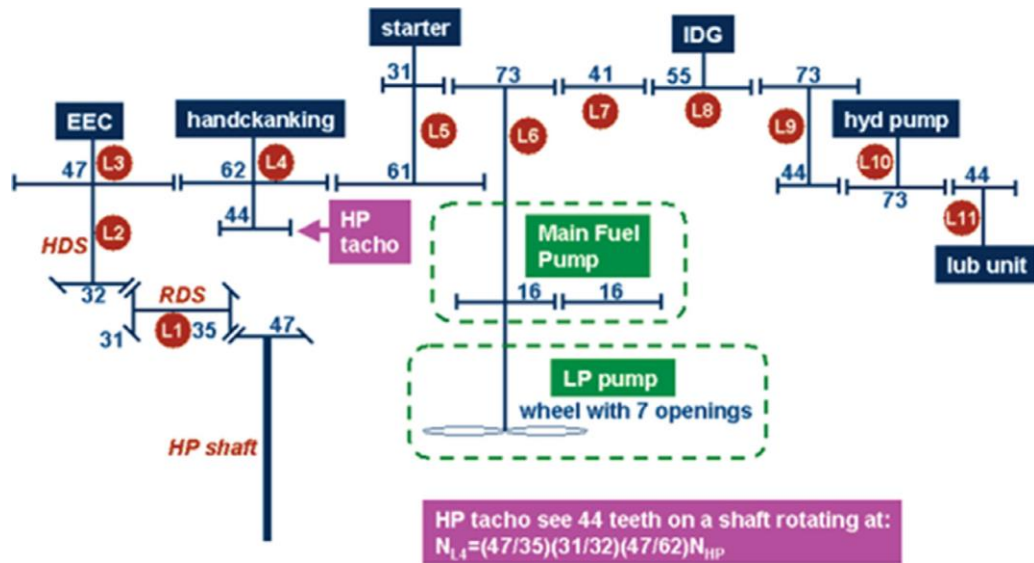


Figure 32. Kinematics of the gearbox

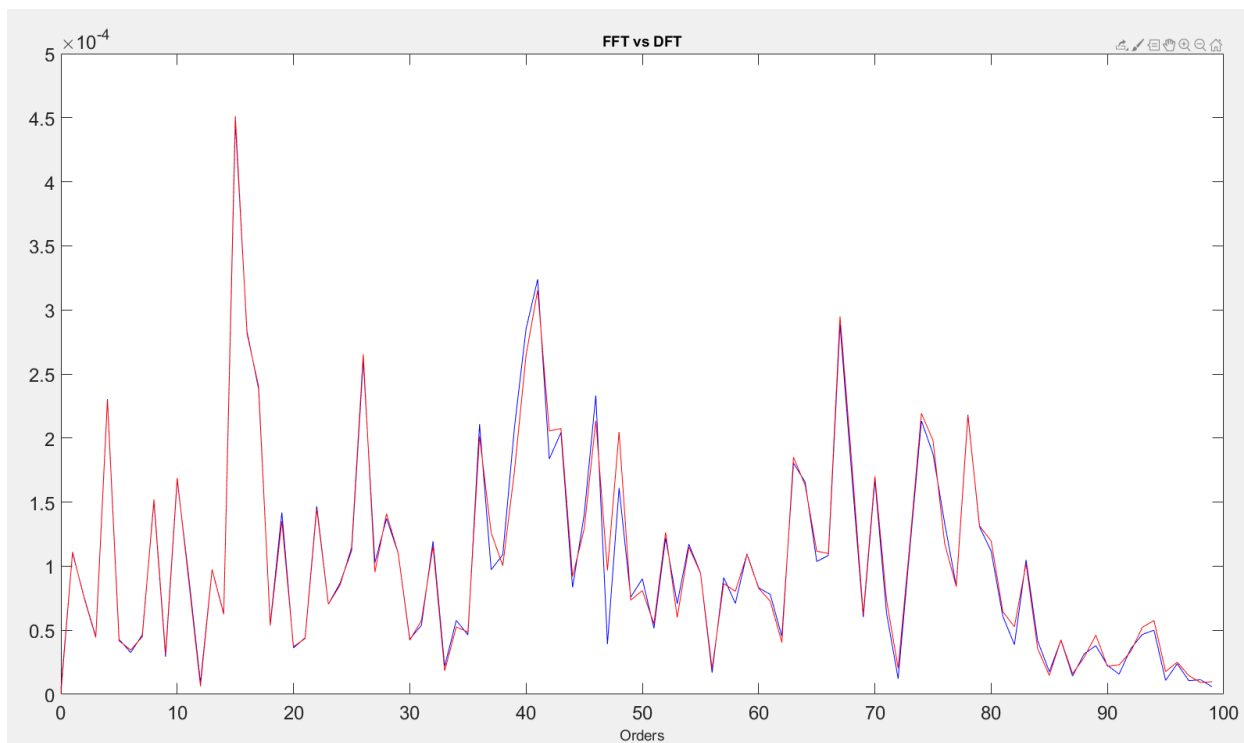


Figure 33. One cycle is used

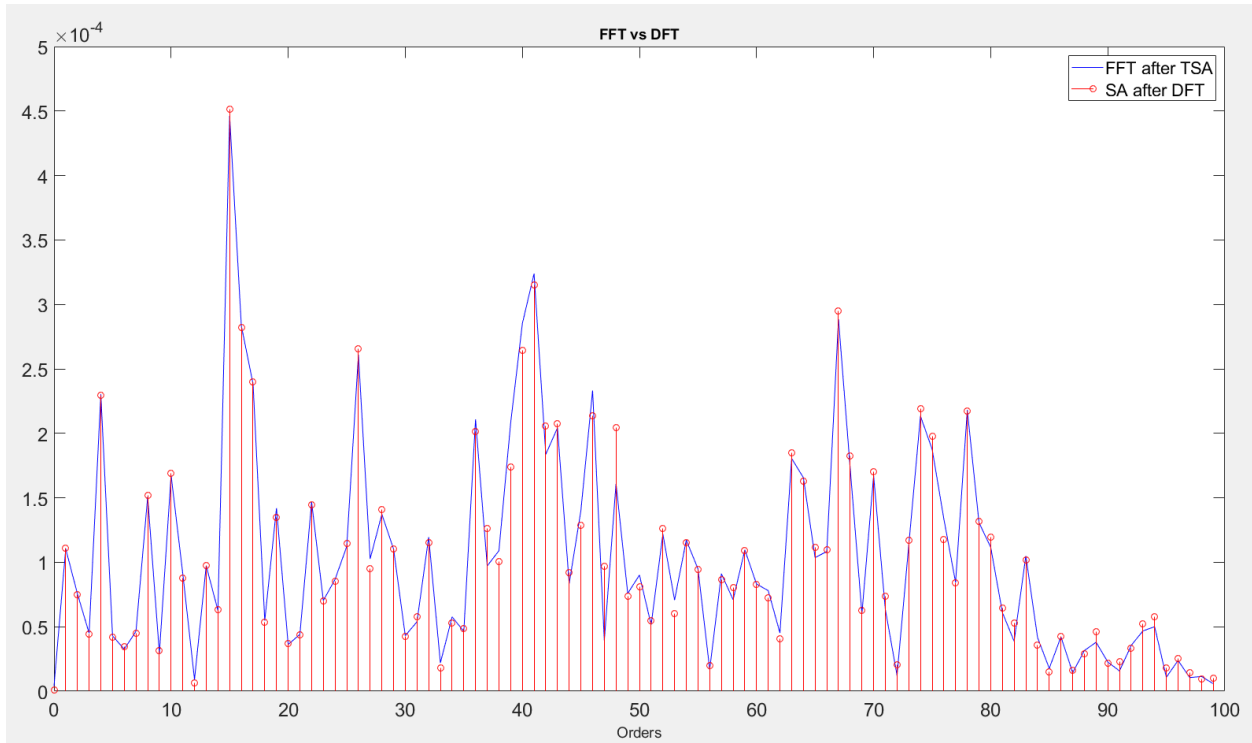


Figure 34. Stem plot

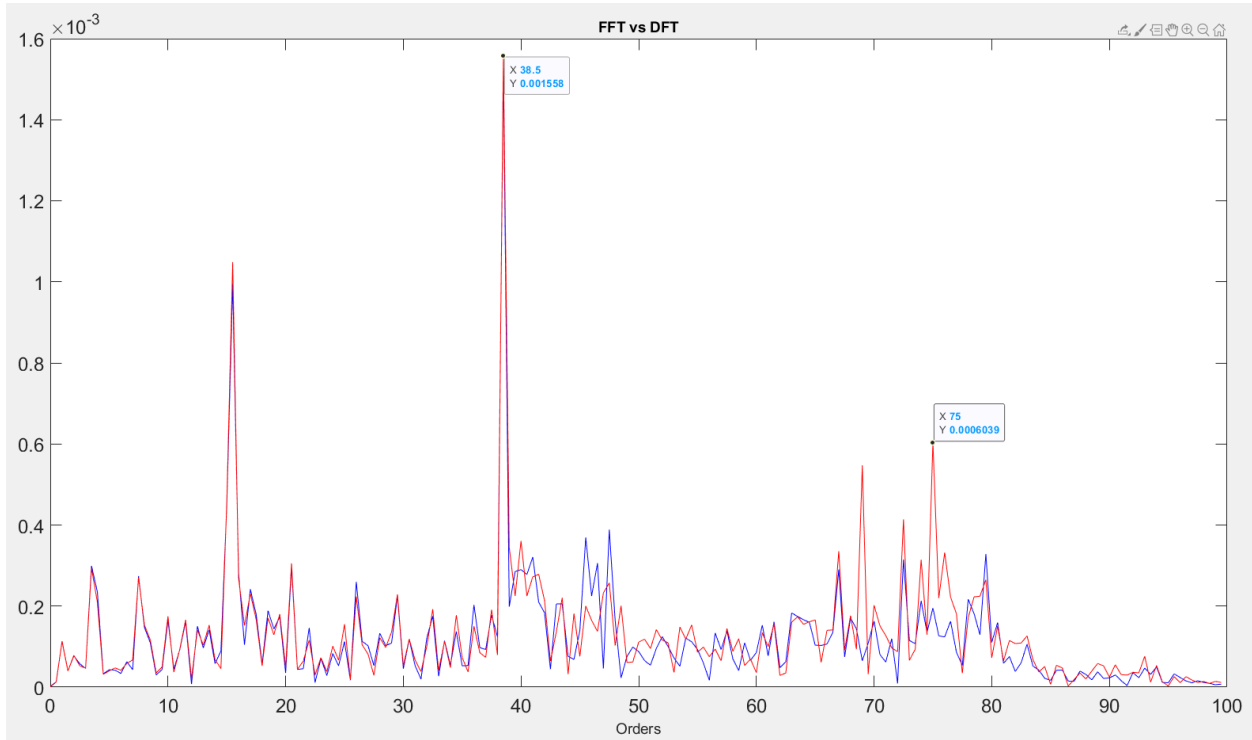


Figure 35. Two cycles are used

In figure 35, clear advantage of using more than one cycle in the synchronizing process is shown. In figure 34 peaks are not so evident, but as soon as it is increased to two cycles some peaks are sharply rose. It is also noteworthy that even the speed is changing in this example, peaks are all aligned in order domain making the condition monitoring easy (figure 36).

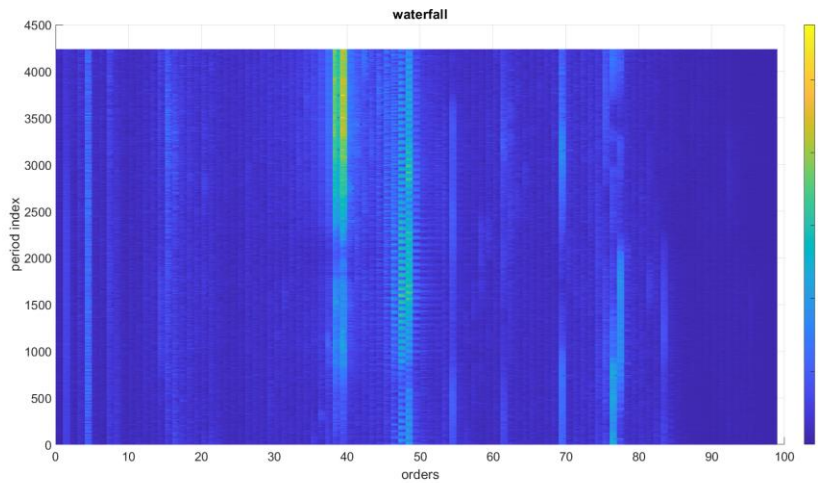
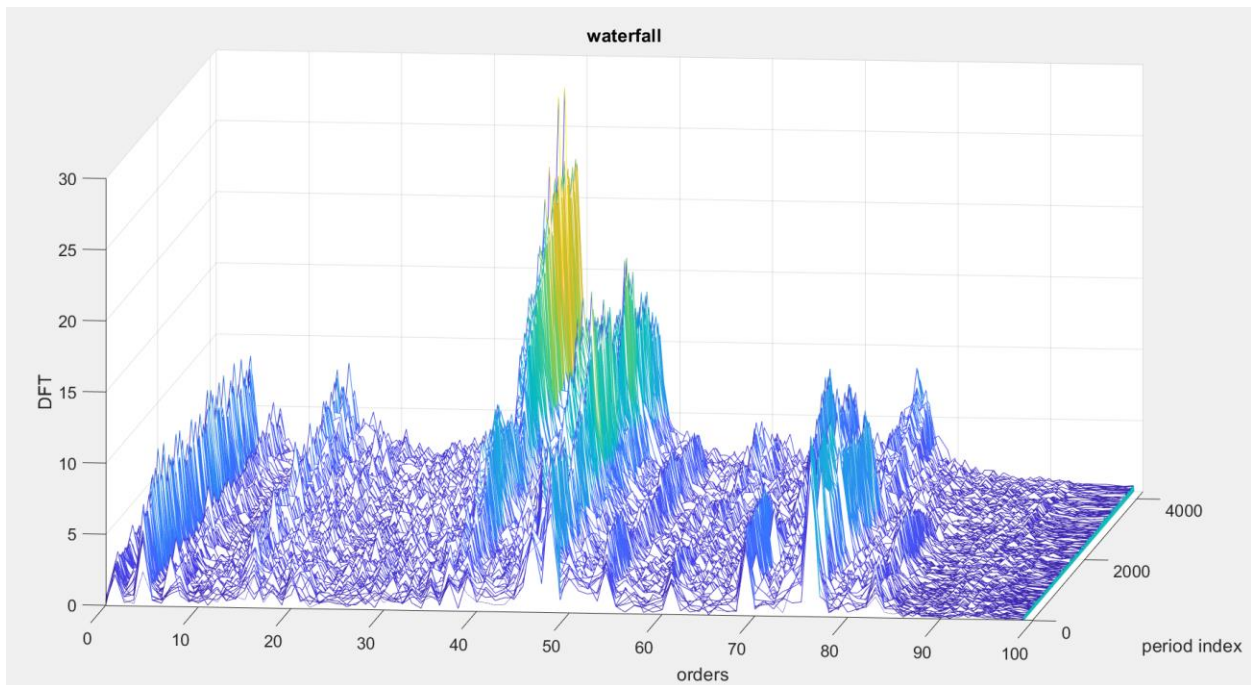


Figure 36. Waterfall plot of the orders

8. Conclusion

In this work, equivalent procedure of Traditional Condition Monitoring is built by using the Novel Frequency Domain Algorithm and tested for quality, arguments for and against the procedure is discussed.

Traditional approach (COT+TSA+FFT) of Rotating Machinery Condition Monitoring is believed to be one of the most efficient techniques to extract the deterministic part of the gearbox signal, the one related to gears, from non-deterministic one coming from noise and bearings in time domain.

Proposed algorithm is based on Synchronous Averaging in frequency domain. Clear advantages at high sampling rates as well as degree of freedom to choose the frequency axis of the procedure shown in the tests makes it interesting to investigate further.

However, there are some downsides of this method too. The most important one is the computation time. As traditional method uses Fast Fourier Transform, its computation time is much faster than the one proposed in this work, which uses Discrete Fourier Transform.

Based on the requirements and working conditions, one can be recommended over the other one.

Reference

Fyfe, K. R., & Munck, E. (1996). *Analysis of computed order tracking*

Randall, R. B., & Antoni, J. (2010). *Rolling element bearing diagnostics - A tutorial*

Ali Moshrefzadeh, Alessandro Fasana. *The Autogram: An effective approach for selecting the optimal demodulation band in rolling element bearings diagnosis*

Ali Moshrefzadeh, Alessandro Fasana, Jérôme Antoni. *The spectral amplitude modulation: A nonlinear filtering process for diagnosis of rolling element bearings*

Jérôme Antoni, Ge Xin, Nacer Hamzaoui. *Fast computation of the spectral correlation*

Garg, M. (2003). *Linear Prediction Algorithms*

Alessandro Paolo Daga, Alessandro Fasana, Stefano Marchesiello, Luigi Garibaldi. *The Politecnico di Torino rolling bearing test rig: Description and analysis of open access data*

Jeffrey D. Scargle. *STUDIES IN ASTRONOMICAL TIME SERIES ANALYSIS. II. STATISTICAL ASPECTS OF SPECTRAL ANALYSIS OF UNEVENLY SPACED DATA*

Edward Lee and Pravin Varaiya. *Structure and Interpolation of Signals and Systems*

Luis F. Chaparro. *Signals and Systems Using MATLAB*

J. Antoni*, R.B. Randall. *Unsupervised noise cancellation for vibration signals: part II—a novel frequency-domain algorithm*

S. Braun. *The synchronous (time domain) average revisited*