Step-by-step algorithm for the simulation of faulted bearings in non-stationary conditions

G. D'Elia [1], **M. Cocconcelli** [2], **E. Mucchi** [1], **R. Rubini** [2], **G. Dalpiaz** [1]
[1] University of Ferrara, Engineering Department,
Via Saragat 1, 44100 Ferrara, Italy
e-mail: **gianluca.delia@unife.it**

[2] University of Modena and Reggio Emilia, Department of Science and Engineering Methods,
Via Amendola 2 - PAd. Morselli, 42122, Reggio Emilia, Italy

Abstract
The early validation of a new diagnostic technique on a proper simulated signal is crucial, in order to provide a feedback to the researcher and increasing the chances of getting a positive result in the real case-studies. While dozens of comprehensive models of ball bearing have proposed in literature so far, the complexity of these models accordingly increased. As supposed, the scientific papers just outline the theoretical foundations of assumptions and features of the model, leaving the reader the task of converting all in lines of code. The aim of this paper is to detail step-by-step an analytical model of faulted bearing that the reader could freely and immediately use to simulate different faults and different operating conditions. It is based on the model proposed by Antoni in 2007 and the features available are the following: selection of the location of the fault, stage of the fault, cyclostationarity of the signal, random contributions, deterministic contributions, effects of resonances in the machine and working conditions (stationary and non-stationary).

1 Introduction

Rolling bearings, together with gears, are one of the most studied components. They are common components in mechanical design and they allow the relative motion between two or more elements of the machine. Unfortunately, the continuous movement between the parts of the bearing leads to wear phenomena and subsequent failure. The degradation of the bearing conditions can be revealed and monitored analysing the vibration signal produced by the contact among the bearing elements. There are other types of techniques to determine the state of health of the bearings, such as monitoring the temperature or analysing the chemical content of the lubricant, but the vibration analysis is, de facto, the main technique used in condition monitoring, despite the ease with which noise and disturbances may enter in the measurement. So far, thousand of algorithms have been published in literature trying to reject disturbances and to obtain a clear and telltale signal to assess the health status of the bearing [1]. All these publications usually provide results on both simulated signals and real measurements, more rarely on only one of those. It is a matter of fact that the availability of a real test-bench is not so common, and this is proven by the number of scientific papers validated on few on-line available data centres (i.e. the Case Western University) providing real measurement data. On the contrary simulated signals are always available, since they are created on the same software for scientific computing used in the post processing. The main advantage of a simulated signal is to avoid the complexity of a real environment, focusing only on the main contributions the developer decided to include. The main drawback is that a too simple model may be too far from reality, making the proposed algorithm not useful. The foundation of a faulted bearing simulation signal is the model proposed by McFadden and Smith [2–4]. The bearing is modelled as an epicyclic gear, where the inner ring is the sun gear, rolling elements are the planet gears, the outer ring is the annular gear and the cage is the planet carrier. This simple but powerful model allows the computation of characteristics fault frequencies which are the fingerprints of a damage on the bearing. Moreover the model took into account also the modulation effects due cyclic passage of the rolling elements on the load zone. Su and Lin [5] studied the models under variable load due to shaft and roller errors. The "gearbox" model for the bearings has a main limitation: the contact among the bearing components is supposed to be a pure rolling contact, while some slippery effect is always present due to the presence of the cage. Ho and Randall [6] proposed to model the bearing fault vibrations as a series of impulse responses of a single-degree-of-freedom system, where the timing between the impulses has a random component simulating the slippery effect. The next fundamental contribution to the modelling of bearings came from the works of Antoni and Randall [7, 8]. Starting from the work of Gardner [9], Antoni and Randall proposed to model the vibration signal from a ball bearing as a cyclostationary signal, i.e. a random process with a periodic autocorrelation function. Cyclostationarity better describes the effect of slippery and has paved the way for later development. Most recent developments regard the modelling of the vibration signal in non-stationary conditions [10], i.e. taking into account the speed or load variations in the working conditions of the machine. Unfortunately, as the proposed models have become more detailed the implementation of the algorithms has become more complex. If the model of McFadden could be easily taught in an introductory course at an engineering school, concepts like cyclostationarity and non-stationary conditions are hardly present in advanced courses at engineering faculties. As a consequence, it could be a gap between the theoretical description of a vibration signal and the algorithm implemented to generate that vibration signal on a computer. A wrong implementation leads to wrong simulated signals used to test diagnostics procedures. The aim of this paper is to provide a detailed step-by-step algorithm for the simulation of the vibration signal provided by a faulted ball bearing. The script is developed in Octave environment, an open source high-level interpreted language, primarily intended for numerical computations and quite similar to Matlab. The base of this model is the one proposed by Antoni [11] with few improvements. Details on the characteristics that the model takes into account will be explained in the next sections. The final goal is to start a discussion with the readers to define a bearing model that can be used as a benchmark, recognized by the scientific community. The paper is structured as follows: Section 2 covers the theoretical background of the bearing model and the numerical implementation. Section 3 focuses a numerical example, showing the output results of the proposed algorithm. After the conclusions in section 4, the appendix A lists the script of the algorithm as Octave code.

2 Vibration signal model

2.1 Theoretical background

At first glance, the vibration signal model of a localized fault in a rolling element bearing could be considered as the repetition of impact forces when a defect in one bearing surface strikes a mating surface, which may excite resonances in the bearing and in the machine. The repetition frequency of these impacts uniquely depends by the location of the defect, be it on the inner race, outer race, or one of the rolling elements. Even if it is likely that the actual response will include several resonances, for simplicity, it will be assumed in the remaining discussion that only one resonance occurs.

The vibration signal of a localized fault in a rolling element bearing can be reasonably modelled as [8, 11]:

$$x(t) = \sum_{i=-\infty}^{+\infty} h(t - iT - \tau_i)q(iT) + n(t) \tag{1}$$

where $h(t)$ is the impulse response to a single impact as measured by the sensor, $q(t)$ takes into account the periodic modulation due to the load distribution, possible bearing unbalance or misalignment, as well as the periodic changes in the impulse response as the fault moves towards and backwards the sensor, T is the inter-arrival time between two consecutive impacts, τ_i accounts for the uncertainties on the inter-arrival time (jitters) of the *i*th impact due to the necessary random slip of the rolling elements and $n(t)$ gathers the background noise.

Since this work focus the attention on the numerical implementation of Equation (1), instead of taking into account uncorrelated (white) jitters τ_i [11], an uncorrelated (white) inter-arrival time difference $\tau_{i+1} - \tau_i$ is used [8]:

$$E\{(\tau_{i+1} - \tau_i)(\tau_{j+1} - \tau_j)\} = \delta_{ij}\sigma_\tau^2 \tag{2}$$

where σ_τ is the standard deviation and δ_{ij} is the Kronecker symbol. Even if Equation (1) embodies a well defined harmonic structure, the presence of very slight random fluctuations of the inter-arrival time of consecutive impulses causes the rapidly turns of the signal into a random one. Therefore, weak harmonic component can be located in the lower-frequency range, and a dominating random cyclostationary component can be located in the higher-frequency range (pseudo-cyclostationary). A detailed theoretical explanation of the frequency content of Equation (1) can be found in [8, 11].

When a localized fault propagates on the surface where it was initiated, a larger area of the bearing becomes involved in the genesis of the vibration signature. In this scenario, no sharp impulses are generated, but the fault signature becomes purely cyclostationary (as opposed to pseudo-cyclostationary) [7, 12]. This pure cyclostationary content is the result of a randomly distributed phase, caused by the different position on the rough surface of the rolling elements for every revolution. However, strong periodic component are generated at the shaft periodicity, when the fault only extends over a limited sector of the race. Moreover, if the bearing is highly loaded, a periodic component can be initiated by the bearing stiffness variation due to the changing numbers and positions of the rolling elements in the load zone. The distributed fault vibration signature may be written [7]:

$$x_d(t) = p(t) + B(t) \tag{3}$$

where $p(t)$ accounts for the periodic component such as shaft and stiffness variation periodicities and $B(t)$ for the purely cyclostationary content with $E\{B(t)\} = 0$.

2.2 Numerical implementation

This work focuses the attention on the numerical implementation of the vibration signal models of Equations (1) and (3). In particular, these models are extended to cover generic speed profile of the bearing shaft. In order to include a speed variation, the vibration signal is firstly defined in the angle domain and then transformed back to the time domain according with the chosen speed profile.

Let $\theta(t)$ be the rotation angle of a bearing moving race (inner and/or outer). Without loss of generality, in the following the bearing outer race is considered fixed whilst the inner race is rotating. A generic speed profile in the angle domain can be constructed as:

$$f_r(\theta) = f_c + 2\pi f_d \int cos(f_m\theta)d\theta \tag{4}$$

where f_c is the carrier component of the rotation frequency, f_d is the frequency deviation and f_m is the frequency modulation. All the main terms (f_c, f_d and f_m) of Equation (4) can or cannot be angle dependent. Figure 1 depict and example of Equation (4) for a case of sinusoidally speed varying profile. Without loss of generality, hereafter is assumed that at time $t = 0$ the defect is located at the position $\theta = 0$ and is in contact with a rolling element.

Concerning localized fault in ball bearing, the angle between two consecutive impulses can be easily obtained from the "gearbox" model of the rolling element bearing (see Table 1 for the usual bearing fault frequencies), for a inner-race fault:

$$\Delta\theta_{imp} = \frac{2\pi}{\frac{n_r}{2}(1 + \frac{d}{D}cos\beta)} \tag{5}$$

Equation (5) can be used to obtain the angular position of a series of equispaced impulses, i.e. a purely deterministic signal. As stated before, in order to take into account the necessary random slip of the rolling

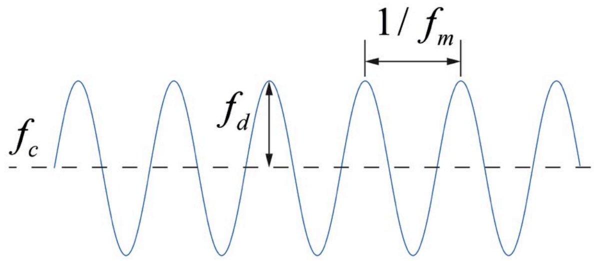

Figure 1: Example of sinusoidally speed varying profile.

elements a random contribution must be added to Equation (5). The angle between two consecutive impulses is strictly positive, and so the gamma law is the best candidate, nonetheless when the variance is low with respect to the mean value, the gamma distribution is well approximated by a normal distribution with the same mean and variance. In this work, the random contribution is taken into account by generating normally distributed random numbers with mean $\Delta\theta_{imp}$ and variance $\sigma^2_{\Delta\theta}$. As the speed profile is defined in terms of the rotation angle θ (Equation (4)), the inter-arrival time among the impulses can be obtained by the generated random numbers as:

$$\Delta T_i = \frac{\Delta\theta_i}{2\pi f_r} \tag{6}$$

where ΔT_i is the *i*th inter-arrival time, $\Delta\theta_i$ is the *i*th angle between two consecutive impulses randomly generated with mean $\Delta\theta_{imp}$ and variance $\sigma^2_{\Delta\theta}$.

The results of Equation (6) are the inter-arrival times of each impulse with the speed profile defined in Equation (4). These times define the beginning of each impulse response $h(t - iT - \tau_i)$ in the time signal itself, such a signal can be obtained in a Matlab/Octave environment as follows:

1. generate a L point vector filled with zeros, corresponding at times $t = n/f_s$, where f_s is the sample frequency in Hz and n is a index ranging from 0 to $L - 1$,
2. place 1 at index values obtained by dividing each inter-arrival time ΔT_i by the chosen sample frequency f_s,
3. weight the so generated vector with the weighting function $q(iT)$,
4. filter the weighted vector with the FFT-based method of overlap-add by choosing as filter coefficients the impulse response function of a SDOF system in terms of acceleration.

Several methods can be found in literature in order to obtain the impulse response of a SDOF system [6, 13], they deal with the implementation of such a response in the frequency domain and then transform it back in the time domain via the Inverse Fourier Transform. However, this procedure involves the generation of a low pass filter as well as a phase correction [6]. In this work Authors decided to generate the response of the SDOF system to the unit impulse in the time domain as:

$$x_{SDOF}(t) = \frac{F/m}{\omega_d} e^{-\zeta\omega_n t} sin(\omega_d t) \tag{7}$$

where F is the amplitude of the force exciting the SDOF system, m the system mass, ζ the damping coefficient, ω_n the natural frequency in [rad/s] and $\omega_d = \omega_n\sqrt{1-\zeta^2}$. The response in terms of acceleration can be simply obtained by a double derivative with respect to time. In this scenario, the numerical derivative does not add high frequency noise inside the signal, because no noise is present in the generated $x_{SDOF}(t)$.

From the procedure heretofore described, the key point is to find inside the time signal, the index n corresponding to the beginning of the impulse. De facto, n must be an integer number, however by dividing ΔT_i

by the selected sample frequency f_s a rational number is usually obtained. Instead of using an interpolation procedure on the time signal itself, Authors decide to rounding the rational numbers to the nearest integers. With this operation, an error is introduced that depends on the selected sample frequency f_s (the greater f_s the lower is the error), which affects both mean and variance of the theoretical ΔT_i. Let $\overline{\Delta T_i}$ the value of ΔT_i obtained via the rounding procedure, the error terms is:

$$\varepsilon = \overline{\Delta T_i} - \Delta T_i \tag{8}$$

with mean and variance:

$$\begin{gathered} E\{\varepsilon\} = E\{\overline{\Delta T_i}\} - E\{\Delta T_i\} \\ \sigma_\varepsilon^2 = \sigma_{\overline{\Delta T_i}}^2 - \sigma_{\Delta T_i}^2 - 2COV\{\overline{\Delta T_i}, \Delta T_i\} \end{gathered} \tag{9}$$

Finally, the last term of Equation (1) which deals with the noise component, can be added to the signal by generating randomly distributed number with a given power. The power of the noise can be set with a desired Signal-to-Noise Ratio (SNR), which is a measure that compares the level of a desired signal to the level of background noise. The SNR is defined as:

$$SNR = 10log_{10}\left(\frac{P_{signal}}{P_{noise}}\right) \tag{10}$$

where P_{signal} is the Power of the signal without noise and P_{noise} is the noise Power. Figure 2 depicts the

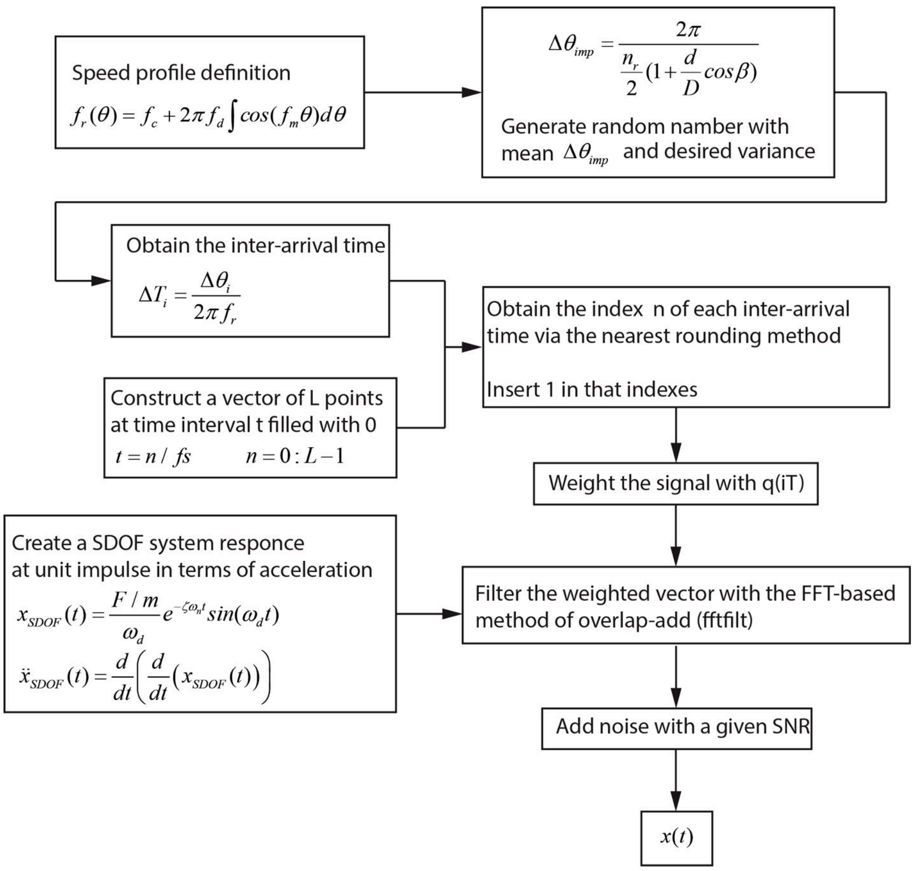

Figure 2: Schema for the numerical implementation of Equation (1).

schema of the proposed procedure. Moreover, in the Appendix A an Octave function called *bearingSignalModelLocal* has been inserted in order to easily implement Equation (1).

The same procedure can be efficiently extended for the case of distributed faults in rolling element bearing. This vibration signal model is a mixture of two terms, one deterministic and one purely cyclostationary. Once the speed profile has been defined with respect to the rotation angle θ, the deterministic part can be described in the angular domain as:

$$p_{rot}(\theta) = q_{rot}cos\left(\frac{fc}{fc}\theta + \frac{fd}{fc}\int cos\left(\frac{fm}{fc}\theta\right)d\theta\right) \tag{11}$$

$$p_{stiff}(\theta) = q_{stiff}cos\left(\frac{fc}{fc}\tau_{stiff}\theta + \frac{fd}{fc}\tau_{stiff}\int cos\left(\frac{fm}{fc}\tau_{stiff}\theta\right)d\theta\right)$$

where q_{rot} and q_{stiff} are two positive numbers which weigh the amplitude of the deterministic components, whilst τ_{stiff} is a geometrical bearing parameter which can be obtained by the "gearbox" bearing model as:

$$\tau_{stiff} = \frac{n_r}{2}\left(1 - \frac{d}{D}cos\beta\right) \tag{12}$$

De facto, in rolling element bearings, the frequency of the stiffness variation is equal to the frequency of an outer-race fault. As done before, by the knowledge of the speed profile, the angle signals can be transformed by a simple interpolation in the time domain.

The purely cyclostationary component ($B(t)$) is a random modulated noise, where the modulation frequency is the fault frequency. Once the speed profile is selected, the modulating function for an inner-race fault (see Table 1 for other types of fault), can be expressed in the angle domain as:

$$q(\theta) = 1 + q_{Fault}sin\left(\frac{f_c}{f_c}\tau_{Fault}\theta + \frac{f_d}{f_c n_r}\tau_{Fault}\int cos\left(\frac{f_m}{f_c n_r}\tau_{Fault}\theta\right)d\theta\right) \tag{13}$$

where q_{Fault} is a number governing the amplitude of the modulating function and τ_{Fault} is a geometrical parameter which can be obtained by the "gearbox" model of the rolling element bearing as:

$$\tau_{Fault} = \frac{n_r}{2}(1 + \frac{d}{D}cos\beta) \tag{14}$$

Equation (13) can be transformed in the time domain by a simple interpolation and the purely cyclostationary component can be obtained by modulating normally distributed random number with the time domain $q_{Fault}(t)$. Finally, stationary noise can be added to the signal with a given SNR via the use of Equation (10). Figure 3 depicts the schema of the proposed procedure. Moreover, in the Appendix A an Octave function called *bearingSignalModelDist* has been inserted in order to easily implement Equation (3).

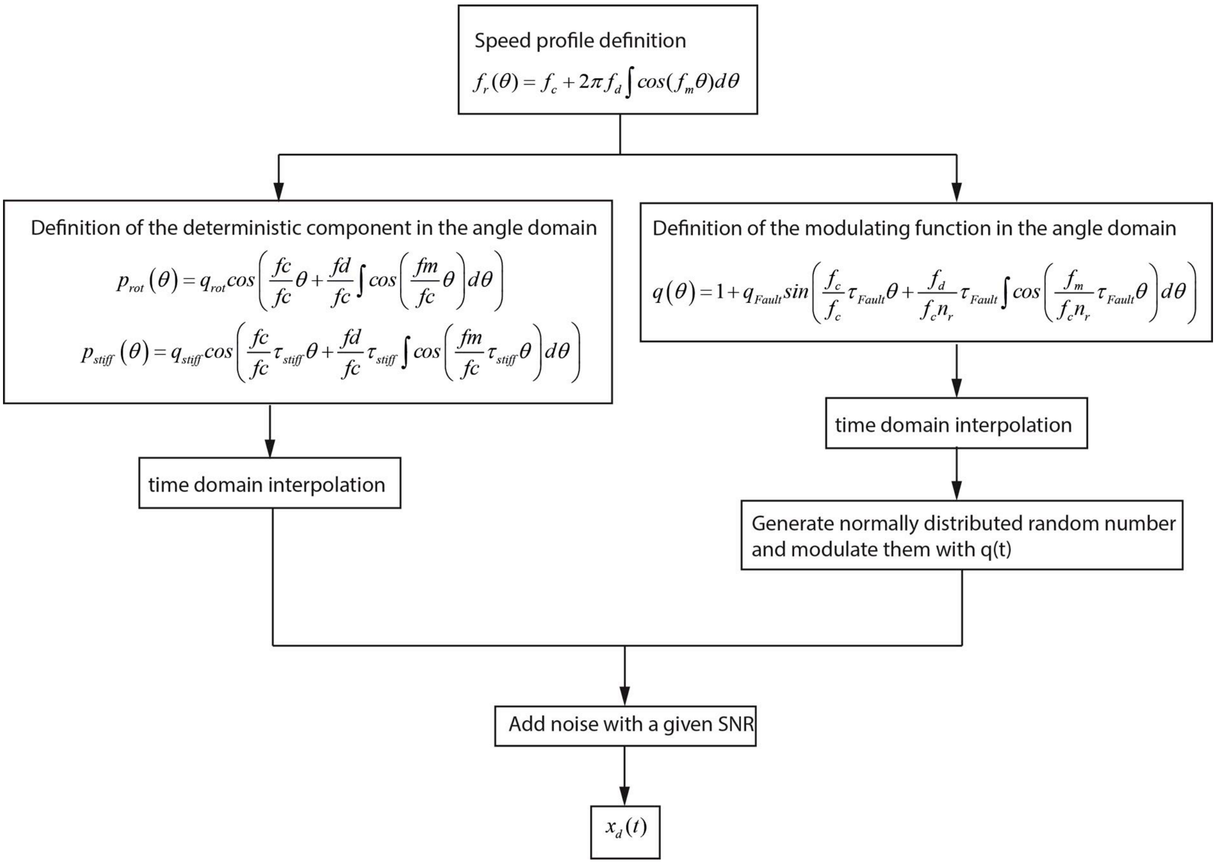

Figure 3: Schema for the numerical implementation of Equation (3).

3 Numerical example

Table 1 depicts the typical equation for the evaluation of bearing fault frequencies as well as the bearing dimensions used in the numerical examples, whilst Table 2 shows the vibration signal model parameters.

Typical fault frequencies [Hz]	
Inner-race fault	$\frac{n_r}{2}f_r(1 + \frac{d}{D}cos\beta)$
Outer-race fault	$\frac{n_r}{2}f_r(1 - \frac{d}{D}cos\beta)$
Rolling-element fault	$\frac{f_r d}{D}(1 - (\frac{d}{D}cos\beta)^2)$
Cage fault	$\frac{f_r}{2}(1 - \frac{d}{D}cos\beta)$

Geometrical Parameters	Value
Bearing roller diameter (d) [mm]	21.4
Pitch circle diameter (D) [mm]	203
Number of rolling elements (n_r)	23
Contact angle (β) [deg]	9.0

Table 1: Typical fault frequencies and bearing dimensions

As stated beforehand, a speed profile has to be generated. The selected speed profile used hereafter in the numerical examples is depicted in Figure 4, and it deals with a constant rotation frequency of 10Hz modulated at 1Hz with an amplitude of 0.8Hz (see Table 2). From now on both localized and distributed fault in the inner-race of a rolling element bearing are taken into account (see Appendix A for Octave scripts). The mean and variance of the random contribution related to the rolling element slips are set in the angle domain as $\Delta\theta_{imp}$ and $0.04\Delta\theta_{imp}$ respectively, which with the selected speed profile gives about $7.8981E-3$ (the inverse of the fault frequency) and $3.0224E-07$. As stated in the previous section, the impulse locations in the time domain signal are approximated by a neighbour interpolation that introduces an error term in both the selected mean and variance, which is related to the sample frequency of the time signal itself. In particular, the final mean and variance are $7.8980E-3$ and $3.0245E-07$ showing that the error is negligible in the generation of the vibration signal for the usual sample frequencies.

Vibration Signal Model Parameters	Localized fault Ex.	Distributed fault Ex.
Number of shaft revolutions	$1E4$	$1E4$
Number of points per revolution	2048	2048
Sample frequency f_s [Hz]	$20E3$	$20E3$
Carrier component of the shaft speed f_c [Hz]	10	10
Frequency deviation f_d [Hz]	$0.08f_c$	$0.08f_c$
Modulation frequency f_m [Hz]	$0.1f_c$	$0.1f_c$
SDOF spring stiffness k [N/m]	$2E13$	/
SDOF damping coefficient ζ	5%	/
SDOF natural frequency f_n [Hz]	$6E3$	/
Amplitude modulation for localized fault	0.3	/
Amplitude value of the deterministic component related to the stiffness variation q_{stiff}	/	0.1
Amplitude value of the deterministic component related to the bearing rotation q_{rot}	/	0.1
Amplitude value of the amplitude modulation at the fault frequency q_{Fault}	/	1
Signal to Noise Ratio [dB]	0	0
Expected fault frequencies		
Inner-race fault frequency [Hz]	≈ 126.97	≈ 126.97
Inner-race fault order [O]	12.69	12.69

Table 2: Vibration signal model data for localized and distributed faults in rolling elements bearing.

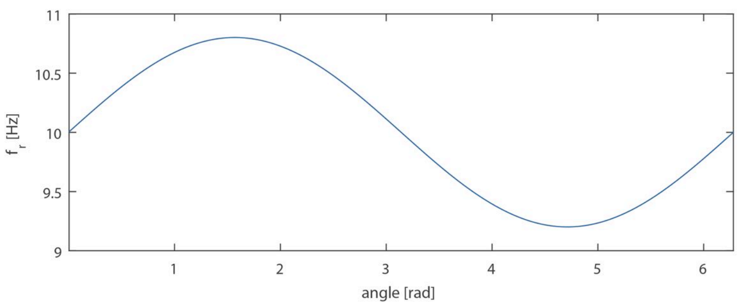

Figure 4: Speed profile used in the numerical examples for a complete revolution of the inner race.

Figure 5 depicts the simulated time signal in case of inner-race localized fault following the data of Table 2. At a first glance, the signal seems strictly deterministic, showing a series of impulse responses (Figure 5(a,b)). However, the random slips of the rolling elements turn the signal to strictly random. This effect can be easily seen from the signal PSD. Figure 5(c) plots the PSD computed with the Welch's method, by using an Hanning window with a 75% of overlap. It is clearly visible from the signal PSD that the noise signal is purely random in nature, in particular the harmonic series related to the repetition of the impulses are strongly masked by the background noise and die quickly. De facto, Antoni and Randall [8, 11] proved that the decay of the harmonic structure strictly depends on the selected variance due to the low-pass filter nature of Equation 1. In order to highlight the fault frequency cyclostationary analysis has to be carried out. The main signal processing technique in the cyclostationary field is Spectral Correlation Density function (SCD) which depicts the cyclostationary content with respect to the frequency content of the signal. This technique has to be used in case of constant speed, however when the speed is changing a cyclo-non-stationary signal is generated. G. D'Elia et al. [14] were the firsts to explore the order-frequency approach extending the SCD to speed varying signals. D. Abboud et al. [15] proposed a more rigorous approach to the analysis of cyclo-non-stationary signals. Figure 7(a) depicts the Order-Frequency Spectral Correlation function (OFSC) for the synthesized signal in case of localized fault. It is possible to see how the order related to the inner-race fault (see Table 2) is highlighted around a frequency region of 6kHz, which is the resonance frequency excited by the bearing impulses. Moreover, the OFSC also highlights the amplitude modulation due to the periodic variation of the load distribution.

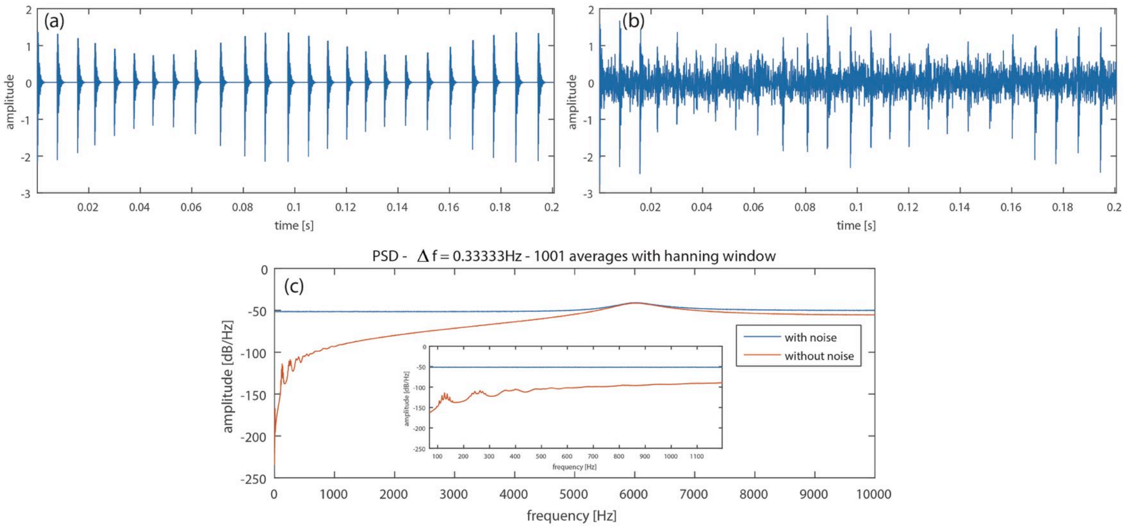

Figure 5: Simulated vibration signal in case of inner-race localized fault: (a) without noise, (b) with noise, (c) Power Spectral Density (PSD).

Figures 6(a,b) depict the time signal for a inner-race distributed fault with and without noise addiction. It is possible to see how the signal seems strictly random. In particular, even without noise the deterministic component related to the stiffness variation as well as shaft rotation are hidden. Figure 6(c) highlights the PSD of such a signal, where the random contribution is clearly visible in the medium/high frequency range, whilst the deterministic component are depicted in the low frequency region. Moreover, due to the speed variation, modulation around the bearing stiffness variation frequency can be easily detected. As done before, in order to highlight the fault frequency the OFSC function is evaluated on the simulated signal. Figure 7(b) plots the result of these operation. The cyclic order frequency concerning the inner-race fault (12.65O) is clearly visible in all the frequency range, focusing the broad band phenomenon involved in the distributed fault signature.

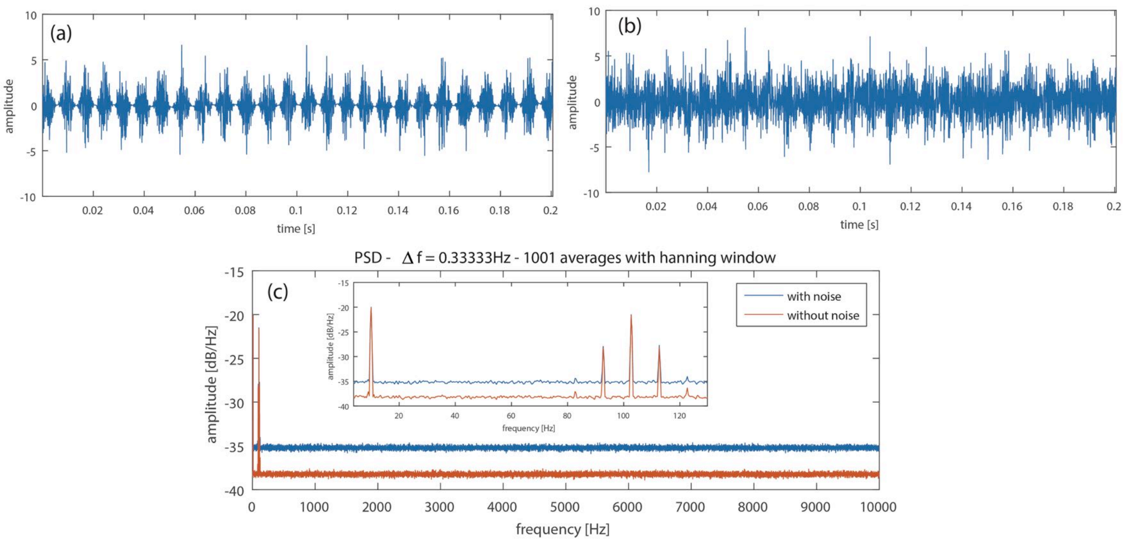

Figure 6: Simulated vibration signal in case of inner-race distributed fault: (a) without noise, (b) with noise, (c) Power Spectral Density (PSD).

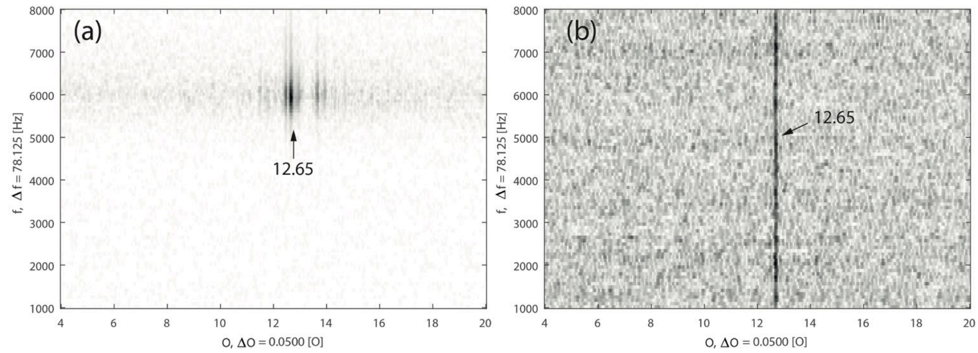

Figure 7: Order-Frequency Spectral Correlation Function (OFSC): (a) localized fault, (b) distributed fault.

4 Conclusion

This paper details an algorithm to simulate the expected vibration signal of a faulted bearing. The model is based on the work of Antoni [11], with few improvements. The basic features that the user could set are listed below:

- selection of the location of the fault (e.g. outer ring, inner ring, etc...),
- selection of the stage of the fault (e.g. punctual fault, distributed fault, etc...),
- cyclostationarity of the signal,
- random contributions,
- deterministic contributions,
- effects of resonances in the machine,
- working conditions (stationary and non-stationary).

This project has been developed under a Creative Commons licence and the vision of the project is a set of tools accepted by the community of researchers on condition monitoring, for the preliminary validation of new diagnostics techniques. The reader could freely and immediately use the script in Appendix A to simulate different faults and different operating conditions.The script is provided for the open-source Octave environment. The paper fully details the theoretical background and the numeric implementation of the vibration model. Examples of the output signals for simulated faulty bearings (localized and generalized faults) have been shown and commented.

References

[1] I. El-Thalji, E. Jantunen, *A summary of fault modelling and predictive health monitoring of rolling element bearings*, Mechanical Systems and Signal Processing, Vol. 60 (2015), pp. 252-272.

[2] P.D. McFadden, J.D. Smith, *Vibration monitoring of rolling element bearings by the high frequency resonance technique a review*, Tribology International, Vol. 117 (1984), pp. 3-10.

[3] P.D. McFadden, J.D. Smith, *Model for the vibration produced by a single point defect*, Journal of Sound and Vibration, Vol. 96 (1984), pp. 69-82.

[4] P.D. McFadden, J.D. Smith, *The vibration produced by multiple point defects in a rolling element bearing*, Journal of Sound and Vibration, Vol. 98 (1984), pp. 263-273.

[5] Y.T. Su, S.J. Lin, *On initial detection of a tapered roller bearing frequency domain analysis*, Journal of Sound and Vibration, Vol. 155 (1992), pp. 75-84.

[6] D. Ho, R.B. Randall *Optimisation of bearing diagnostic techniques using simulated and actual bearing fault signals*, Mechanical Systems and Signal Processing, Vol. 14 (2000), pp. 763-788.

[7] J. Antoni, R.B. Randall *Differential Diagnosis of Gear and Bearing Faults*, Journal of Vibration and Acoustics, Vol. 124 (2002), pp. 165-171.

[8] J. Antoni, R.B. Randall, *A Stochastic Model for Simulation and Diagnostics of Rolling Element Bearings With Localized Faults*, Journal of Vibration and Acoustics, Vol. 125 (2003), pp 282-289.

[9] W.A. Gardner, *Introduction to random processes with application to signals and systems*, Macmillan, New York.

[10] A. Bourdon, H. André, D. Rémond, *Introducing angularly periodic disturbances in dynamic models of rotating systems under non-stationary conditions*, Mechanical Systems and Signal Processing, Vol. 44 (2014), pp. 60-71

[11] J. Antoni, *Cyclic spectral analysis of rolling-element bearing signals: Facts and fictions*, Journal of Sound and Vibration, Vol. 304 (2007), pp. 497-529.

[12] G. D'Elia, S. Delvecchio, M. Coccconcelli, E. Mucchi, G. Dalpiaz, *Application of cyclostationary indicators for the diagnostics of distributed faults in ball bearings*, *Proceedings of the ASME 2013 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE 2013*. August 4-7, 2013, Portland.

[13] J.B. Roberts, *On the reponse of a simple oscillator to random impulses*, Journal of Sound and Vibration, Vol. 4 (1966), pp. 51-61.

[14] G. D'Elia, Z. Daher, J. Antoni, *A nover apporach for the cyclo-non-stationary analysis of speed varying signals*, *Proceedings of ISMA 2010*. September 22-27, 2010, Leuven Belgium.

[15] D. Abbod, S. Baudin, J. Antoni, D. Rémond, M. Eltabach, O. Sauvage, *The spectral analysis of cyclo-non-stationary signals*, Mechanical Systems and Signal Processing, Vol. 75 (2016), pp. 280-300.

Appendix A: Octave code

```
%% Simulated localized and distributed fault in rolling
% element bearing
%
% G. D'Elia and M. Cocconcelli

clear
clc

%% Bearing geometry
d = 21.4; % bearing roller diameter [mm]
D = 203; % pitch circle diameter [mm]
```

```
n = 23; % number of rolling elements
contactAngle = 9*pi/180; % contact angle
faultType = 'inner';

%% Speed profile
N = 2048; % number of points per revolution
Ltheta = 10000*N; % signal length
theta = (0:Ltheta-1)*2*pi/N;
fc = 10;
fd = 0.08*fc;
fm = 0.1*fc;
fr = fc + 2*pi*fd.*(cumsum(cos(fm.*theta)/N));

%% Localized fault
varianceFactor = 0.04;
fs = 20000; % sample frequency [Hz]
k = 2e13;
zita = 5/100;
fn = 6e3; % natural frequency [Hz]
Lsdof = 2^8;
SNR_dB = 0;
qAmpMod = 0.3;
[tLocal,xLocal,xNoiseLocal,frTimeLocal,meanDeltaTLocal,varDeltaTLocal,meanDeltaTimpOverLocal,varDeltaTimpOverLocal,errorDeltaTimpLocal] = bearingSignalModelLocal(d,D,contactAngle,n,faultType,fr,fc,fd,fm,N,varianceFactor,fs,k,zita,fn,Lsdof,SNR_dB,qAmpMod);

%% Distributed fault
fs = 20000; % sample frequency [Hz]
SNR_dB = 0;
qFault = 1;
qStiffness = 0.1;
qRotation = 0.1;
[tDist,xDist,xNoiseDist,frTimeDist] = bearingSignalModelDist(d,D,contactAngle,n,faultType,fc,fd,fm,fr,N,fs,SNR_dB,qFault,qStiffness,qRotation);
```

```
function [t,x,xNoise,frTime,meanDeltaT,varDeltaT,meanDeltaTimpOver,varDeltaTimpOver,errorDeltaTimp] = bearingSignalModelLocal(d,D,contactAngle,n,faultType,fr,fc,fd,fm,N,varianceFactor,fs,k,zita,fn,Lsdof,SNR_dB,qAmpMod)
    %% Generation of a simulated signal for localized fault in rolling element bearing
    %
    % Input:
    % d = bearing roller diameter [mm]
    % D = pitch circle diameter [mm]
    % contactAngle = contact angle [rad]
    % n = number of rolling elements
    % faultType = fault type selection: inner, outer, ball [string]
    % fr = row vector containing the rotation frequency profile
    % fc = row vector containing the carrier component of the speed
    % fm = row vector containing the modulation frequency
    % fd = row vector containing the frequency deviation
    % N = number of points per revolution
    % varianceFactor = variance for the generation of the random contribution (ex. 0.04)
    % fs = sample frequency of the time vector
    % k = SDOF spring stiffness [N/m]
    % zita = SDOF damping coefficient
    % fn = SDOF natural frequency [Hz]
    % Lsdof = length of the in number of points of the SDOF response
    % SNR_dB = signal to noise ratio [dB]
    % qAmpMod = amplitude modulation due to the load (ex. 0.3)
    %
    % Output:
    % t = time signal [s]
    % x = simulated bearing signal without noise
    % xNoise = simulated bearing signal with noise
    % frTime = speed profile in the time domain [Hz]
    % meanDeltaT = theoretical mean of the inter-arrival times
    % varDeltaT = theoretical variance of the inter-arrival times
    % menDeltaTimpOver = real mean of the inter-arrival times
    % varDeltaTimpOver = real variance of the inter-arrival times
    % errorDeltaTimp = generated error in the inter-arrival times
    %
    % G. D'Elia and M. Cocconcelli

```

```
if nargin < 14,
    qAmpMod = 1;
end

switch faultType
    case 'inner'
        geometryParameter = 1 / 2 * (1 + d/D*cos(contactAngle)); % inner race fault
    case 'outer'
        geometryParameter = 1 / 2 * (1 - d/D*cos(contactAngle)); % outer race fault
    case 'ball'
        geometryParameter = 1 / (2*n) * (1 - (d/D*cos(contactAngle))^2)/(d/D); % outer race fault
end

Ltheta = length(fr);
theta = (0:Ltheta-1)*2*pi/N;

deltaThetaFault = 2*pi/(n*geometryParameter);
numberOfImpulses = floor(theta(end)/deltaThetaFault);
meanDeltaTheta = deltaThetaFault;
varDeltaTheta = (varianceFactor*meanDeltaTheta)^2;
deltaThetaFault = sqrt(varDeltaTheta)*randn([1 numberOfImpulses-1]) + meanDeltaTheta;
thetaFault = [0 cumsum(deltaThetaFault)];
frThetaFault = interp1(theta,fr,thetaFault,'spline');
deltaTimp = deltaThetaFault ./ (2*pi*frThetaFault(2:end));
tTimp = [0 cumsum(deltaTimp)];

L = floor(tTimp(end)*fs); % signal length
t = (0:L-1)/fs;
frTime = interp1(tTimp,frThetaFault,t,'spline');

deltaTimpIndex = round(deltaTimp*fs);
errorDeltaTimp = deltaTimpIndex/fs - deltaTimp;

indexImpulses = [1 cumsum(deltaTimpIndex)];
index = length(indexImpulses);
while indexImpulses(index)/fs > t(end)
        index = index - 1;
end
indexImpulses = indexImpulses(1:index);

meanDeltaT = mean(deltaTimp);
varDeltaT = var(deltaTimp);
meanDeltaTimpOver = mean(deltaTimpIndex/fs);
varDeltaTimpOver = var(deltaTimpIndex/fs);

x = zeros(1,L);
x(indexImpulses) = 1;

% amplitude modulation
if strcmp(faultType,'inner')

    if length(fc) > 1,
        thetaTime = zeros(1,length(fr));
        for index = 2:length(fr),
               thetaTime(index) = thetaTime(index - 1) + (2*pi/N)/(2*pi*fr(index));
        end
        fcTime = interp1(thetaTime,fc,t,'spline');
        fdTime = interp1(thetaTime,fd,t,'spline');
        fmTime = interp1(thetaTime,fm,t,'spline');

        q = 1 + qAmpMod * cos(2*pi*fcTime.*t + 2*pi*fdTime.*(cumsum(cos(2*pi*fmTime.*t)/fs)));
    else
        q = 1 + qAmpMod * cos(2*pi*fc*t + 2*pi*fd*(cumsum(cos(2*pi*fm*t)/fs)));
    end
    x = q .* x;
end

[sdofRespTime] = sdofResponse(fs,k,zita,fn,Lsdof);
x = fftfilt(sdofRespTime,x);

L = length(x);
rng('default'); %set the random generator seed to default (for comparison only)
```

```
    SNR = 10^(SNR_dB/10); %SNR to linear scale
    Esym=sum(abs(x).^2)/(L); %Calculate actual symbol energy
    N0 = Esym/SNR; %Find the noise spectral density
    noiseSigma = sqrt(N0); %Standard deviation for AWGN Noise when x is real
    nt = noiseSigma*randn(1,L);%computed noise
    xNoise = x + nt; %received signal
```

```
function [t,x,xNoise,frTime] = bearingSignalModelDist(d,D,contactAngle,n,faultType,fc,fd,fm,fr,N,fs,SNR_dB,qFault,qStiffness,qRotation)
        %% Generation of a simulated signal for distributed fault in rolling element bearing
    %
    % Input:
    % d = bearing roller diameter [mm]
    % D = pitch circle diameter [mm]
    % contactAngle = contact angle [rad]
    % n = number of rolling elements
    % faultType = fault type selection: inner, outer, ball [string]
    % fr = row vector containing the rotation frequency profile
    % fc = row vector containing the carrier component of the speed
    % fm = row vector containing the modulation frequency
    % fd = row vector containing the frequency deviation
    % N = number of points per revolution
    % SNR_dB = signal to noise ratio [dB]
    % qFault = amplitude modulation at the fault frequency
    % qStiffness = amplitude value of the deterministic component related to the stiffness variation
    % qRotation = amplitude value of the deterministic component related to the bearing rotation
    %
    % Output:
    % t = time signal [s]
    % x = simulated bearing signal without noise
    % xNoise = simulated bearing signal with noise
    % frTime = speed profile in the time domain [Hz]
    %
    % G. D'Elia and M. Cocconcelli

    switch faultType
        case 'inner'
            geometryParameter = 1 / 2 * (1 + d/D*cos(contactAngle)); % inner race fault
        case 'outer'
            geometryParameter = 1 / 2 * (1 - d/D*cos(contactAngle)); % outer race fault
        case 'ball'
            geometryParameter = 1 / (2*n) * (1 - (d/D*cos(contactAngle))^2)/(d/D); % outer race fault
    end

    Ltheta = length(fr);
    theta = (0:Ltheta-1)*2*pi/N;
    thetaTime = zeros(1,length(fr));
    for index = 2:length(fr),
        thetaTime(index) = thetaTime(index - 1) + (2*pi/N)/(2*pi*fr(index));
    end

    L = floor(thetaTime(end)*fs); % signal length
    t = (0:L-1)/fs;
    frTime = interp1(thetaTime,fr,t,'spline');

    % generating rotation frequency component
    xRotation = qRotation * cos(fc/fc.*theta + fd./fc.*(cumsum(cos(fm./fc.*theta)/N)));
    xRotationTime = interp1(thetaTime,xRotation,t,'spline');

    % generating stiffness variation
    tauStiffness = n / 2 * (1 - d/D*cos(contactAngle));
    xStiffness = qStiffness * cos(fc./fc*tauStiffness.*theta + fd./fc*tauStiffness.*(cumsum(cos(fm./fc*tauStiffness.*theta)/N)));
    xStiffnessTime = interp1(thetaTime,xStiffness,t,'spline');

    % amplitude modulation
    tauFautl = n*geometryParameter;
    q = 1 + qFault * sin(fc./fc*tauFautl.*theta + fd./fc*geometryParameter.*(cumsum(cos(fm./fc*geometryParameter.*theta)/N)));
    qTime = interp1(thetaTime,q,t,'spline');
    xFaultTime = randn(1,L);
```

```
    xFaultTime = xFaultTime .* qTime;

    % adding therms
    x = xFaultTime + xStiffnessTime + xRotationTime;

    % Adding noise with given SNR
    rng('default'); %set the random generator seed to default (for comparison only)
    SNR = 10^(SNR_dB/10); %SNR to linear scale
    Esym=sum(abs(x).^2)/(L); %Calculate actual symbol energy
    N0 = Esym/SNR; %Find the noise spectral density
    noiseSigma = sqrt(N0); %Standard deviation for AWGN Noise when x is real
    nt = noiseSigma*randn(1,L);%computed noise
    xNoise = x + nt; %received signal
```

```
function [sdofRespTime] = sdofResponse(fs,k,zita,fn,Lsdof)
    %% Acceleration of a SDOF system
    % [sdofRespTime] = sdofResponse(fs,k,zita,fn,Lsdof)
    %
    % Input:
    %   fs = sample frequency [Hz]
    %   k = spring stiffness [N/m]
    %   zita = damping coefficient
    %   fn = Natural frequency [Hz]
    %   Lsdof = desired signal length [points]
    %
    % Output:
    %   sdofRespTime = acceleration (row vector)
    %
    % G. D'Elia and M. Cocconcelli

    m = k/(2*pi*fn)^2;
    F = 1;
    A = F/m;
    omegan = 2*pi*fn;
    omegad = omegan*sqrt(1-zita^2);

    t = (0:Lsdof-1)/fs;
    % system responce
    xt = A/omegad * exp(-zita*omegan*t).*sin(omegad*t);     % displacement
    xd = [0 diff(xt)*fs];    % velocity
    sdofRespTime = [0 diff(xd)*fs];      % acceleration
```