VIBRATION SIGNAL PROCESSING USING MATLAB®

Ian Howard
Department of Mechanical Engineering
Curtin University of Technology
Perth, Western Australia

ABSTRACT: Vibration signal processing has traditionally been accomplished using analog and digital signal analysers, or by writing assembler, Fortran or C compiled code. The advent of higher level interpretive based signal processing software products like MATLAB has added a new dimension to vibration signal processing. This paper will outline the application of MATLAB to analysis of the vibration from rotating machinery such as rotors, gears and bearings. Sophisticated signal processing techniques can be developed within a very short period of time given the flexibility and interactive nature of MATLAB and the range of in-built functions such as Fourier transforms, cubic interpolations and digital filters. The three dimensional colour graphics included in this package provides a sophisticated visualisation capability for the more advanced signal processing techniques.

1. INTRODUCTION

Vibration is the study of the oscillatory motion of objects relative to a reference frame. Vibration is evident everywhere, and in many cases greatly affects engineering design and the performance of engineering devices. In some cases vibration can be harmful and should be avoided, in other cases it can be the crucial element in the success of a particular engineering process. In Mechanical Engineering, knowledge about vibration is very important and the measurement, analysis, modelling and prediction of vibration have provided engineers with important tools with which to understand engineering devices.

The measurement and analysis of engineering machinery vibration can be used to determine the performance of and the condition of the machine. Machinery vibration is often the important parameter which can be unobtrusively measured while the machine is operating, and from which knowledge about the condition or performance of the machine can be inferred using vibration signal processing. Vibration analysis of measured machinery vibration has been used for a long time and quite sophisticated techniques were developed over twenty years ago, [1].

Two basic approaches have been used for analysis of machinery vibration over the last twenty years. Firstly, there has been the development of dedicated hardware, (digital signal analysers), by a number of companies. These systems, many of which are extremely sophisticated, allow the engineer to connect the vibration measurement sensor directly to the hardware, where the signal is then digitised and may be analysed by a number of sophisticated techniques. The range of techniques now available on these systems is quite large and includes spectral analysis, narrow band zoom, time synchronous signal averaging, computer order tracking, high frequency envelope analysis and time-frequency analysis amongst others. These techniques form the bulk of the comprehensive techniques which are necessary for routine preventative maintenance of the majority of industrial type machines.

The second approach for vibration analysis was used widely prior to the advent of the dedicated vibration signal analysers, and involved writing software in Fortran, Pascal or C. This software was then used to analyse the vibration data which had previously been digitised by other hardware. This approach has become more important recently because of the need to analyse and interpret the measured vibration of complex machinery for condition monitoring. For complex machinery like helicopter gearboxes [2], and the Space Shuttle Main Engine Oxygen Turbopump [3], the dedicated hardware of the digital signal analysers were found to not have the necessary techniques or flexibility to correctly detect and diagnose the mechanical condition of the rotating machinery.

The writing of sophisticated vibration signal processing software from scratch using compilers is no easy task. In the past very few establishments have had the resources, knowledge or skills to dedicate manpower to the development of the software. The software tended to be platform specific and the adaptation to multiple computer types and the continuous computer operating system upgrades meant a considerable ongoing expense. Over the years new languages have arisen, sometimes rendering quite sizeable code obsolete.

More recent software developments have begun to have considerable impact upon the way that software is written for the more complex vibration signal processing tasks. In particular, a numeric computation and visualisation software product called MATLAB has been developed by The MathWorks Inc. This paper will attempt to address the MATLAB product, to describe its benefits in as far as the development of vibration signal processing software is involved, and provide examples relating to the vibration analysis of rotating machinery.

2. VIBRATION SIGNAL PROCESSING

An enormous wealth of information can be gained from the measurement and analysis of vibration data of rotating machinery in particular. The vibration signal contains the information of the oscillatory motion of the machine and its sub-components. Each of the sub-components of rotating machinery, ie: gears, bearings, shafts, couplings, engines, electric generators, pumps, fans, etc, can be modelled such that the vibration from various modes of failure can be predicted. For most of these sub-elements, such models can be quite sophisticated. A review of vibration failure models developed for rolling element bearings with localised damage [4, 5] found that they included the effects of load zone, variation of the transmission path between the defect location and the transducer and exponential decay of defect impulses due to internal damping. They also included single and multiple localised defects. Models have likewise been developed for the majority of mechanical components of rotating machinery covering the gross failure modes.

The goal of vibration signal processing is to analyse the vibration signal measured at particular locations on the machine, and extract enough information to determine the condition of each of the sub-elements of the machine. The techniques which are readily available for the bulk of rotating machines using digital signal analysers include spectral analysis, narrow band zoom, time synchronous signal averaging, computer order tracking, high frequency envelope analysis and time-frequency analysis. For those machines where the vibration environment is such that the existing techniques need further refinement or the mode of failure is complex, the standard techniques will need to be adapted or new approaches used to successfully detect impending failure. Whereas signal processing software can be written using Fortran, Pascal or C compilers, the author believes that the recent arrival of advanced signal processing tools such as MATLAB greatly changes the sphere of development of vibration software. The next section outlines the major reasons for using these latest tools instead of the traditional compilers.

3. MATLAB

MATLAB is a high performance numeric computation and visualisation software product developed by The Math Works Inc, Massachusetts, USA [6]. It involves the use of an interactive environment containing numeric computation, matrix computation, signal processing and graphics capabilities. The features which are particularly important in as far as vibration signal processing is concerned include its interactive nature, extensibility, multi platform capability, ability to link in existing code, standardised data format, range of built in functions and graphics capability.

Interactive Nature The interactive nature of MATLAB provides the vibration analyst with the powerful capability to develop and test what-if ideas on the run and with very little effort. This is particularly useful for vibration analysis of complex machinery as the analysis is most productive in a sequential operation with the next step often depending on what is observed in the current operation.

Extensibility MATLAB is open ended in the sense that a large number of toolboxes or additional built in signal processing capabilities can be added if needed, or written by the user if required. The user written code or additional toolboxes are ASCII text files (called M-files) which use the standard built in functions. The interactive nature often leads to processes which can be linked together to form a new function or M-file, allowing the analyst to quickly and easily develop his/her own suite of customised functions.

Multi Platform Capability MATLAB is available across a large number of popular computer and operating system platforms from the Cray to the PC. The same range of in-built functions, interactive nature, etc., is available on all platforms. This is particularly powerful for research groups having say a data acquisition capability on the PC and a more powerful computation platform for computationally extensive work. It also allows the ready exchange of techniques around the world due to the simple exchange of ASCII text files, regardless of the platform they were originally written on.

Linking With Existing Code An argument against changing software languages or operating systems is often the enormous amount of software that is already in place and working on the current system. MATLAB has the ability to link to software written in a number of languages, so that the original software is still useable, whilst the benefit of the new tools becomes available as well. One of the obvious benefits for vibration analysis is that an existing data acquisition capability can also be linked in to run with MATLAB as well. This should be particularly useful for the PC environment where it should now be possible to conduct data acquisition under Microsoft Windows straight into the MATLAB environment.

Standardised Data Format One of the headaches with vibration analysis is being given digital data to analyse where the format of the data is different to that which has previously been used. The binary data files used with MATLAB (MAT-files) have been standardised across computer platforms by having the original platform noted in the file header. When the file is read, the software takes account of the differing file formats automatically and does the correct read on the new platform. This functionality has been tested to the extent that MAT files have been sent around the other side of the world using internet and have been read correctly on differing platforms. This standardising feature of MATLAB may help to encourage research and development cooperation between differing organisations such as Universities and Industry.

Range of Built-In Functions The new versions of MATLAB, such as 4.0 released for the PC platform in August 1992, contain some 20 main categories of built-in functions ranging from the low-level file input/output functions, the polynomial and interpolation functions, three dimensional graphics functions and the data analysis and Fourier transform functions. It has been found by experience that the development of the majority of advanced signal processing techniques for vibration analysis can be accomplished using the built in functions and the additional signal processing toolbox functions.

Graphics Capability The three dimensional colour graphics capability of the recent versions of MATLAB have been a dramatic improvement over what could be readily accomplished using compiled code, particularly for the PC environment. Three dimensional colour surface and mesh plots with hidden line removal and axis labelling, from differing view points and with differing colour maps can now be achieved with the use of a small number of commands. The ability to animate the three dimensional plots and play back at predefined frame speeds has been particularly useful in the analysis of vibration from variable speed rotating machinery. The use of time - frequency analysis is very helpful in this regard and is one situation where the graphics display capabilities can be of significant benefit in understanding the nature of the vibration signal being analysed.

Other Comments Software environments like MATLAB change the typical nature of computing. For the vibration analyst, it opens up the possibility of conducting signal processing in a way that wasn't readily available before. MATLAB is fastest when the computation can be written using vectors or matrices. This means that the complete data array should be manipulated in memory at the same time. For typical machine condition monitoring techniques such as high frequency envelope analysis or time synchronous signal averaging, the implication is that the memory requirement is now much larger than that involved using smaller file based software routines. However, given enough memory (memory is cheap after all) the software now becomes easier to write, understand and use and hopefully faster than the previously file read write based routines. The amount of memory required to optimise processing speed for a particular application now becomes a significant factor, possibly even more significant than processor speed itself.

4. SIGNAL PROCESSING EXAMPLES

The following examples are given to indicate the type of vibration signal processing which can readily be accomplished using MATLAB. They relate to practical industrial type problems and illustrate both the sophistication and flexibility of the processing M-files which can be developed and the nature of the information which can be obtained from vibration data.

Rotor Vibration Displacement proximity probes are typically used to measure the relative motion and position of shafts with respect to the journal bearing housing of large steam turbine generator units. Normally two probes are set up 90° apart (usually horizontally or vertically) to measure displacement of the shaft at each bearing position. A tachometer signal is also obtained from the turbine shaft and provides a pulse once per revolution to determine the phase of the motion with respect to shaft rotation. Figure 1 shows a tachometer signal over eleven shaft revolutions and the corresponding x and y shaft displacements, digitally resampled to give 512 points per shaft revolution. This data was taken from a large turbine generator system with a rotational frequency of nominally 50 Hz.

Having digitally resampled the two shaft displacement signals over 11 shaft revolutions, an average orbit plot of the shaft motion can be constructed as a polar diagram. The resulting shaft orbit is shown in Figure 2a. The shaft motion in the two orthogonal axes can be considered as motion in the real and imaginary complex plane, which with the use of the complex Fast Fourier Transform (FFT) within MATLAB, provides the vibration analyst with knowledge about the resulting positive and negative rotating frequency vectors. The rotating frequency vectors are given in terms of shaft orders, and indicate the forwards and backwards whirl of the shaft motion, [7].

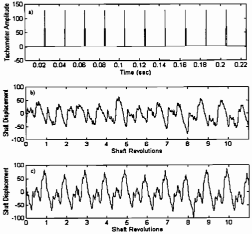

Figure 1. Rotor vibration of a steam turbine generator unit operating at 50 Hz. a) Tachometer signal over 11 shaft revolutions. b) Shaft displacement in one orthogonal axis, (relative units), c) Shaft displacement in the other orthogonal axis, (relative units).

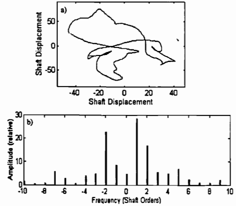

Figure 2. a) The average orbit plot over eleven shaft revolutions, (relative units), b) The forwards and backwards relative shaft whirl components obtained from the complex FFT of the orbit motion.

The trending of the complex Fourier transform frequency vectors has been found to be useful for monitoring the condition of rotor shafts for the presence of cracks [7], and it is likely that it could be used to determine the condition of the turbine generator journal bearings as well. The actual MATLAB commands to transform the orbit motion, as shown in Figure 2a, into the positive and negative frequency components, shown in Figure 2b, may be written using the complex FFT as given below,

*f = abs (fftshift (fft (x+i*y , length(h)))) / length(h);* (1)

where the in-built *fft* command is used to transform the real x and imaginary y shaft motion components into the frequency domain. The frequency components are then shifted using *fftshift* so that the spectrum runs from the negative frequency components through DC to the positive frequency components, and finally, the magnitude of the spectrum is obtained using the *abs* function. As shown in Figure 2b, the spectrum is drawn as a line spectrum since the orbit has been forced to be periodic over one shaft revolution.

Shaft Speed Estimation of a Diesel Engine A second example to highlight the flexibility of MATLAB involves the analysis of a tachometer signal obtained as a once per revolution pulse from the output shaft of a diesel engine gearbox coupled to a DC generator. Gearbox vibration analysis requires the precise angular position of all shafts in order to accomplish time synchronous signal averaging [8]. Normally, for approximate constant speed operation, a pulse once per revolution on one of the fastest shafts of the gearbox is sufficient to accurately determine angular positions for all shafts as a function of time.

A four stroke four cylinder diesel engine was coupled to a DC generator via a gearbox with a speed increase of approximately 1.6863. A pulse was obtained on the output side of the gearbox once per revolution. A portion of the tachometer signal over 0.4 seconds is shown in Figure 3a. By analysing when the tachometer pulse crosses a threshold level, the precise time of arrival of the pulse can be determined using cubic or spline interpolation. By repeatedly computing the time of arrival over a number of shaft revolutions, the actual speed of the gearbox shaft and hence the engine can be determined as a function of time, and more importantly, the precise angular position can be found at any instant in time. Figure 3b shows the speed of the gearbox shaft over a period of approximately three seconds obtained from the threshold crossing of the tachometer pulses.

The shaft frequency versus time illustrated in Figure 3b shows that the nominal shaft speed was approximately 42 Hz, and at times a deviation of over 1 Hz occurred in the actual shaft speed between single shaft revolutions. The large deviation in shaft speed for consecutive revolutions occurs because of the large torsional vibrations which exist in the diesel engine as the predominant forced excitation is due to the piston firing which occurs twice per revolution in a four stroke four cylinder engine. The large shaft speed difference between consecutive revolutions of the gearbox output shaft indicates that it is not possible to use a once per revolution pulse to infer variations in shaft speed that occur within each shaft revolution. Rather it indicates only the average shaft speed over that revolution. For a more precise indication of shaft speed and angular position, under torsional oscillatory environments, a higher frequency measure of shaft speed will be required such as is available using an optical encoder or a torsional laser velocity probe.

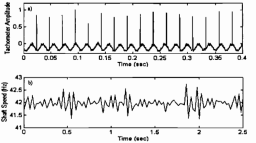

Figure 3. Shaft speed determination using a once per revolution tachometer pulse, a) Portion of a digitised tachometer signal, b) Resulting massaged gearbox shaft speed.

The MATLAB commands to detect threshold crossings are based upon the use of the *find* and *interp1* functions as shown below.

```
% create a new vector, time shifted by one data point
shiftdata = [data(1); data([1:length(data)-1])];

% find the time points which cross the threshold with
positive going slope.
z = find(data>threshold & shiftdata<threshold);

% interpolate on each zero crossing to find an accurate
estimate of actual
% crossing time

n = length(z);
for k=1:n
   % construct the short 3 pt x and y vectors for interpolation
   at each revolution
      y = [data(z(k)-1),data(z(k)),data(z(k)+1)];
      x = [(z(k)-1)*dt,z(k)*dt,(z(k)+1)*dt];
      s(k) = interp1(y,x,threshold,'spline');
end
```

The MATLAB vector operations as shown above make the resulting code very succinct and easy to understand once the nature of the in-built functions is understood.

The use of MATLAB is envisaged to be particularly useful for industrial applications of vibration analysis in a number of areas such as the monitoring of variable speed rotating machinery where the traditional black box approach is unlikely to provide the required flexibility to detect and diagnose faulty mechanical components.

High Resolution Frequency Analysis Frequency analysis of vibration data forms the major part of traditional vibration analysis. A number of techniques exist for estimating the power spectrum of a digitised signal apart from the use of the FFT, such as the maximum likelihood method, the auto regressive (AR) estimation technique and the moving average (MA) technique [9, 10].

The estimate of power spectrum using the FFT is based upon the power or magnitude obtained from the Fourier transform as

$$P(f) = E\,[\,X(f)\,X^*(f)\,], \qquad (2)$$

where $X(f)$ is the fast Fourier transform of the discrete time signal $x(t)$, * represents complex conjugation and $E[\ ]$ represents the expected value as obtained by spectral averaging. The smooth estimate of the power spectrum computed from stationary random time data is obtained by averaging over a number of windowed data records and is often referred to as the Welch method of power spectrum estimation.

The portion of an M-file to compute the power spectrum using the Welch method is shown below, where the time vector *data* is transformed into the power spectrum vector *ps*.

```
% loop over data vector using window of length lwin
for n = 1:nwin
        % extract data window
        dwin = data(istart:iend);
        % fft the windowed data segment
        freq = fft(win.*dwin);
        % compute the magnitude squared and sum
        ps = ps + (freq.*conj(freq)*dt/lwin)';
        istart = n*lwin + 1;
        iend = (n+1)*lwin;
end
% eliminate negative frequencies,
ps(np+1:lwin) = [];

% double positive frequencies;
ps(2:np) = 2*ps(2:np);
```

An example of the use of the power spectrum estimation on vibration data is shown in Figure 4, where torsional velocity vibration data was measured from the output shaft of a gearbox coupled between a diesel engine and a DC generator. The nominal speed of the engine, *fe*, was 24.9 Hz, the speed of the gearbox output shaft, *fo*, was 42 Hz and the intermediate gearbox shaft speed, *fi*, was approximately 16.66 Hz. A 100 second time record was digitised at a sample frequency of 2048 Hz, and a high resolution frequency spectrum was generated and averaged using twenty five 8192 point FFTs as shown in Figure 4.

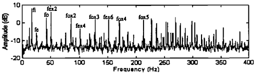

Figure 4. Torsional velocity spectrum computed from a diesel engine - gearbox - dynamometer test facility, with a resolution of 0.25 Hz.

The torsional spectrum up to 400 Hz shows a number of distinct frequency components, including harmonics of engine firing frequency (fex2), gearbox output shaft frequency (fo) and the fundamental component of the intermediate shaft frequency, fi. The flexible nature of MATLAB code allowed the high resolution spectrum to be computed across the complete frequency band with a number of averages. It would not normally be possible to obtain the number of averages and the frequency resolution together using the standard digital signal analysers currently on the market. However, using the MATLAB signal processing software, the actual frequency resolution and number of averages can be expanded as desired up to the full memory capacity of the computer platform.

5. DISCUSSION & CONCLUSION

This paper has outlined the use of MATLAB software for vibration signal processing. It has been argued that the interactive nature, extensibility and multi platform capability of MATLAB along with the ability to link in existing code, the use of a standardised data format, the range of built in functions and the graphics capability all make for a very productive software environment for the development of vibration signal processing software. Three vibration analysis examples based on data measured from rotating machinery have been outlined and portions of the actual MATLAB code have been shown to illustrate its succinct nature. It is envisaged that the industrial application of MATLAB should be particularly useful in solving the more complex vibration condition monitoring problems where the traditional black box approach does not provide the required flexibility.

ACKNOWLEDGMENTS

The help of staff and a number of final year students in Mechanical Engineering at Curtin University of Technology to collect the vibration data for this paper is gratefully acknowledged.

REFERENCES

1. Houser, D.R., and Drosjack, M.J., "Vibration signal analysis techniques". *US Army Air Mobility Research and Development Laboratory Technical Report-73-101*, December 1973.
2. Forrester, B.D., "Analysis of gear vibration in the time-frequency domain". *Proc. 44th meeting, Mechanical Failures Prevention Group*, Virginia Beach, Virginia, April 1990, 225-234.
3. Howard, I.M., "Signal processing techniques for mechanical diagnosis of Space Shuttle Main Engine turbomachinery using vibration analysis". *Technology Integration Incorporated Report R9301-002-RD.* Contract NAS8-38955, 1993.
4. Howard, I.M., "A review of factors affecting bearing vibration monitoring". *Proc. International Mechanical Engineering Congress and Exhibition,* pp 163-172, Perth, May 1994. The Institution of Engineers, Australia National Conference Publication No. 94/13.
5. Howard, I.M., "A review of rolling element bearing vibration: detection, diagnosis and prognosis". *DSTO Research Report 0013*, October 1994.
6. The Math Works Inc., *Reference and User's Guide for MATLAB*, 1992.
7. Gasch, R, "A survey of the behaviour of a simple rotating shaft with a transverse crack". *J. of Sound Vib.* **160**(3), 313 - 332(1993).
8. McFadden, P.D., "Interpolation techniques for the time domain averaging of vibration data with application to helicopter gearbox monitoring". Aeronautical Research Laboratory Technical Memorandum, ARL-AERO-PROP-TM-437, (1986).
9. Kay, S.M. and Marple, S.L., "Spectrum analysis - A modern perspective". *Proc. IEEE* **69**(11), 1380-1419 (1981).
10. Mechefske, C.K. and Mathew, J., "Parametric spectral trending and classification for monitoring low speed bearings". *Proc. Asia-Pacific Vibration Conference*, Monash University, Melbourne, Australia, November 1991, pp 5.42-5.48.