A module-based software system for spindle condition monitoring

Ruqiang Yan and Robert X. Gao*

Department of Mechanical Engineering,
University of Connecticut,
Storrs, CT 06269, USA
E-mail: rqyan@ieee.org
E-mail: rgao@engr.uconn.edu
*Corresponding author

Li Zhang

Global Research Center,
General Electric Corporation,
Niskayuna, NY 12309, USA
E-mail: zhangli@ge.com

Kang B. Lee

Manufacturing Metrology Division,
National Institute of Standards and Technology,
Gaithersburg, MD 20899, USA
E-mail: kang.lee@nist.gov

Abstract: Accurate identification of spindle working conditions is one of the key features of the next generation smart machining systems with built-in, self-diagnosis capability. This paper presents a module-based software system for online spindle defect identification and localisation through an analytic wavelet envelope spectrum algorithm. The software is designed in accordance with the architectural structure of OSA-CBM, and implemented using the graphical programming language LabVIEW. Spindle condition is displayed online in both a basic window for machine operators and a diagnosis window for advanced analysis. The software provides a user-friendly human-machine interface and contributes to realising a smart machine tool.

Keywords: modular software design; OSA/CBM architecture; spindle condition monitoring; analytic wavelet; SSI; stochastic subspace identification; smart machining system.

Reference to this paper should be made as follows: Yan, R., Gao, R.X., Zhang, L. and Lee, K.B. (2009) 'A module-based software system for spindle condition monitoring', *Int. J. Mechatronics and Manufacturing Systems*, Vol. 2, Nos. 5/6, pp.532–551.

Biographical notes: Ruqiang Yan received his PhD Degree in Mechanical Engineering from the University of Massachusetts Amherst (UMass) in 2007, and his MS Degree from the University of Science and Technology of China in 2002. He is currently a Senior Research Scientist with the Department of Mechanical Engineering, University of Connecticut, and worked as a Guest Researcher at the National Institute of Standards and Technology, Gaithersburg, MD from August 2006 to August 2008. His research interests include nonlinear time-series analysis, multi-domain signal processing, and energy-efficient sensing and sensor networks for the condition monitoring and health diagnosis of large-scale, complex, dynamical systems.

Robert X. Gao received his PhD and MS from the Technical University of Berlin, Germany, in 1991 and 1985. He was on the Faculty of the University of Massachusetts Amherst (UMass, 1995–2008), and is currently the Pratt & Whitney Endowed Chair Professor of Mechanical Engineering at the University of Connecticut. His research focuses on sensing, mechatronic design, and energy-efficient sensor networks. He received the NSF CAREER award (1996), the Barbara H. and Joseph I. Goldstein Outstanding Junior Engineering Faculty Award (1999) and the Outstanding Senior Faculty Award (2007) from UMass. He is a Fellow of the ASME and IEEE.

Li Zhang received the MS Degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2000 and the PhD Degree from the University of Massachusetts (UMass), Amherst, in 2004. He was a Postdoctoral Research Associate and Guest Researcher at the National Institute of Standards and Technology, Gaithersburg, MD, during August 2004–2006, and is currently a Research Scientist at the Global Research Centre of the General Electric Corporation. His research interests include electromechanical systems design, smart sensing systems, machine condition monitoring and health diagnosis, and sensor signal processing.

Kang B. Lee received the BS. and MS Degrees in Electrical Engineering from the Johns Hopkins University, Baltimore, and the University of Maryland, College Park, respectively. He is the Leader of the Sensor Development and Application Group, Manufacturing Engineering Laboratory, National Institute of Standards and Technology (NIST), Gaithersburg, MD. Since he joined NBS, now NIST, in 1973, he has worked in the fields of electronic instrumentation design, sensor based closed-loop machining, robotic manufacturing automation, smart sensor networking, and Internet-based distributed measurement and control systems. He is a fellow of the IEEE and was recognised by EE Times as one of 29 innovators in the world in 2006.

1 Introduction

Unexpected failure of machine tools can cause severe part damage and costly machine down time, affecting productivity as well as maintenance cost (Prickett and Johns, 1999; Rehorn et al., 2005). Since spindles are essential elements in virtually all machine tools and their working condition directly reflects upon machine tool performance, effective and reliable spindle condition monitoring is highly desired to capture potential failure at its early stage based on the sensor measurement data, to enhance the overall performance of the machine tool system. A spindle with such added capability would represent one of the key components in the next generation of smart machine tools with

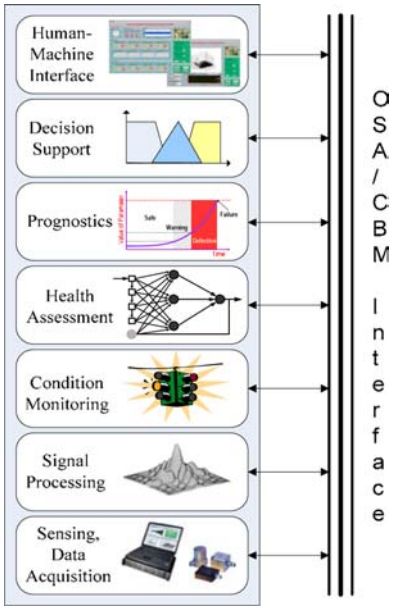
self-monitoring and diagnosis functionality (NIST, 2005). Over the past several years, research has been conducted to investigate fundamental issues related to the realisation of a self-diagnostic, sensor-integrated 'smart' spindle (Zhang et al., 2006a, 2006b). These include investigation of the correlation between spindle-integrated sensing (e.g., number of sensors and sensor placement strategy) and sensing quality (e.g., Signal-to-Noise Ratio (SNR) and effectiveness in feature coverage) on a machine spindle (Zhang et al., 2006a), and the development of advanced signal-processing techniques that combine features extracted from the time, scale and frequency domains to enhance spindle defect diagnosis and health assessment (Zhang et al., 2006b). In addition, an important aspect of the research is to devise a dynamic, data-driven software interface for communication with machine operators and decision-makers in online spindle condition monitoring and diagnosis (Zhang et al., 2007).

Efficient software design and implementation requires a modular and interchangeable architecture. A related effort is the Open System Architecture for Condition-Based Maintenance (OSA-CBM) programme that was set up by the Machinery Information Management Open Systems Alliance (MIMOSA) (Discenzo et al., 1998). The objective of the OSA-CBM programme is to develop an open architecture and standards for distributed CBM software components. Such an architecture has been defined in terms of functional layers (Figure 1), which include

- Sensing and Data Acquisition
- Signal Processing
- Condition Monitoring
- Health Assessment
- Prognostics
- Decision Support
- Human–Machine Interface.

Data communication among the layers is enabled by the OSA-CBM interface standards. These layers represent a logical flow of information from sensors in the physical layer to decision support in the system layer. The Human–Machine Interface layer can communicate with all other layers. For instance, a signal measured by the Sensing and Data Acquisition layer is used by the Signal-Processing layer to extract features on working conditions of the machine. Such information is in turn used by the Condition-Monitoring layer to compare against expected values and output condition indicators. The Health Assessment layer then utilises the input from the Condition-Monitoring layer to derive the current state of the system, which is subsequently used by the Prognostics layer to predict future performance of the system. The current state and predictions are fed into the Decision-Support layer to provide recommended actions for system maintenance. In addition, the current state and predictions, together with all measured and computed data, are displayed by the Human–Machine Interface layer such that the users can have visual interaction with the system.

Figure 1 Functional layers of the OSA-CBM architecture (see online version for colours)



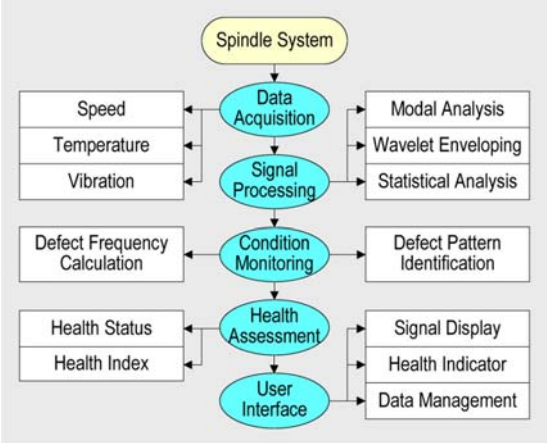
The layered architecture shown in Figure 1 facilitates the integration and interchangeability among sensors, electronics and software components (Thurston and Lebold, 2001; Lee et al., 2002), which allows flexibility for upgrading or expanding the system by incorporating new functions into corresponding layers. By taking advantage of the OSA-CBM architecture, this paper presents the design and implementation of a module-based software system for online spindle condition monitoring and diagnosis. After introducing the software configuration in Section 2, various functions of the software are discussed in Section 3, in which each of them is implemented as a module using the programming language LabVIEW. In Section 4, a Graphical User Interface (GUI) is designed to integrate all the modules into one entity for spindle condition monitoring and diagnosis. The performance of the integrated modules is then experimentally evaluated on a custom-designed spindle test system. Finally, conclusions are drawn in Section 5.

2 Software configuration

The designed software presents a unified platform for spindle signal measurement, processing, visualisation and management. It takes a modular approach to integrate various functions into one entity and allows for evaluation of the spindle working condition. Whereas the overall architectural design has taken into consideration the specific functionalities of the various modules concurrently to ensure consistency, each module is programmed independently. When a certain function needs to be modified, only the module related to that function will be reprogrammed. New functions can be added into the software package as independent modules. To be relevant to the OSA-CBM architecture, each OSA-CBM layer consists of one or more modules,

and each module can perform functions associated with one or more OSA-CBM layers. As illustrated in Figure 2, different types of sensor measurement data (e.g., speed, temperature and vibration) obtained in the Data Acquisition layer are transferred to the Signal-Processing layer such that features characterising the spindle dynamics are extracted. In addition to statistical analysis, several advanced signal-processing algorithms (e.g., Stochastic Subspace Identification (SSI)-based modal analysis and wavelet transform-based enveloping spectrum analysis) are embedded in this layer. As an example, the magnitudes of the power spectrum at the defect-related characteristic frequency lines extracted in the Signal Processing layer are fed into the Condition-Monitoring layer. The ratio of the magnitude of each characteristic frequency to the noise floor is then compared against a predefined threshold to identify potential defect patterns. In the Health Assessment layer, the output of the Condition-Monitoring layer is assessed, based on the trending information recorded in the system, to determine if the system health is degraded and specify the type and location of the identified degradation. The result is an enumerated condition indicator, which describes the operational state and health index of the spindle, and is visually displayed through the GUI.

Figure 2 Configuration and data flow of the designed monitoring software (see online version for colours)



The core functions of the software are designed as individual modules. Each module features a hierarchical structure in that it can call its second-level submodule, and each submodule can further call its next lower-level submodules to ultimately realise a specific function. In the following sections, details of the major modules (e.g., data acquisition module, WES module, SSI module, spindle health indicator module and data management module) are discussed in accordance with the structure of the OSA/CBM layers.

3 Software module design

The effectiveness and efficiency of a software design is dependent on how the programming process is executed. When compared with a text-based programming

language, such as C and C++, the programming language of LabVIEW is graphic-based, and uses graphic icons to replace the text command (NI LabVIEW User Manual, 2003). These graphic icons are wired together through drag-and-drop operations to realise various functions, such as data acquisition and data management. As a result, the software development process can be significantly simplified (Wang and Gao, 2000). Therefore, the graphical programming language of LabVIEW has been chosen for the design of the software system for spindle condition monitoring and diagnosis.

3.1 Data acquisition module

The data acquisition module has been designed to provide an interface to sensors that measure three types of signals: temperature, speed and vibration, and fulfil the functions associated with the Sensing and Data Acquisition layer of the OSA/CBM. Figure 3 illustrates major components of the data acquisition module in the spindle environment. Thermocouples were employed to measure temperature fluctuations of the spindle structure. A speed encoder was installed to measure the spindle rotating speed, which provides input to Condition-Monitoring layer for estimating the theoretical values of the structural-defect-related characteristic frequency. Four accelerometers were placed at the front and rear ends of the spindle, both within and outside the loading zones of the bearings, to measure their vibrations. Preliminary study has identified that spindle vibrations measured contain information on structural defects of the spindle system, and the corresponding frequency components are generally within the range of 50 kHz. In this study, a commercially available data acquisition board from National Instruments (NI PCI-6070E) with the highest sampling frequency of 1.25 M/s has been chosen. This board can provide eight analogue inputs in differential measurement mode. It also has the characteristics of 12-bit resolution, dedicated A/D converter and anti-aliasing filter per channel individually. Such configuration ensures that requirements for online spindle condition monitoring are met, for which all of the signals are to be measured simultaneously at over 100 kHz sampling rate without distortion.

Figure 3 Experimental set-up for spindle data acquisition (see online version for colours)

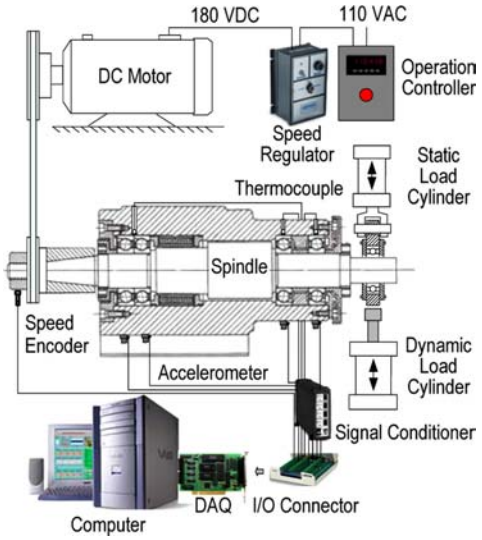


Figure 4 illustrated the designed data acquisition module in the LabVIEW programming environment. The channel information is provided from either the default settings saved in the configuration file (an example is shown in Figure 5) or the parameter set-up user interface, which will be discussed in the next section. After the channel information is formatted according to the specification of the DAQ board, data acquisition task is created and transferred to initialise the DAQ board. All the analogue output from the sensors are converted into a digital format, and then sampled and temporally stored in on-board buffer. According to the individual channel set-up, vibration, spindle speed and temperature signals are obtained. To provide software users with a clear display of information, a formula node that evaluates mathematical formulas and expressions has been designed to covert the voltage input for temperature readings in Celsius degree. The acquired signals are used as input to the Signal-Processing layer for feature extraction.

Figure 4 Code for performing the spindle data acquisition (see online version for colours)

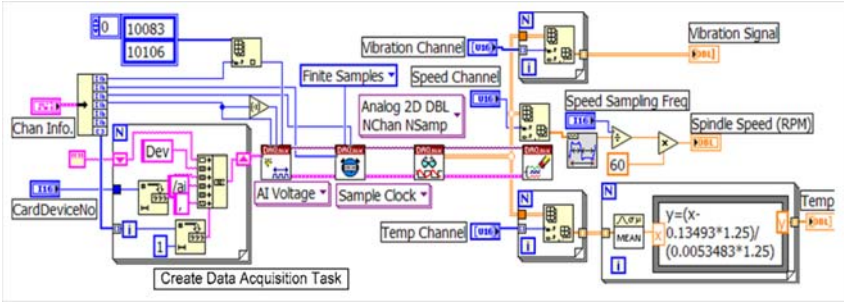


Figure 5 An example of the configuration file used for the data acquisition board (see online version for colours)

```
/* AdvancedSetting.INI */
VibrationChMode(0-SE,1-DIFF)      1
VibrationChSpeed(samples/second)          20000
VibrationChSampleNo         20000
VibrationChrange(V) 10
VibrationMeasureWaitTime(mS)      0
VibrationActive?(0-No,1-Yes)          1
VibrationChNo         0,1,2,3
TemperatureChMode(0-SE,1-DIFF)1
TemperatureChSpeed(samples/second)    10
TemperatureChSampleNo    10
TemperatureChrange(V)     10
TemperatureMeasureWaitTime(mS)          0
TemperatureActive?(0-No,1-Yes)    1
TemperatureChNo     5,6,7
SpeedChMode(0-SE,1-DIFF)          1
SpeedChSpeed(samples/second)    20000
SpeedChSampleNo   20000
SpeedChrange(V)     10
SpeedMeasureWaitTime(mS)          0
SpeedActive?(0-No,1-Yes)  1
SpeedChNo   4
OtherChMode(0-SE,1-DIFF) 0
OtherChSpeed(samples/second)    0
OtherChSampleNo    0
OtherChrange(V)     0
OtherMeasureWaitTime(mS)          0
OtherActive?(0-No,1-Yes)    0
OtherChNo
CardDeviceNo1
```

3.2 Wavelet Envelope Spectrum module

Once signals are obtained through the Data Acquisition Module embedded in the Sensing and Data Acquisition layer, they are converted to 'features' by the Signal-Processing layer for purpose of signal characterisation. Since bearings constitute a critical as well as vulnerable component in a machine tool spindle, the key module in Signal-Processing layer of the designed software is designed for bearing defect feature extraction. Previous research has shown that spectrum analysis of the envelope of a vibration signal is more effective in detecting and identifying bearing structural defect than that of the vibration signal itself (Ho and Randall, 2000; Yan and Gao, 2004). Traditionally, bandpass filtering was employed to pre-process vibration signals for subsequent envelope spectrum analysis (Jones, 1996; Tse et al., 2001). However, it suffers from a low SNR, especially when defect-induced vibration is weak in amplitude and overwhelmed by strong structural-borne noise (Yan and Gao, 2003). Various advanced techniques, such as blind source separation (Gelle et al., 2003), empirical mode decomposition (Yan and Gao, 2006) and wavelet transform (Wang and Gao, 2003), have been investigated for bearing signal processing, among which the wavelet transform has shown to provide an effective means for extracting a weak signal component out of a strong noise environment through a time–scale domain analysis (Wang and Gao, 2003). The wavelet transform of a signal at a given scale s can be essentially viewed as the signal passing through a bandpass filter (Yan, 2007). Therefore, combining the advantage of wavelet transform and envelope spectrum analysis would enhance the defect feature extraction (Zhang et al., 2006b). Furthermore, since the imaginary part of a complex wavelet is inherently the Hilbert transform of its real part, the wavelet coefficients of a transformed signal, in which the complex wavelet is used as the base wavelet, are analytic in nature. Thus, the corresponding modulus forms the signal's envelope. As a result, a complex wavelet-based signal transformation combines the ability of bandpass filtering with enveloping in one single computational step, eliminating the need for additional operations such as the Hilbert transform (Hahn, 1996) or low-pass filtering to extract signal envelope. Figure 6 illustrates the WES algorithm developed during the course of this study. Both the signal $x(t)$ and the scaled wavelet $\psi(s,t)$ are first processed using the Fourier transform. The inner product operation between the transformed coefficients $X(f)$ and $\sqrt{s}\Psi(sf)$ is then performed to obtain the wavelet coefficients of the signal in the frequency domain. Through an inverse Fourier transform, the wavelet coefficients are converted back into the time domain. The wavelet envelopes are then extracted by taking the absolute values (i.e., modulus) of the wavelet coefficients. Finally, the envelope spectrum is formed by performing the Fourier transform on the wavelet envelope.

To ensure effective feature extraction for defect identification of the spindle bearings, an appropriate complex wavelet function need to be chosen before applying the developed algorithm to the sensor data. Previous research has identified the complex Morlet wavelet as an appropriate wavelet function for bearing defect identification (Zhang et al., 2006b; Yan, 2007). Another advantage of the complex Morlet wavelet lies in its explicit expression in the frequency domain:

$$\Psi(f) = \frac{1}{f_b} e^{-\frac{\pi^2 (f - f_c)^2}{f_b^2}} \tag{1}$$

with the symbols f_b and f_c being the bandwidth and wavelet centre frequency parameters, respectively. This implies that no Fourier transform needs to be performed on the wavelet function when it comes to the code development, which saves memory buffers for data storage and improve the memory efficiency. Figure 7 illustrates the coded algorithm of the WES module.

Figure 6 Procedure of the Wavelet Envelope Spectrum algorithm (see online version for colours)

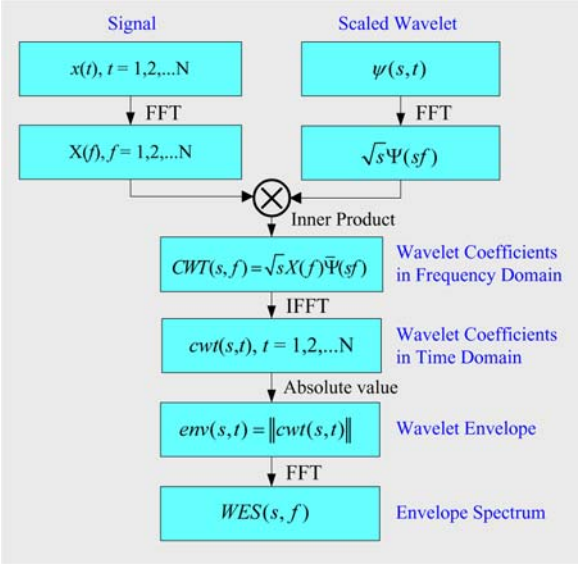
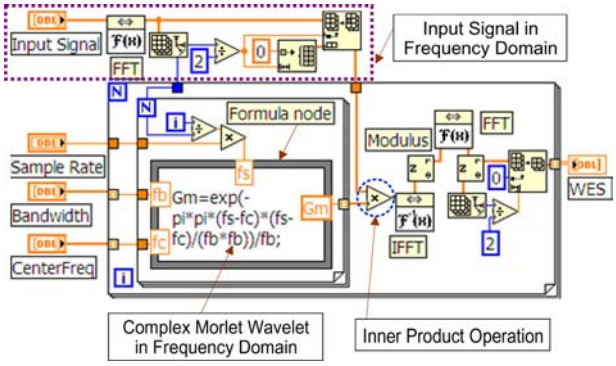


Figure 7 Code for performing the Wavelet Envelope Spectrum (see online version for colours)



It should be noted that instead of the scaling parameter s as commonly used in the analytical expression for wavelet transform, a pair of bandwidth f_b and centre frequency f_c parameters, as indicated in equation (1), are used to stretch the wavelet function, in the actual code realisation. Given the inputs from the sampling rate, the wavelet bandwidth and the centre frequency, the Morlet wavelet function in the frequency domain

(denoted as *Gm*) is designed and implemented in the module as a formula node, which is used to evaluate the mathematical expressions. By multiplying *Gm* with coefficients that have resulted from Fourier transform of the input signal, and then applying the inverse Fourier transform, the wavelet transformation of the input signal is realised. Further operation is taken through the '*complex-to-polar*' submodule to obtain the modulus of wavelet coefficients (i.e., wavelet envelope). Subsequent operation of Fourier transform on the wavelet envelope results in the WES.

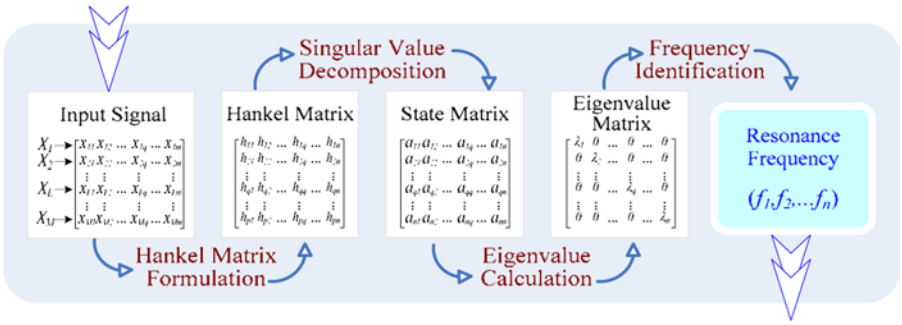
3.3 Stochastic subspace identification module

To ensure complete frequency coverage of the vibration signal when the WES module is executed, a data-driven scheme for wavelet centre frequency f_c selection is needed. The appropriate wavelet centre frequency should also be dynamically modifiable by the programme. This is realised by utilising the resonance frequencies of the spindle structure identified using the SSI technique (Tasker et al., 1998; Peeters and De Roeck, 1999; Kushnir, 2004). Instead of fitting an empirical model to the Frequency Response Function (FRF) using both the input and output measurement data as the traditional approach does (Németh et al., 2000), the SSI technique is taken to extract resonance frequencies from the spindle's measured output only. Mathematically, the SSI technique is formulated and solved using a discrete-time state space model of a linear, time-invariant system (e.g., the spindle) without known external inputs, according to the following equation:

$$\begin{cases} x_{k+1} = Ax_k + w_k \\ y_k = Cx_k + v_k \end{cases} \tag{2}$$

where $x_k = x(k\Delta t)$ is the discrete-time state vector, y_k is the system response vector, A is the state matrix and C is the output matrix. The two components, w_k and v_k, represent the disturbance noise to the spindle and measurement noise owing to sensor inaccuracy, respectively, and are stochastic in nature. Equation (2) indicates that the new state of the spindle system, x_{k+1}, can be obtained by the sum of the state matrix A multiplied with the old state vector x_k and the disturbance noise vector w_k. This means the dynamic behaviour of the spindle can be completely characterised by the state matrix A. As a result, resonance frequencies of the spindle system can be estimated from the eigenvalues of the state matrix A. Figure 8 illustrates the algorithm that identifies spindle resonance frequencies.

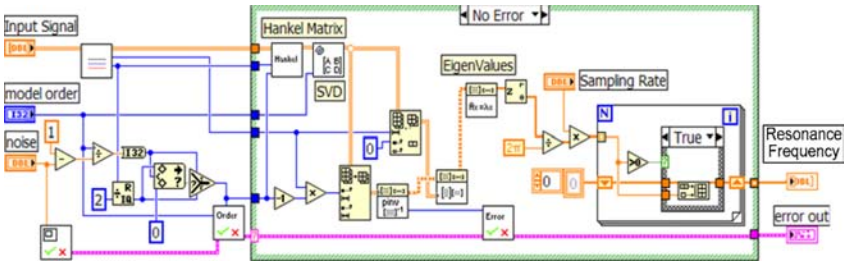
Figure 8 Algorithm for resonance frequency identification (see online version for colours)



As illustrated in Figure 8, vibration signals measured from multiple sensors are used to formulate the Hankel matrix. Numerical techniques, such as singular value decomposition, are then applied to estimate the state matrix A, which is subsequently decomposed to derive the corresponding eigenvalues. Through a mathematical transformation, the eigenvalues are converted to resonance frequencies.

On the basis of the developed algorithm, Figure 9 illustrates how the SSI module contained in the Signal-Processing layer is realised in the LabVIEW environment. As shown in Figure 9, the '*Hankel*' submodule is called first to construct the Hankel matrix from the input signals. Then, the '*Observability Matrix*' submodule, whose major function is the singular value decomposition, is called to estimate the state matrix A. After this step, the eigenvalues of the state matrix A are extracted by calling the "*Eigenvalues and Eigenvector*" submodule to estimate the resonance frequencies. Since a structural defect may excite the spindle system at any of the identified resonance frequencies, the equally spaced wavelet centre frequencies, which cover the range of these resonance frequency components, are chosen for implementing the WES algorithm.

Figure 9 Code for SSI-based resonance frequency identification (see online version for colours)



3.4 *Spindle health indicator module*

With output from the Signal-Processing layer, working condition of the spindle is determined by tracking the magnitude of the WES within the Condition-Monitoring layer. Furthermore, to present a generic data model for quantifying the working status of various types of spindles and machine tools, trending information on the magnitude of the WES needs to be recorded to construct a database, which is used subsequently for setting up a spindle health indicator in the Health Assessment layer. Since both the Condition-Monitoring layer and the Health Assessment layer utilise the WES results to fulfil respective functions, a single spindle health indicator module was designed in the present software to serve these two layers. The spindle health indicator includes two major components: health index, which provides a quantitative measure of the spindle defect severity level (e.g., healthy, small defect, medium defect and severe defect), and health status, which presents the type and location of the spindle defect. Both of them are established based on the SNR of the WES output. The SNR is defined as:

$$SNR = 10\log\frac{\text{Energy of defect-related charateristic frequency components}}{\text{Energy of other frequency components}}. \tag{3}$$

Figure 10 shows the flow chart of the spindle health indicator module. For the WES data, the '*SNR calculation*' submodule is called first at each pair of wavelet bandwidth and centre frequency. If the calculated SNR is larger than a predefined threshold value,

then "*Health Index Evaluation*" and "*Health Status Assessment*" submodules are called to update the spindle working conditions. The threshold value is also updated to the next condition level. Such a process is repeated for each sampling cycle when new WES data are generated from vibration signals measured from each sensor.

Figure 10 Flow chart of the spindle health indicator module (see online version for colours)

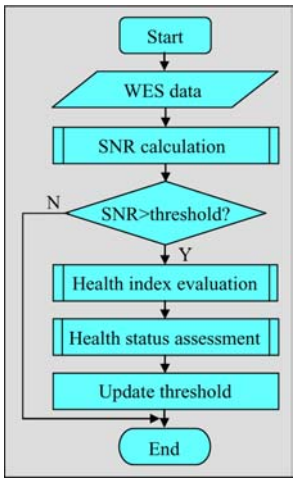
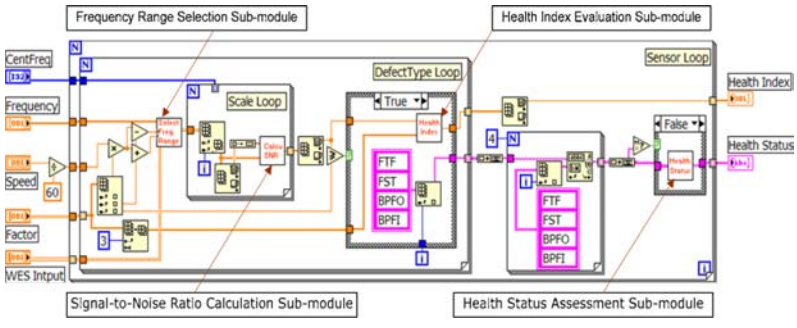


Figure 11 illustrates the corresponding code in LabVIEW. As indicated in Figure 8, before the SNR calculation, the magnitude information for several frequency intervals that cover possible defect-related frequency components (e.g., Ball Pass Frequency of the Inner raceway, denoted as *BPFI*) is extracted from the WES by a "*Frequency Range Selection*" submodule. The calculated SNR is then compared with the predefined thresholds at different levels. If the SNR is within the threshold interval, a corresponding health index is determined. Also, the health status (i.e., defect type and location) is derived through the defect-related frequency lines whose magnitudes are above the predefined thresholds.

Figure 11 Code for spindle health indicator module (see online version for colours)



3.5 Data management module

To log the spindle working condition (e.g., health index and health status) identified by the health indicator module and provide historical assessment values for subsequent

usage in the Prognostics Layer, a data management module, which archives the spindle working condition in a unified format, was designed in the Health Assessment layer. Because the *eXtensible Mark-up Language* (XML) provides standardised information exchange and is independent of the platform and language (W3C, 2006), an XML-based data format is adopted in the module development. A further advantage of the XML-based data format is that it can be used with a wide range of networking technology (e.g., Time Control Protocol/Internet Protocol (TCP/IP)) for data transfer (Lebold et al., 2003), and all of the information stored with such a format can be retrieved through web-based applications. Figure 12 illustrates the code implementation for logging health status and health index using XML. During each data sampling cycle, vibration signals are acquired and then processed to output spindle health status and index values. These values, together with a time stamp, are converted to XML format. On the basis of predefined XML schema, which is embedded in the LabVIEW development environment, the time information, spindle health status and spindle health index are logged into an XML file. Figure 13 shows an example on a piece of the XML-based data format, where the health index and health status of the spindle bearing are logged during one sampling cycle.

Figure 12 Code for data management module (see online version for colours)

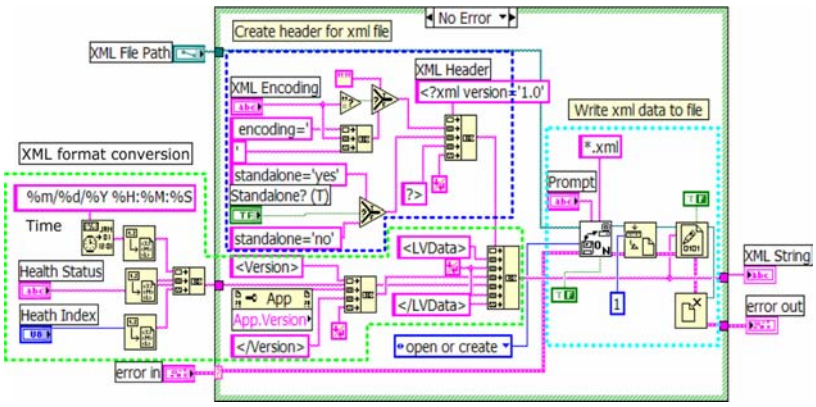


Figure 13 Representation of spindle health status and health index using XML-based data format (see online version for colours)

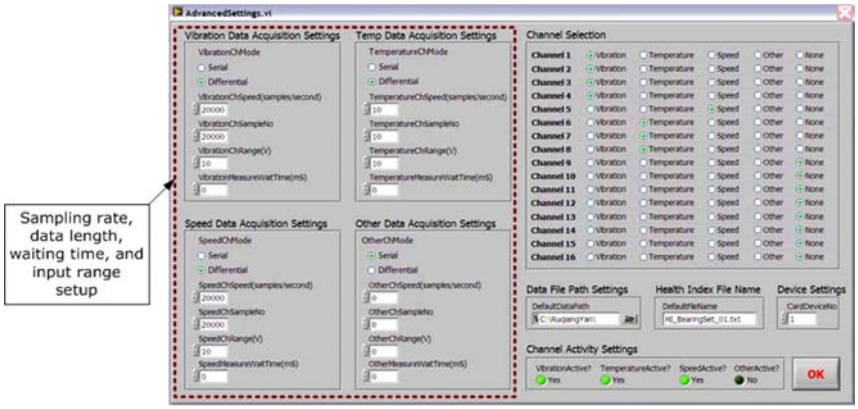
```
<?xml version="1.0" standalone="yes" ?>
- <LVData>
    <Version>8.0</Version>
  - <String>
      <Name>Sampling Time</Name>
      <Val>11/22/2006 10:58:56</Val>
    </String>
  - <String>
      <Name>Health Status</Name>
      <Val>Inner raceway defect</Val>
    </String>
  - <U8>
      <Name>Heath Index</Name>
      <Val>3</Val>
    </U8>
    ...
  </LVData>
```

Whereas the present software design does not include functions associated with the Prognosis and Decision-Support layers, they can be added into the system as individual modules if such requirement is needed. For example, a module for establishing an *empirical* model through continuous curve fitting of the combined historical and current assessment values can be designed in the Prognosis layer to extrapolate the trend of the spindle structural damage development and consequently, predict its future working condition. Also, based on information about the current and predict assessment values of the spindle structure, a module that utilises the e-mail notification function in LabVIEW can be designed in the Decision-Support layer to notify an expert who is not onsite to monitor the working condition of the spindle but can provide recommended actions remotely using web-based programme interface.

4 Software implementation and verification

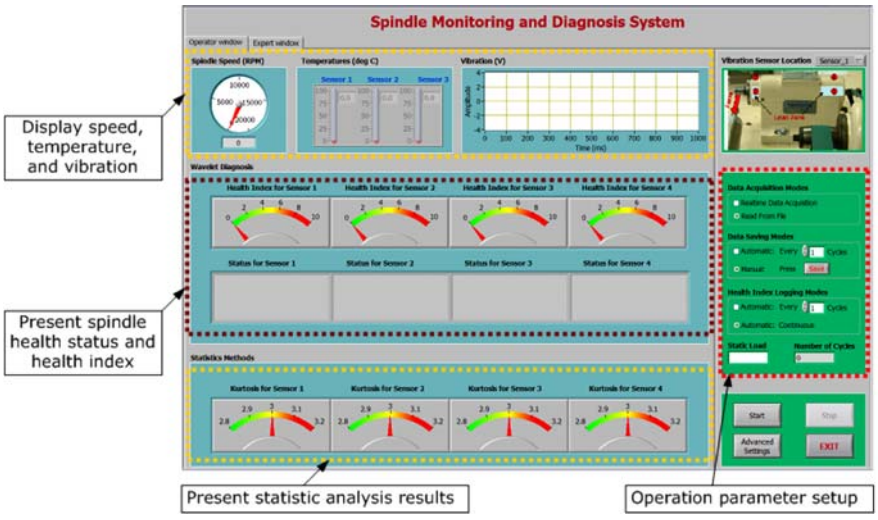
By integrating all the modules discussed in the previous section, the software is implemented to present a user-friendly human–machine interface. The design takes into account various aspects, such as visual appearance, ease of operation and accommodation of the needs for different users. Figure 14 illustrates the designed user interface of the spindle condition monitoring software. As shown in Figure 14(a), the parameter-setup window enables a machine operator to select parameters for data acquisition and archiving, such as device channels, sampling frequency and data length. It provides input to the data acquisition module such that the data acquisition board is initialised and ready to sample sensor measurement data. In addition, the right side of the user interface in Figure 14(b) and (c) also demonstrates several general parameter set-up options, such as the *data saving mode* and *spindle health index logging mode*. The left side of the user interface in Figure 14(b) and (c) has integrated previously designed modules and presents the spindle working conditions in two types of windows: a simplified spindle condition display window for machine operators (operator window, Figure 14(b)), and an advanced diagnosis window for machine experts (expert window, Figure 14(c)), respectively.

Figure 14 A Graphical User Interface for the designed software: (a) screenshot of the parameter-setup window; (b) screenshot of the operator window and (c) screenshot of the expert window (see online version for colours)

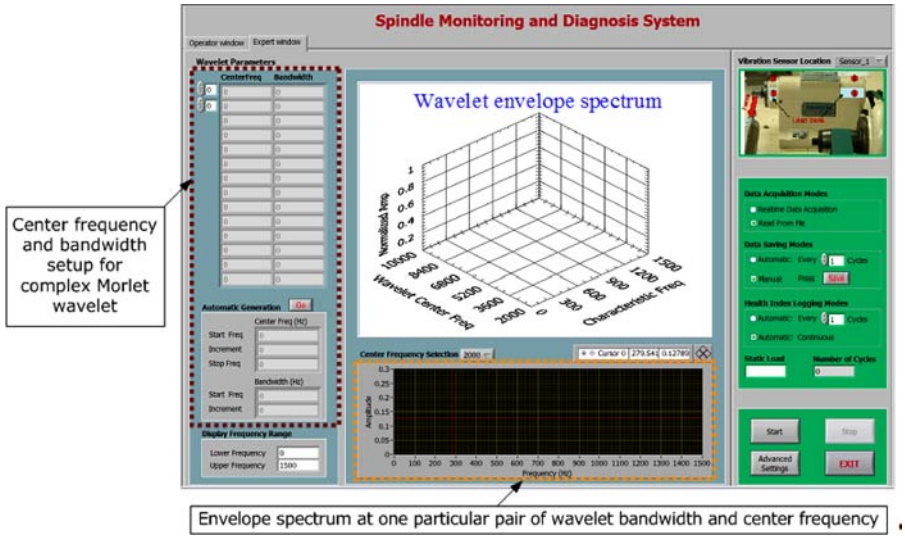


(a)

Figure 14 A Graphical User Interface for the designed software: (a) screenshot of the parameter-setup window; (b) screenshot of the operator window and (c) screenshot of the expert window (see online version for colours) (continued)



(b)



(c)

The operator window (as shown in Figure 14(b)) allows interactive communication between the machine operator and the software regarding the current status of the spindle, without the distraction from behind-the-window calculations. It displays the speed, temperature and vibration signals in real time. Furthermore, the health status and health index of the spindle are updated online based on the results from the embedded modules, and an alarm will be set off when defects are detected. A statistical parameter, Kurtosis, is also implemented in this operator window to track the working condition of the spindle.

The expert window shown in Figure 14(c) allows a machine expert to interactively adjust input parameters for the WES module to conduct a complete investigation of the spindle working condition. For example, the wavelet bandwidth and centre frequency can be adjusted based on the identified spindle resonance frequencies using the SSI module, thus enhancing online defect detection capability. Furthermore, quantitative evaluation of the envelope spectrum resulting from each pair of wavelet bandwidth and centre frequency is conducted at the bottom of the expert window, which provides detailed frequency information of the vibration signal.

To experimentally evaluate its performance, the designed software was installed and run on a dedicated computer for a custom-designed spindle test system as shown in Figure 3. During the first stage of the experimental study, no dynamic load was applied to the spindle to establish a reference base that characterises an undamaged, 'healthy' spindle. The sampling rates for temperature, speed and vibration sensors were set to 10 Hz, 1 kHz and 20 kHz, respectively. Signals measured from these sensors were continuously acquired and displayed on the computer through the GUI of the software, and subsequently processed to indicate the spindle working status. The measured spindle rotating speed was used to determine the potential defect-related characteristic frequencies, which was calculated as a constant related to the spindle rotating speed, as indicated in Table 1. The upper portion of Figure 15(a) displays the measured signals for one sampling cycle, where the spindle was operated at 8400 RPM. The results shown in the operator window (Figure 15(a)) indicate that the spindle was under a healthy condition (indicated by health index '*0*' and status '*Bearing Healthy*'). This is verified by the WES shown in the expert window (Figure 15(b)), in which no defect-related characteristic frequency components were identified.

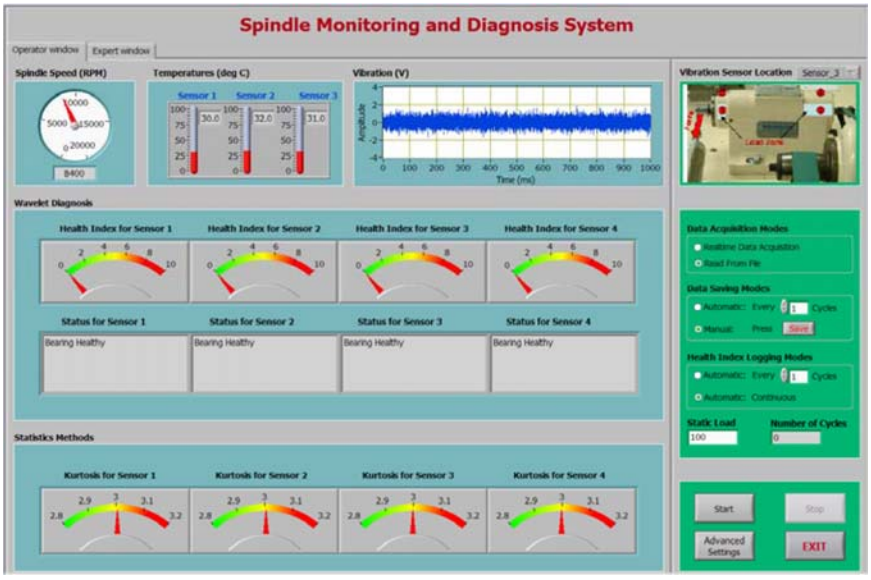
Table 1 Defect-related characteristic frequencies of the bearing

Defect type	*Characteristic frequency*
Unbalance	$f_r \approx rpm/60$
Rolling element	$f_{BSF} \approx 2.346 f_r$
Outer raceway	$f_{BPFO} \approx 4.414 f_r$
Inner raceway	$f_{BPFI} \approx 6.586 f_r$

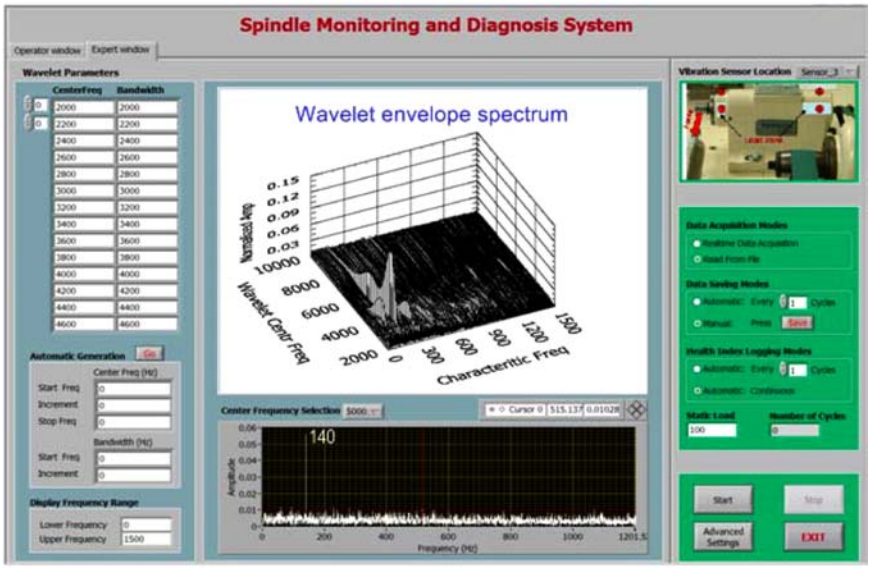
After establishing the reference base, dynamic impacts with a force of 13,300 N were applied to the rotating spindle to expedite degradation of the spindle structure. The time interval for the two successive impacts was set to 1 s. As shown in Figure 15(c), the software system diagnosed that a localised defect on its inner raceway had developed after 700 impacts. This is indicated in the operator window, where a health index value of 2 is shown, and the health status is described as '*Bearing Inner Raceway Defect*'. In addition, all the kurtosis values of the measured vibration signals deviated from the normal value of 3, which is characteristic for a healthy bearing. This is also verified by the WES shown in Figure 15(d), in which the frequency peak at 935 Hz is identified as the BPFI. Theoretically, the BPFI frequency at 8400 RPM is calculated to be 922 Hz using the equation given in Table 1. The 1.4% difference between the theoretical and experimental values can be traced back to the combined effect of rolling element slippage and the slight drift of spindle speed from the nominal input values to the spindle drive controller. The spectrum also displayed several other frequency peaks at

1075 Hz, 1215 Hz and 1355 Hz, respectively, which can be mathematically specified as $BPFI + k \cdot rpm$, with $k = 1, 2, \ldots, n$. They demonstrate the combined effect of spindle unbalance and inner raceway defect. From these experimental results, the effectiveness of the developed software for spindle condition monitoring is confirmed.

Figure 15 Experimental results of the test spindle: (a) screenshot of the expert window before impacts; (b) screenshot of the expert window before impacts; (c) screenshot of the expert window after 700 impacts and (d) screenshot of the expert window after 700 impacts (see online version for colours)

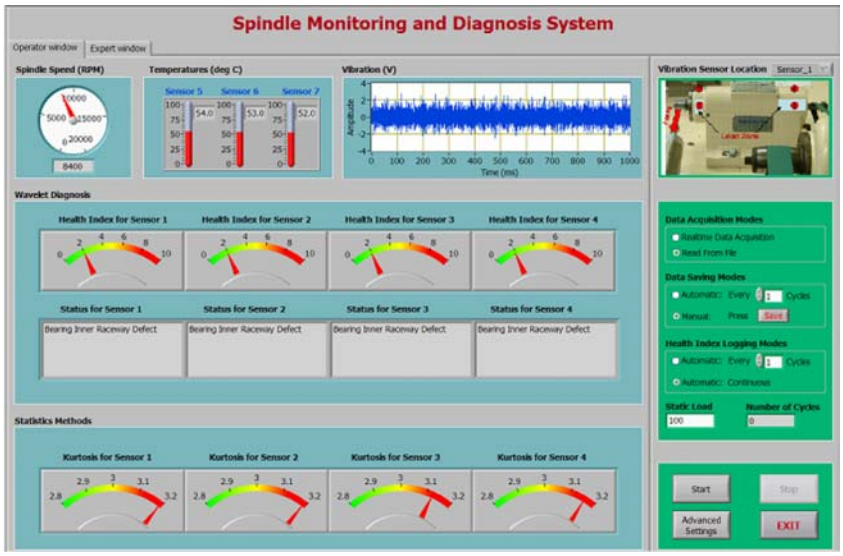


(a)

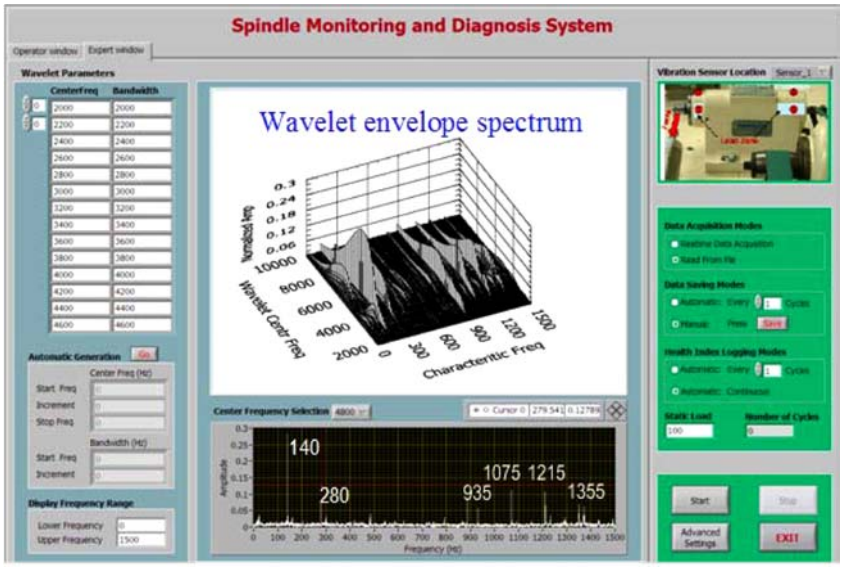


(b)

Figure 15 Experimental results of the test spindle: (a) screenshot of the expert window before impacts; (b) screenshot of the expert window before impacts; (c) screenshot of the expert window after 700 impacts and (d) screenshot of the expert window after 700 impacts (see online version for colours) (continued)



(c)



(d)

5 Conclusion

Consistent with the OSA-CBM architecture, a software package for online spindle condition monitoring and health diagnosis has been designed and realised using the

LabVIEW programming language. The software design takes a modular approach to implement each individual functions, thus providing flexibility to software update and functionality extension, in case such needs arise. Experimental evaluation conducted on a custom-designed spindle test system has demonstrated that the software has met the design requirements, and was able to detect bearing defects caused by accumulated dynamic impacts. The software is functionally adaptive and presents a new tool that enables effective and efficient monitoring and diagnosis of machine spindles, thus contributing directly to the development of a new generation of smart machine tools. In addition to spindles, the software can be applied to the health diagnosis of other types of rotary machines used in manufacturing systems.

Acknowledgements

This work was supported by the Smart Machining Systems Program of the National Institute of Standards and Technology (NIST). Help and suggestions provided by Dr. Steven Fick during the experimental studies are greatly appreciated. Any opinions, findings, conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of NIST.

References

Discenzo, F.M., Nickerson, W., Mitchell, C.E. and Keller, K.J. (1998) *Open Systems Architecture Enables Health Management for Next Generation System Monitoring and Maintenance*, http://www.osacbm.org/

Gelle, G., Colas, M. and Servière, C. (2003) 'Blind source separation: a new pre-processing tool for rotating machine monitoring?', *IEEE Transactions on Instrumentation and Measurement*, Vol. 52, No. 3, pp.790–795.

Hahn, S.L. (1996) *Hilbert Transform in Signal Processing*, Artech House, Inc., Norwood, MA.

Ho, D. and Randall, R.B. (2000) 'Optimization of bearing diagnostic techniques using simulated and actual bearing fault signals', *Mechanical Systems and Signal Processing*, Vol. 14, No. 5, pp.763–788.

Jones, R. (1996) 'Enveloping for bearing analysis', *Sound and Vibration*, Vol. 30, pp.10–15.

Kushnir, E. (2004) 'Application of operational modal analysis to a machine tool testing', *Proceedings of IMECE04*, 14–19 November, CA, USA, pp.57–62.

Lebold, M., Ferullo, D. and Reichard, K. (2003) 'An XML-based implementation of the OSA-CBM standard using SOAP over HTTP', *Proceedings of the 57th Meeting of the Society for Machinery Failure Prevention Technology*, 14–18 April, VA, USA, pp.231–246.

Lee, K., Gao, R. and Schneeman, R. (2002) 'Sensor network and information interoperability – integrating IEEE 1451 with MIMOSA and OSA-CBM', *Proceeding of IEEE Instrumentation and Measurement Technology Conference*, 21–23 May, AK, USA, pp.1301–1305.

Németh, J.G., Vargha, B. and Kollár, I. (2000) 'Online frequency domain system identification based on a virtual instrument, *IEEE Transactions on Instrumentation and Measurement*, Vol. 49, No. 6, pp.1260–1263.

NI LabVIEW User Manual (2003) National Instruments Corporation.

NIST (2005) http://www.mel.nist.gov/proj/pdf/sms.pdf

Peeters, B. and De Roeck, G. (1999) 'Reference-based stochastic subspace identification for output-only modal analysis', *Mechanical Systems and Signal Processing*, Vol. 13, No. 6, pp.855–878.

Prickett, P.W. and Johns, C. (1999) 'An overview of approaches to end milling tool monitoring', *International Journal of Machine Tools and Manufacture*, Vol. 39, pp.105–122.

Rehorn, A.G., Jiang, J. and Orban, P.E. (2005) 'State-of-the-art methods and results in tool condition monitoring: a review', *International Journal of Advanced Manufacturing Technology*, Vol. 26, pp.693–710.

Tasker, F., Bosse, A. and Fisher, S. (1998) 'Real-time modal parameter estimation using subspace methods: theory', *Mechanical Systems and Signal Processing*, Vol. 12, pp.795–806.

Thurston, M. and Lebold, M. (2001) 'Standards developments for condition-based maintenance systems', *Proceeding of the 55th Meeting of the Society for Machinery Failure Prevention Technology*, 2–5 April, VA, USA.

Tse, P.T., Peng, Y.H. and Yam, R. (2001) 'Wavelet analysis and envelope detection for rolling element bearing fault diagnosis – their effectiveness and flexibilities', *ASME Journal of Vibration and Acoustics*, Vol. 123, No. 4, pp.303–310.

W3C (2006) *Extensible Markup Language (XML) 1.0*, http://www.w3.org/TR/xml/

Wang, C. and Gao, R. (2000) 'A virtual instrumentation system for integrated bearing condition monitoring', *IEEE Transactions on Instrumentation and Measurement*, Vol. 49, No. 2, pp.325–332.

Wang, C. and Gao, R. (2003) 'Wavelet transform with spectral post-processing for enhanced feature extraction', *IEEE Transactions on Instrumentation and Measurement*, Vol. 52, No. 4, pp.1296–1301.

Yan, R. and Gao, R. (2003) 'A hybrid signal processing approach to sensor data analysis', *Proceedings of the ASME International Mechanical Engineering Congress and Exposition*, November, Washington, DC, pp.1159–1166.

Yan R. and Gao, R. (2004) 'Enhanced signal demodulation for machine health diagnosis', *Proceedings of Japan-USA Symposium on Flexible Automation*, paper # UL_074, 19–21 July, Denver, CO., USA, pp.1–8.

Yan, R. and Gao, R. (2006) 'Hilbert-Huang transform-based vibration signal analysis for machine health monitoring', *IEEE Transactions on Instrumentation and Measurement*, Vol. 55, No. 6, pp.1327–1334.

Yan, R. (2007) *Base Wavelet Selection Criteria for Non-stationary Vibration Analysis in Bearing Health Diagnosis*, PhD Dissertation, University of Massachusetts, Amherst.

Zhang, L., Gao, R. and Lee, K. (2006a) 'Optimization of sensor locations for spindle condition monitoring', *Transactions of the North American Manufacturing Research Institution of SME*, Vol. 34, pp.55–32.

Zhang, L., Gao, R. and Lee, K. (2006b) 'Spindle health diagnosis based on analytic wavelet enveloping', *IEEE Transactions on Instrumentation and Measurement*, Vol. 55, No. 5, pp.1850–1858.

Zhang, L., Yan, R., Gao, R. and Lee, K. (2007) 'Open architecture software design for online spindle health monitoring', *Proceeding of IEEE Instrumentation and Measurement Technology Conference*, 1–3 May, Warsaw, Poland, pp.1892–1897.