FAULT DIAGNOSIS

According to *Webster's New World Dictionary of the American Language,* the word diagnosis means "deciding the nature and the cause of a diseased condition of a machine, a process, or a system by examining the symptoms." In recent years, fault diagnosis has been playing an increasingly important role and expanding far beyond the traditional vibration analysis of mechanical systems and failure detection of control systems. This is due to the fact that machines, processes, and systems are becoming much more complicated, and the demand for better, faster, and more cost-effective performance is constantly increasing. It is also because the great advances in computer technology make fault diagnosis feasible and profitable.

THE PROCEDURE OF FAULT DIAGNOSIS

Regardless of the differences in machines, processes, and systems, it seems that most fault diagnosis follows a simple three-step procedure: (1) sensing (to acquire necessary information), (2) sensor signal processing (to capture the symptoms that characterize the faults), and (3) decision making (to determine the cause of the fault and the methods of correction, if applicable). Following this procedure, if the faults cannot be diagnosed by directly examining the sensor signals, then signal processing is needed; and if the signal processing fails to diagnose the faults, then decision making must be used.

SENSORS AND SENSING

Sensor signals are the window to the complicated world of the system. Sensing acquires necessary information for fault diagnosis. Depending on the applications, various sensors would be used. For electrical engineering applications, voltage and current are the most commonly used sensor signals. For mechanical engineering applications, typical sensors include force and pressure sensors; displacement, velocity, and acceleration sensors; heat and temperature sensors; flow sensors; sound and acoustic emission sensors; as well as optical sensors.

The choice of sensors depends on the physical properties of the application. In addition, various other factors must be considered, including cost, installation, sampling frequency (which must be greater than the Nyquist frequency), and number of samples. Should multiple sensors be used, it may be necessary to consider synchronization as well. For fault diagnosis, a rule of thumb in choosing sensors is to get close to the fault as much as possible. For example, for large rotating machinery such as turbine and power generators, mechanical faults often cause increased vibrations. To diagnose the cause of these faults, vibration sensors are used, such as eddy current displacement transducers and strain gauge accelerometers.

Sensing also involves acquiring data from the sensors. Today, fault diagnosis is usually done using computers. Hence, we will have to deal with digitized sensor signals.

SENSOR SIGNAL PROCESSING AND MODELING

Sensor signals contain the information necessary for fault diagnosis. However, they may also contain noises, including system noise, environment noise, and sampling noise. Hence, it is necessary to conduct signal processing to minimize the effect of the noises. Considering a sensor signal is composed of a number of components. The information may be associated with certain components while the noises are associated with the others. In this case, one can use filters for which the reader is referred to FILTERING THEORY.

Information may not be explicitly presented in the signal. To extract the information, which will be called features, various signal processing methods have been developed.

Time-Domain Method

Sensor signals are time series. Hence, the time-domain features of the signals are very important to fault diagnosis. Most time-domain features have clear physical meaning and can be obtained by means of simple calculations. Assuming that $x(t)$, $t = 1, 2, \ldots, N$ is a sensor signal, Table 1 presents a number of commonly used time-domain features with their mathematical definition and physical interpretation. These features are particularly useful if the signal is stationary or near stationary (a signal is stationary if the signal mean is a constant and the signal variance is independent of time). When a signal is nonstationary, we may use features such as rising rate, rising time, delay time, overshot, and steady state, as shown in Fig. 1.

The other useful time-domain features include envelop, short time energy, histogram, medium, mode, and number of threshold crossing. Also, before calculating the time domain features, the signals can be preprocessed by averaging:

$$y(t) = \frac{[x(t) + x(t+1)]}{2} \tag{1}$$

or by differencing:

$$z(t) = \frac{[x(t+1) - x(t)]}{T} \tag{2}$$

where T is the sampling frequency. Multiple steps of averaging and differencing could be applied as well. One may wonder what features should actually be selected and whether the selected features contain sufficient information for fault diagnosis. Unfortunately, there is no simple answer to these questions. As a special case, if a signal is stationary, then it can be characterized by sufficient statistics (1). For example, if a signal is stationary with normal distribution, then it can be described by its mean and variance. They are the sufficient statistics, which completely characterize the signal. On the other hand, if a signal is a period signal then one should use the frequency-domain method.

Frequency-Domain Method

Fault diagnosis using frequency-domain information is the most commonly used method today. It is known that the frequency-domain information can be obtained by means of the fast Fourier transform (FFT). Applying FFT to a signal results in a complex series:

$$X(f) = \mathrm{FFT}[x(t)] \tag{3}$$

where $f = (1/NT), (2/NT), \ldots, (1/2T)$ is the frequency index. The angular frequency $\omega = 2\pi f$ is often used for convenience.

Table 1. A List of Time-Domain Features and Their Mathematical Definition

Time-Domain Feature	Mathematical Definition	Physical Interpretation
Mean	$\overline{X} = \frac{1}{N}\sum_{t=1}^{N} x(t)$	The average value of the signal
Variance	$\sigma^2 = \frac{1}{N-1}\sum_{t=1}^{N} (x(t) - \overline{X})^2$	The variation of the signal
Root mean squares (rms)	$\text{rms} = \frac{1}{N}\sqrt{\sum_{t=1}^{N} x^2(t)}$	The energy of the signal
Skewness	$\text{SK} = \frac{N}{(N-1)(N-2)} \frac{\sum_{t=1}^{N} (x(t) - \overline{X})^3}{\sigma^3}$	The symmetry of the signal distribution
Kurtosis	$\text{KU} = \frac{\sum_{t=1}^{N} (x(t) - \overline{X})^4}{\sigma^4} - 3$	The shape of the signal
Maximum/minimum	$X_{\max} = \max\{x(t), t = 1, 2, \ldots, N\}$ $X_{\min} = \min\{x(t), t = 1, 2, \ldots, N\}$	The maximum/minimum of the signal
Range	$R = X_{\max} - X_{\min}$	The variation of the signal
Crest factor	$\text{CF} = \frac{R}{\overline{X}}$	The shape of the signal

Based on $X(f)$, the spectrum density, or simply the spectrum, can be found:

$$S(f) = \sqrt{\text{Re}^2[X(f)] + \text{Im}^2[X(f)]} \tag{4}$$

and the phase spectrum is

$$\Phi(f) = \arctan \frac{\text{Im}[X(f)]}{\text{Re}[X(f)]} \tag{5}$$

Note that in the spectrum, the frequency range is (0, 1/2T] with a resolution of $(1/NT)$. For example, sampling 1000 samples at a sampling frequency of 1 kHz ($T = 0.001$), the frequency range would be (0, 500 Hz] with a resolution of 1 Hz. If it is necessary to obtain the information at a specific frequency between two resolution frequencies, we can increase the sampling frequency, increase the number of samples, or use an approximation method (2). Also, we can use various windowing techniques to prevent the information lost (leakage) because of the limited samples.

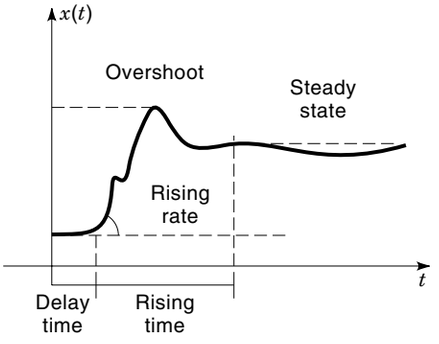

Figure 1. Illustration of several time-domain features for nonstationary signals.

Stationary signals can be described by their spectrum without information loss. From the spectrum, the frequency characteristics, denoted as a tuple $\{f, S(f), \Phi(f)\}$ (or $\{\omega, S(\omega), \Phi(\omega)\}$), can be represented in the form of a graph, also called a spectrum. Visual examination of the spectrum is called spectral analysis, requires skill and experience, and is often objective. The spectrum is usually stored in an array in a computer. To use computers for automated fault diagnosis, we need to characterize the spectra. Two types of frequency-domain characteristics are often used. The first one is the energy at specific frequency bands, which is the sum of square of $S(f)$ at the frequency bands. It is interesting to note that, according to the Passval formula, the energy of all the frequencies is related to the root mean square (rms) in time-domain, that is,

$$\frac{1}{\sqrt{N}} \sum_{f=-\infty}^{\infty} X^2(f) = \frac{1}{\sqrt{N}} \sum_{t=-\infty}^{\infty} x^2(t) = (\text{rms})^2 \tag{6}$$

The other type of frequency-domain features include peak value, peak frequency, natural frequency (ω_n), and damping ratio (ξ). The peak values are the local maximums in the spectrum. After finding the peak values, the corresponding peak frequencies can then be found. In general, there may be several peaks in a spectrum, and the corresponding frequencies are referred to as concerned frequencies (ω_i, $i = 1, 2, \ldots$). The concerned frequencies may include the machine rotating frequency and its harmonics, as well as the nature frequency of the machine. As an example, Fig. 2 illustrates a spectrum from a spindle vibration signal containing two concerned frequencies: ω_1 and ω_2. The first sharp peak is related to the rotation speed of the spindle. The second peak is related to the dynamics of the spindle system. The dynamics of the sys-

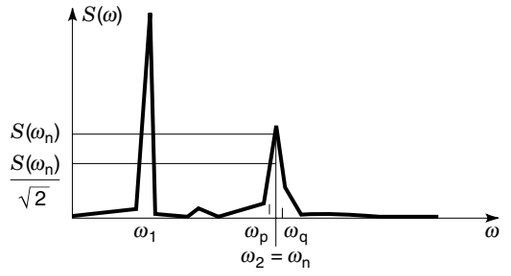

Figure 2. Illustration of several frequency-domain features.

tem can be characterized by its natural frequency, $\omega_n(\omega_n = \omega_2)$, and damping ratio, δ, which can be approximated by

$$\delta = \frac{\omega_q - \omega_p}{2} \tag{7}$$

where, ω_q and ω_p are the bandwidth frequencies as illustrated in Fig. 2. Frequency-domain information can also be obtained through the time-series model and the dynamic system models, which are discussed in a later section of this article. Frequency-domain features usually have clearly physical meanings. For instance, in the vibration signals from a rolling element bearing, there are characteristic frequencies associated to the out race, inner race, and rollers. By examining these frequencies, we can diagnose the bearing faults.

An extension of spectral analysis is modal analysis, which uses the frequency information from multiple vibration sensors to analyze the vibration of a structure or a machine. In particular, the frequency characteristics of the vibration are described by natural frequencies and the structural characteristics of the vibration are described by mode shapes. For the details of modal analysis, the reader is referred to SPECTRAL ANALYSIS.

Time-Frequency Method

Spectrum analysis is effective for stationary signals. For nonstationary signals (e.g., the signals whose frequency characteristics vary and/or amplitudes undulate), it may be necessary to use time-frequency domain information for fault diagnosis. The most commonly used method for analyzing time-frequency domain information is a waterfall diagram. A waterfall diagram is actually a number of spectra stacked together along a time axis. The use of the waterfall diagram is based on the assumption that within a short time period (e.g., in minutes or hours), the signal is nonstationary. The waterfall diagram is very useful for tracking slowing developed faults.

If a signal is nonstationary even within a short time period, then one can use time-frequency distributions. A number of time-frequency distributions have been developed. Among them, one of the most commonly used distribution is the Wigner-Ville distribution:

$$d(t,\omega) = \frac{1}{2\pi}\int\int\int e^{i(\xi\mu-\tau\omega-\xi t)}x(t+\tau/2)x^*(t-\tau/2)d\mu\, d\tau\, d\xi \tag{8}$$

where $x(\cdot)$ represents the sensor signal and $x^*(\cdot)$ is its complex conjugate. Another commonly used time-frequency distribution is the exponential time-frequency distribution (3):

$$d(t,\omega) = \int\int \frac{e^{-j\omega\tau}}{\tau\sqrt{4\pi/\sigma}}\, e^{(\mu-t)^2/4\tau^2/\sigma} x(t+\tau/2)x^*(t-\tau/2)d\mu\, d\tau \tag{9}$$

where σ is a scale factor and r is a constant. The exponential time-frequency distribution has a number of desirable properties; for example, its integration over time is equal to the ordinary spectrum and its integration over frequency is equal to the autocorrelation function.

Time-frequency domain information is usually described by a two-dimensional figure. Its quantitative analysis is similar to that of spectral analysis. One can use a baseline to compare against others, or use the energy in certain time windows and frequency bands as the fault indices. However, the applications of time-frequency distributions are often limited by the fact that an increase of time window would cause a reduced frequency resolution, which results in information loss. As a result, for nonstationary signals that are strongly time dependent, it would be difficult to capture the useful information at the right time with sufficient accuracy. This problem could be solved by using wavelet transform.

Wavelet Transform

Wavelet transform was first developed for image processing in the late 1980s and early 1990s. Since then, it has been applied to many fields with great success. Similar to the Fourier transform, the wavelet transform of a signal is an integration transform defined as follows (4):

$$W_s[x(t)] = \int_{-\infty}^{+\infty} x(\tau)\frac{1}{2}\Psi\left(\frac{t-\tau}{s}\right)d\tau \tag{10}$$

where $\tau = 1, 2, \ldots$ are times, $s = 1, 2, \ldots$ are scales, and $\Psi(\cdot)$ is the wavelet base function, also called the mother wavelet. The mother wavelet may take various forms, such as the Morlet's function, the Mexican hat function, the piecewise constant wavelet function, and the Lemarie and Battle's function, most of which are symmetric and continuous. (Hence, there are various wavelet transforms.) Different wavelets have different features, advantages, and limitations. Next, as shown in Fig. 3, through a process of dilation (which changes

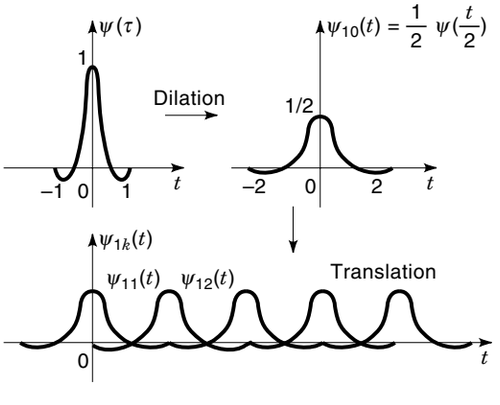

Figure 3. A mother wavelet, its dilation and translation.

the shape of the mother wavelet) and translation (which translates wavelet bases along the time axis), the mother wavelet generates a family of wavelet bases:

$$\Psi_{s\tau}(t) = \frac{1}{s}\Psi\left(\frac{t-\tau}{s}\right) \tag{11}$$

Each wavelet base represents a time window at a specific frequency band. Using the wavelet bases, a signal, $x(t)$, can be represented as follows:

$$x(t) = \frac{1}{C_\Psi}\int_{-\infty}^{+\infty}\int_{0}^{\infty} W_s[x(\tau)]\frac{1}{s}\Psi_{s\tau}(t)ds\,d\tau \tag{12}$$

where C_Ψ is a constant dependent on the base function. This implies that the signal can be decomposed onto the wavelet bases, and at the base $\Psi_{s\tau}(t)$ the weighting coefficient is $W_s[x(t)]$. Note that the wavelet bases are two-dimensional functions, and hence, like time-frequency distributions, wavelet transforms are two-dimensional transforms. Equation (12) is also called a reconstruction or inverse wavelet transform since it converts the wavelet function, $W_s[x(\tau)]$, back to its original.

A detailed description of wavelet transforms can be found in WAVELET TRANSFORMS. Briefly, all wavelet transforms possess four important properties:

1. *Multiresolution.* A wavelet transform decomposes a signal into various components at different time windows and frequency bands. These components form a surface in a time-scale plane. The size of the time window is controlled by the translation, while the length of the scale is controlled by the dilation. Hence, one can examine the signal at different time windows and scales by controlling the translation and the dilation. This is called multiresolution. In comparison, time-frequency distributions use only fixed time windows and frequency bands.
2. *Localization.* As shown in Fig. 3, the dilation changes the shapes of the wavelet bases. The smaller the dilation j, the sharper the shape. On the other hand, the translation shifts the wavelet bases along a time window. By controlling the dilation and the translation, specific features of a signal at any specific time-scale can be explicitly obtained. Called localization, this allows us to magnify specific features of the signal. In comparison, in time-frequency distributions, the information in every time-frequency window can only be equally weighted.
3. *Zoom-in and zoom-out.* From Fig. 3, it is seen that the time window and the scale of the wavelet bases change correspondingly through the dilation. The wider the time window, the narrower the scale, and vice versa. This is called zoom-in and zoom-out. It implies that the wavelet transforms are capable of capturing both the short-time high-frequency information and the long-time low-frequency information of the signal. In comparison, in the Fourier transforms and time-frequency distributions, an increase of time window causes reduced frequency resolution and, hence, results in information loss.
4. *Reconstruction.* A signal $f(t)$ can be reconstructed from its wavelet transform at any resolution without information loss. These features make the wavelet transforms very effective for analyzing nonlinear time-varying sensor signals.

Equation (11) represents continuous wavelet transforms. For digitized signals, discrete wavelet transforms should be applied (4) in which the scale parameter, s, is taken as an integer of base 2 (i.e., $s = 2j$, $j = 1, 2, \ldots$) and the time parameter, τ, is taken as a series of integer k (i.e., $\tau \to k = 1, 2, \ldots$). That is,

$$\psi_{jk}(t) = \frac{1}{2^j}\Psi\left(\frac{t}{2^j} - k\right) \tag{13}$$

Discrete wavelet transform can be calculated recursively. Given the wavelet base function, $\Psi(t)$, and an orthogonal function, $\phi(t)$, there exists a pair of mirror filters, $h(t)$ and $g(t)$:

$$\phi_j(t) = h(t) * \phi_{j-1}(t) \tag{14}$$

$$\Psi_j(t) = g(t) * \Psi_{j-1}(t) \tag{15}$$

where * denotes convolution. Furthermore, let operators H and G be the convolution sum:

$$H = \sum_k h(k-2t) \tag{16}$$

$$G = \sum_k g(k-2t) \tag{17}$$

Then the discrete wavelet transform can be represented as follows:

$$A_j[x(t)] = H\{A_{j-1}[x(t)]\} \tag{18}$$

$$D_j[x(t)] = G\{A_{j-1}[x(t)]\} \tag{19}$$

where $A_j[x(t)]$ is called the (wavelet) approximation and $D_j[x(t)]$ is called the detail signal, which represents information loss. It is seen that the binary wavelet transforms uses H and G only on the approximation $A_{j-1}[f(t)]$ and, hence, loss information at each recursive step. If the operators H and G are applied on both $A_{j-1}[f(t)]$ and $D_{j-1}[f(t)]$, then the wavelet packet transform is delivered:

$$A_j[f(t)] = H\{A_{j-1}[f(t)]\} + G\{D_{j-1}[f(t)]\} \tag{20}$$

$$D_j[f(t)] = G\{A_{j-1}[f(t)]\} + H\{D_{j-1}[f(t)]\} \tag{21}$$

Let $P_j^i(t)$ be the ith packet on jth resolution; then the wavelet packet transform can be computed by the following recursive algorithm:

$$P_0^1(t) = x(t) \tag{22}$$

$$P_j^{2i-1}(t) = HP_{j-1}^i(t) \tag{23}$$

$$P_j^{2i}(t) = GP_{j-1}^i(t) \tag{24}$$

where $t = 1, 2, \ldots, 2^{J-j}$, $i = 1, 2, \ldots, 2^j$, $j = 1, 2, \ldots, J$, and $J = \log_2 N$.

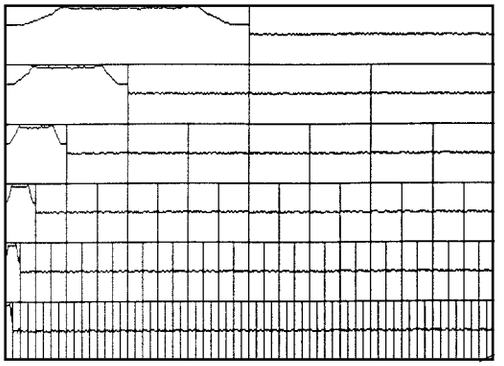

Figure 4. Example of wavelet packet transform.

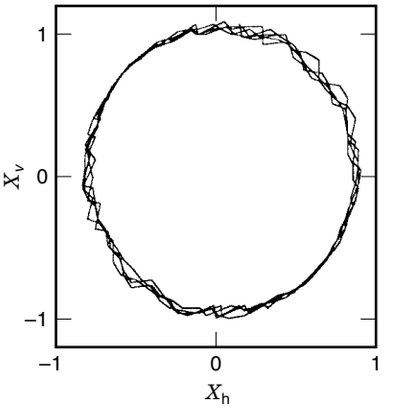

Figure 6. Example of a filtered orbit diagram.

Figure 4 shows an example of a wavelet packet transform. It is seen that the signal is decomposed into a number of packets, with each packet representing a component of the signal at a specific time window and frequency band. This is the multiresolution. We can focus on selected packets. This is the localization. Also, we can examine a larger packet at lower resolution or smaller packets at higher resolution. This is the zoom-in and zoom-out.

The quantitative description of the wavelet packet transform of a signal involves the packet selection and packet characterization. The selected packets should contain principal components of the original signal (5). For example, in Fig. 4 the selected packets will be $P_5^2(t)$ and $P_5^{12}(t)$. Furthermore, each packet can be viewed as a compressed or filtered time series and hence can be described by the time-domain indices and/or frequency-domain indices discussed previously.

Time-Space Method (Orbit Diagram)

In some applications, sensor signals may contain spatial information. For example, the vibration of a rotating machinery is in two dimensions and the force of a machining process is in three dimensions. The time-space domain information represents the spatial coordination of a system and is often used for fault diagnosis.

To capture the spatial information, sensors must be built in a specific configuration. Figure 5 illustrates a typical sensor setup used in large rotating machinery. It consists of two vibration (displacement) sensors set up perpendicularly; the sensor signals, X_h and X_v, are sensed simultaneously.

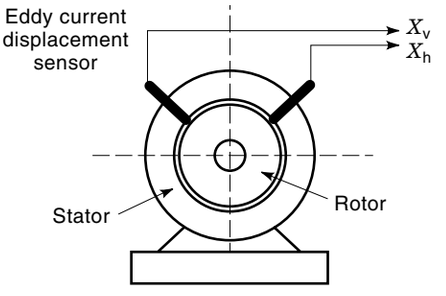

Figure 5. A typical sensor setup used in diagnosis of large rotating machinery.

The spatial correlation of the two sensor signals can be obtained by the orbit diagram and/or filtered orbit diagram. The orbit diagram is obtained by simply plotting the time-domain signals, X_h and X_v, against each other at same time instance in a two-dimensional plot. Due to the effect of noises, orbit diagrams often exhibit unrecognizable random patterns. In this case, the filtered orbit diagrams should be used. The filtered orbit diagram is obtained in two steps: First, the sensor signals, X_h and X_v, are filtered by a non-phase-shifting band-pass filter. Then the filtered sensor signals are plotted against each other just like the orbit diagram. One may also use a keyphasor or an encoder mounted on the shaft to relate sensor signals in an orbit diagram to an angular orientation of the shaft.

The orbit diagram (or filtered orbit diagram) represents the spatial information of the sensor signals. Take, for example, two signals:

$$x_h(t) = \sin(\omega t) \tag{25}$$

$$x_v(t) = \sin(\omega t + 90^\circ) \tag{26}$$

The orbit diagram is a unit circle. On the other hand, if the two signals have no phase difference, the orbit diagram will be a straight line. As an example, Fig. 6 shows a filtered orbit diagram from a rotating machinery. From the figure, it is seen that the vibration in the x direction is the same as in the y direction. According to the analysis above, this indicates that the two signals are 90° apart in phase, which could be caused by an unbalanced mass hitting the two sensors 90° apart spatially.

For complicated signals (e.g., signals consisting of many frequency components), quantitative description of orbit diagrams becomes very difficult. Hence, the use of orbit diagrams may not be automated.

A related technique is the phase diagram. It depicts the relationship between a signal and its derivative. For a unit sine waveform, the phase diagram is an unit circle. Similar to the orbit diagram, it has clear physical meaning. For example, a signal phase difference across a coupling typically indicates misalignment. Also, the phase difference from one end of a rotor to another may indicate a coupled imbalance or looseness. Similar to the use of orbit diagrams, an inherited problem in the use of phase diagram is the quantification and interpretation of the diagram.

Frequency-Space Method (Holospectrum)

Arguably, the most effective tool for analyzing spatial information is the holospectrum, which describes the frequency-space domain information of the signals. The basic idea of the holospectrum is rather straightforward. Using the preceding notation, the signals from the horizontal sensor, X_h, will be described by $\{\omega, S_h, \Phi_h\}$, and the signals from the vertical sensor, X_v, will be described by $\{\omega, S_v, \Phi_v\}$. Furthermore, let us assume that the concerned frequencies are $\omega_1, \omega_2, \ldots, \omega_n$. Then, the Fourier approximations of the signals are

$$X_h(\omega_i) = \sum_{i=1}^{n} A_h(\omega_i) \exp(-\delta_h(\omega_i)t) \sin\big(\omega_i t + \phi_h(\omega_i)\big) \tag{27}$$

$$X_v(\omega_i) = \sum_{i=1}^{n} A_v(\omega_i) \exp(-\delta_v(\omega_i)t) \sin\big(\omega_i t + \phi_v(\omega_i)\big) \tag{28}$$

Assuming that $\delta = 0$, which is true for most mechanical systems, then at each frequency ω_i the frequency characteristic of the signal is described by the amplitude $A(\omega_i)$ and phase $\phi(\omega_i)$ forming an ellipse in a two-dimensional space. The holospectrum is composed of a number of such ellipses. Figure 7 shows an example of holospectrum.

Holospectrum can be described quatitatively. At the frequency ω_i, denote

$$A_i = A_h^2(\omega_i) + A_v^2(\omega_i) \tag{29}$$

$$B_i = 2|A_h(\omega_i)A_v(\omega_i)\sin(\phi_h(\omega_i) - \phi_v(\omega_i))| \tag{30}$$

Then the major axis and the minor axis of the corresponding ellipse in the holospectrum are

$$2a_i = \sqrt{A_i + B_i} + \sqrt{A_i + B_i} \tag{31}$$

$$2b_i = \sqrt{A_i + B_i} - \sqrt{A_i + B_i} \tag{32}$$

The eccentric ratio of the ellipse is

$$e_i = \frac{\sqrt{a_i^2 - b_i^2}}{a_i} \tag{33}$$

and the inclination angle (i.e., the angle between the major axis and the horizontal axis) is

$$c_i = \cos^{-1} \pm \sqrt{\frac{1 - b_i/S_h(\omega_i)\sin(\Phi(\omega_i) - \Phi(\omega_i)}{1 - (b_i/a_i)^2}} \tag{34}$$

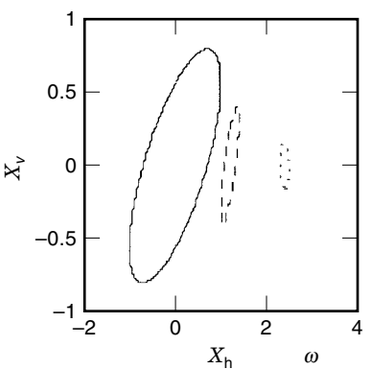

Figure 7. Example of a holospectrum.

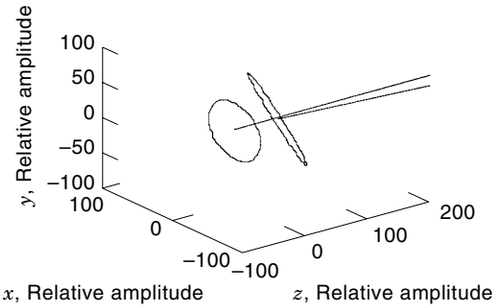

Figure 8. Example of a four-dimensional holospectrum.

where a positive sign is used if $\cos(\phi_v(\omega_i) - \phi_h(\omega_i)) > 0$; otherwise, a negative sign is used.

The indices a_i, b_i, c_i, and d_i quantitatively describe the ellipse, which in turn describes the spatial correlation of the sensor signals. Following the preceding rotating machinery vibration analysis example, at the rotating frequency ω_1, $a_1 = b_1$ (i.e., the ellipse becomes a circle) implies that the vibration amplitudes in both horizontal and vertical directions are the same, but the vibration phases are 90° apart. This would indicate that the machinery is in a state of unbalance because the unbalance mass hits the two sensor exactly 90° spatially.

If the sensor signal is a three-dimensional signal, such as force, then we can use the four-dimensional (three spatial dimensions plus the frequency dimension) holospectrum (6). Similar to the holospectrum, at a concerned frequency, ω_i, the signals can be approximated by

$$X_x(\omega_i) = A_x(\omega_i) \exp(-\delta_x(\omega_i)t) \sin\big(\omega_i t + \phi_x(\omega_i)\big) \tag{35}$$

$$X_y(\omega_i) = A_y(\omega_i) \exp(-\delta_y(\omega_i)t) \sin\big(\omega_i t + \phi_y(\omega_i)\big) \tag{36}$$

$$X_z(\omega_i) = A_z(\omega_i) \exp(-\delta_z(\omega_i)t) \sin\big(\omega_i t + \phi_z(\omega_i)\big) \tag{37}$$

Again, assuming that $\delta = 0$, the preceding equations represents an elliptic curve in three-dimensional space. A four-dimensional holospectrum consists of several such curves, and each curve describes the spatial-frequency correlation of the signals at a concerned frequency. An example of a four-dimensional holospectrum is shown in Fig. 8. The quantitative indices of a four-dimensional holospectrum include the major and minor axes, the eccentric ratio and the inclination angle of the ellipses, and the orientation of the ellipses (i.e., whether the ellipse is formed clockwise or counterclockwise).

Other Signal Processing Methods

There are several other signal processing methods that have been used for fault diagnosis. These include the higher-order spectrum and cepstrum. The higher-order spectrum is another technique for nonstationary sensor signal processing. The motivation of using higher-order spectra is 3-fold: (1) to extract information due to deviations from Gaussian distributions, (2) to estimate the phase information of non-Gaussian signals, and (3) to detect and characterize the nonlinear properties of mechanisms that generate time series via phase relations of their harmonic components.

The most commonly used higher-order spectra is the bispectrum:

$$B(\omega_1, \omega_2) = \sum_{\tau=-\infty}^{\infty} \sum_{\nu=-\infty}^{\infty} b(\tau, \nu) e^{-j(\omega_1 \tau + \omega_2 \nu)} \tag{38}$$

where $b(\tau, \nu) = E\{x(t)x(t + \tau)x(t + \nu)\}$ is the third-order moment of the signal. The bispectrum has a number of distinct properties for stationary signals, and it is capable of representing the phase information of nonstationary signals (7). However, it is often difficult to perceive the physical meaning of the higher-order spectrum.

Cepstrum is the spectrum of the spectrum. It is obtained by taking a Fourier transform of the Fourier transform of a signal. It relates to the phase information of the signal.

Time-Series Models

The signal processing methods discussed previously are based on examination of the appearance of the signal in the time domain, frequency domain, time-frequency domain, and frequency-space domain. Another type of signal processing method is to model the signal using specific models, among which the most popular one is the time-series models.

Assuming that the sensor signal, $\{x_1, x_2, \ldots, x_n\}$, or denoted as $\{x_t, t = 1, 2, \ldots, n\}$, is the output of a dynamic system, then the system's current output is likely dependent on the system's previous output. Assume such a dependence is linear; then

$$X_t - \phi_1 X_{t-1} - \phi_2 X_{t-2} -, \ldots, -\phi_p X_{t-p} = a_t \tag{39}$$

where p is the order of the system and a_t represents an impetus, called shock or noise, which induces the variation to the system output. Equation (39) is called an autoregressive (AR) model. Assuming further that the impetuses affect the system output in several steps (e.g., yesterday's cold front affects today's temperature), then

$$\begin{aligned} X_t - \phi_1 X_{t-1} - \phi_2 X_{t-2} -, \ldots, -\phi_p X_{t-p} \\ = a_t - \theta_1 a_{t-1} -, \ldots, -\theta_q a_{t-q} \end{aligned} \tag{40}$$

where q is the order of the moving average (MA) part of model. Equation (40) is called an autoregressive and moving average (ARMA) model. In general, we assume that the parameters of the model $\{\phi_1, \phi_2, \ldots, \phi_p, \theta_1, \theta_2, \ldots, \theta_q\}$ are constants and a_t is a white noise $a_t \sim N(0, \sigma_a)$. By introducing the back-shift operator B (i.e., $BX_t = X_{t-1}$), the ARMA model can be rewritten to a compact form:

$$\Phi(B)X_t = \Theta(B)a_t \tag{41}$$

where $\Phi(B) = 1 - \phi_1 B - \phi_2 B^2 - , \ldots, \phi_p B^p$ and $\Theta(B) = 1 - \theta_1 B - \theta_2 B^2 - , \ldots, - \theta_q B^q$. The ARMA model may look simple, but it is actually a nonlinear model since at the right hand side of the equation both the model parameters $\{\theta_1, \theta_2, \ldots, \theta_q\}$ and the noise series $\{a_t, t = 1, 2, \ldots, n\}$ are unknown (though their statistical properties are known). This makes the construction of the model mathematically and computationally complicated. In general, building an ARMA model consists of two steps: (1) determining the structure of the model (the orders of AR and MA as well as the nonzero terms in the model, if applicable); and (2) estimating the model parameters. According to literature, dozens of methods have been developed, though none has been proved better than the others for all applications.

In general, there are two ways to use time-series models for fault diagnosis. They both are based on the assumption that the faults will result in a change of the time-series models. The first method is the prediction error method. Assume that a time-series model is built using the data obtained when the system is known in normal condition and denote it as $(\hat{\phi}_1, \hat{\phi}_2, \ldots, \hat{\phi}_p, \hat{\phi}_1, \hat{\phi}_2, \ldots, \hat{\phi}_q)$. This model can be considered as a filter, which transforms the correlated time series, x_t, to an uncorrelated white noise series, a_t. When a new time series is filtered through the model, the prediction error, $\hat{a}_t$ can be computed recursively using the following equation:

$$\begin{aligned} \hat{a}_t = X_t - \hat{\phi}_1 X_{t-1} - \hat{\phi}_2 X_{t-2} -, \ldots, -\hat{\phi}_p X_{t-p} \\ + \hat{\theta}_1 \hat{a}_{t-1} +, \ldots, +\hat{\theta}_q \hat{a}_{t-1}, t > q \end{aligned} \tag{42}$$

If the new data correspond to the normal condition, then, according to the definition, the prediction error series should be a white noise. On the other hand, if the new data correspond to a fault, the prediction error series would not be a white noise as the data correspond to a different model. To examine whether the series $\hat{a}_t$ is a white noise series, we can use the Quantile-Quantile (Q–Q) plot. If $\hat{a}_t$ is a white noise series, then it must conform to a normal distribution $N(\mu_a, \sigma_a)$ and the following relationship must be true:

$$\hat{a}_t = \mu_a + \sigma_a Z_t \tag{43}$$

where Z_t is a random variable conforming to the standard normal distribution $N(0, 1)$. To test whether there exists a linear relationship between a_t and Z_t, first rearrange a_t in ascending order. Then for the kth data, there are k/N values less than or equal to it as it is the (k/N)th sample percentile. If a_t is normally distributed, it should be linearly related to the (k/N) percentile of $N(0, 1)$, which can be found from statistics books. In other words, plotting the rearranged a_t against the Z_t, a straight line would indicate such a linear relationship. Otherwise, the relationship is nonlinear, which, in turn, implies that a_t is not a white noise series and it must correspond to a fault.

The second method is to examine the variation of the model parameters. In general, the parameters of the model $\{\phi_1, \phi_2, \ldots, \phi_p, \theta_1, \theta_2, \ldots, \theta_q\}$ do not have physical meanings and, hence, are inconvenient to use. However, we can examine the roots of $\Phi(B)$, or the eigenvalues of the model. It is known from a pair of eigenvalues, λ_1 and λ_2, that we can calculate the natural frequency (ω_n) and the damping ratio (ζ):

$$\omega_n = \frac{1}{T} \sqrt{\frac{[\ln(\lambda_1 \lambda_2)]^2}{4} + \left[\cos^{-1} \frac{\lambda_1 + \lambda_2}{2\sqrt{\lambda_1 \lambda_2}}\right]^2} \tag{44}$$

$$\zeta = \frac{-\ln(\lambda_1 \lambda_2)}{\sqrt{\dfrac{[\ln(\lambda_1 \lambda_2)]^2}{4} + 4\left[\cos^{-1} \dfrac{\lambda_1 + \lambda_2}{2\sqrt{\lambda_1 \lambda_2}}\right]^2}} \tag{45}$$

Obviously, if we build two time-series models from two sets of data, both obtained from the same system condition, then

the eigenvalues (the natural frequencies and damping ratios) of the two models will be rather similar (though the parameters of the two models may not). On the other hand, a change of the eigenvalues would indicate the change of the system and may correspond to a fault. Since many systems can be modeled by time-series models and their eigenvalues have distinct meanings, the time-series models are often used for fault diagnosis. However, it should be noted that time-series models are sensitive to not only the system health conditions but also to the system working conditions, and they do not work for nonlinear systems.

For nonlinear systems, we may use nonlinear time-series models, for which the reader is referred to AUTOREGRESSIVE PROCESSES. Also, the idea of using time-series models for fault diagnosis can be extended to using other systems models such as transfer function models and state-space models.

Remarks on Using Signal Processing Methods

In summary, the following rules are recommended for choosing signal processing methods for fault diagnosis:

1. Start at time-domain features such as mean, variance, rms, skewness, kurtosis, and crest factor. Also, use histograms, threshold crossing counts, as well as other special features. Averaging, differentiating, and filtering will be helpful. Note that before calculating these features, applying a band–pass filter to the signals is always helpful.
2. If the signal is stationary, use frequency-domain features and spectral analysis.
3. If the signal is nonstationary, use wavelet transform.
4. If the signal has spatial information, use holospectrum, 4D holospectrum, or orbit diagram.

SENSOR SIGNAL CLASSIFICATION

The sensor signal processing techniques described in the previous section are usually effective in detecting the faults. However, to diagnose the faults (i.e., to pin-point the cause of the faults) requires to extract distinct signal features that are correlated to each and every specific fault. This is much more difficult due to the following four reasons: (1) Engineering systems may be very complicated and the sensor signals are just a window to the system providing only limited information; (2) The signal processing techniques used may introduce distortions, such as phase shifting, causing the loss of information; (3) The system operating conditions may vary (e.g., the change of speed and/or the load) resulting mixed information; and (4) The system may be affected by various noise disturbance, such as environment noise and sampling noise. Consequently, for fault diagnosis, it is often necessary to conduct signal classification to correlate signals (or the features of the signals) to the specific faults.

Let us assume that there exists a relationship between a fault, denoted as c (there may be many different faults: c_1, c_2, . . ., c_m), and the signal features, denoted by a vector $\boldsymbol{x}$. As shown in Fig. 9, the fault can be considered as the input, the features are the outputs, and the relationship is a function $R(\cdot)$. Mathematically, the relationship can be represented as

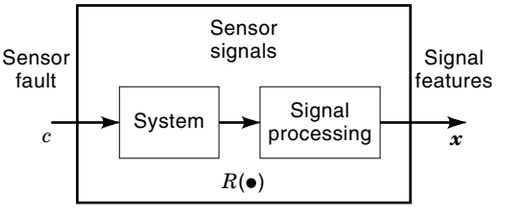

Figure 9. The model of signal classification.

$$\boldsymbol{x} = R(c) \tag{46}$$

Note that the relationship may take various forms, such as patterns, fuzzy membership functions, decision rules, and artificial neural networks (ANN). It is the key to the signal classification.

In general, signal classification consists of two phases: learning and reasoning. In the learning phase, also called training, the relationship $R(c)$ is built based on available learning samples and domain knowledge, or a combination of both. The reasoning, also called classification, can be viewed as an inverse operation: Based on the relationship, estimate the corresponding system condition of a new sample, $\boldsymbol{x}$; that is,

$$c = R^{-1}(\boldsymbol{x}) \tag{47}$$

where, depending on the forms of relationship R, the inverse of relationship, R^{-1}, may be pattern matching, fuzzy classification, decision tree searching, and ANN classification.

In general, assume that through sensing and signal processing, we obtain N sets of training samples from m different system conditions, which may include the normal system condition and various faults. The system conditions will be referred to as classes and denoted as $c_1, c_2, \ldots, c_m$. On the other hand, each sample is described by a set of signal features X_1, $X_2, \ldots, X_i, \ldots, X_n$. Note that the signal features may be the signal itself or the features of the signal, such as mean and variance. Arrange the training samples as in Table 2, where $c(\boldsymbol{x}_i) \in \{c_1, c_2, \ldots, c_m\}$ implies that the sample $\boldsymbol{x}_i = \{x_{i1}, x_{i2}, \ldots, x_{in}\}$ is from one of the predefined classes. Note that both the values and the classes of all the training samples must be known.

Although many classification methods are available, from a mathematical point of view, what these methods do is either weighting or decomposition. Figures 10(a) and 10(b) show a simple example where two features, X_1 and X_2, are used to classify two classes, c_1 and c_2. In Fig. 10(a), a partition line is used to separate the two classes. The partition line can be represented by

$$a_0 + a_1 X_1 + a_2 X_2 = 0 \tag{48}$$

where a_0, a_1, and a_2 are constants. This is called the weighting method because the classification is determined by the

Table 2. The Organization of the Training Samples

	X_1	X_2	. . .	X_i	. . .	X_n	Class
$\boldsymbol{x}_1$	$x(1, 1)$	$x(1, 2)$	. . .	$x(1, i)$	. . .	$x(1, n)$	$c(\boldsymbol{x}_1)$
$\boldsymbol{x}_2$	$x(2, 1)$	$x(2, 2)$	. . .	$x(2, i)$	. . .	$x(2, n)$	$c(\boldsymbol{x}_2)$
. . .	. . .	. . .	. . .	. . .	. . .	. . .	. . .
$\boldsymbol{x}_N$	$x(N, 1)$	$x(N, 2)$	. . .	$x(N, i)$	. . .	$x(N, n)$	$c(\boldsymbol{x}_N)$

"weighting" factors a_0, a_1, and a_2. For a new sample $\boldsymbol{x} = \{x_1, x_2\}$, the classification rule is represented as follows:

$$\text{If } a_0 + a_1X_1 + a_2X_2 > 0, \text{ then } c(\boldsymbol{x}) = c_1 \tag{49}$$

$$\text{If } a_0 + a_1X_1 + a_2X_2 \leq 0, \text{ then } c(\boldsymbol{x}) = c_2 \tag{50}$$

The partition line can also be piecewise linear, quadric, and so on. Pattern recognition, fuzzy classification, and ANN are all weighting methods.

In comparison, the decomposition method decomposes the feature space into two areas, as shown in Fig. 10(b). The decomposition methods may look attractive since they are more effective. For example, there are two misclassified samples in Fig. 10(a) and there is none in Fig. 10(b). However, the best decomposition is difficult to find. With an increase in the number of learning samples and features, possible decomposition quickly becomes unmanageable. For example, suppose there are 100 learning samples and 10 features. Then, according to the permutation rule, there will be

$$P_{10}^{100} = 100 \cdot 99 \cdot 98 \cdot \ldots \cdot 90 \approx 10^{20}$$

possible decompositions. It is unlikely to examine all these decompositions to determine the optimal decomposition. As a result, we are forced to search for the suboptimal solutions. Decomposition is usually described by decision rules, which leads to the decision tree method.

Before choosing a classification method (from those described in the following subsections), it is interesting to know that none has been proved to outperform the others for all applications, either mathematically or practically. Therefore, it is best to try several methods and choose the one that performs the best.

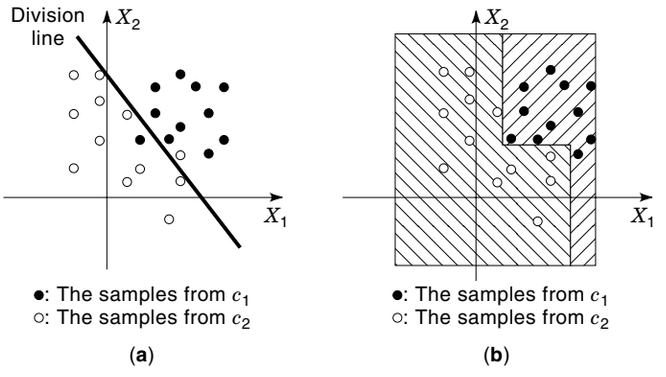

Figure 10. Classification methods. (a) Weighting method, and (b) decomposition method.

Pattern Recognition Method

In general, the pattern recognition methods can be divided into two categories: statistical methods (also called nondeterministic methods) and distribution-free methods (also called deterministic methods).

Statistical pattern recognition methods are based on the Bayes estimation. Assume that the probability density function that a sample $\boldsymbol{x}$ corresponds to class c_j is $f_j(\boldsymbol{x}/\Omega_j)$, where Ω_j represents the parameters of the probability density function and is known or can be found from the training samples. Also assume that p_j is the a priori probability that the sample $\boldsymbol{x}$ corresponds to c_j, and $C_{\alpha j}$ is the cost of misclassification (relates $\boldsymbol{x}$ to c_α when it actually corresponds to c_j). Then the posterior probability density function would be

$$q_j(\boldsymbol{x}) = \sum_{\alpha=1}^{n} p_j C_{\alpha j} f_j(\boldsymbol{x}/\Omega_j) \tag{51}$$

This equation is rather difficult to use; however, if $f_j(\boldsymbol{x}/\Omega_j)$ is Gaussian and the mean vector μ_j and covariance matrix V_j are known, and the costs $C_{\alpha j}$ are all equal, then it can be simplified as follows:

$$q_j(\boldsymbol{x}) = -\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_j)^T V_j^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_j) + \ln\, p_j - \ln\sqrt{V_j} \tag{52}$$

This is relatively easy to calculate. Based on the posterior probability, the Bayes estimation can be found by the following formula:

$$j^* = \arg\max_j \left(q_j(\boldsymbol{x})\right) \tag{53}$$

where arg max implies finding the maximum respect of the argument.

A modified version of the Bayes estimation is the nearest neighbor method. Instead of posterior probability, it uses the following discriminate function:

$$q_\alpha(x) = \frac{f_j p_j}{\sum_{\alpha=1}^{n} f_\alpha P_\alpha} \tag{54}$$

where f_j is called the nearest neighbor. There are a number of ways to define the nearest neighbor. For example, the nearest neighbor defined based on the Mahalanobis distance, $\boldsymbol{x}^T V^{-1}\boldsymbol{x}$, is as follows:

$$f_j = (V_j^{-1/2})^T \boldsymbol{x} \tag{55}$$

In comparison to Eq. (51), the nearest neighbor method is independent of the probability distribution and, hence, is easier to use.

In the learning phase, the cost $C_{\alpha j}$ and a priori probability p_j are first defined (a common assumption is $C_{\alpha j} = 1$ and $p_j = 1/m$, where $\alpha, j = 1, 2, \ldots, m$). Also, based on the available learning samples, we can estimate the mean and the covariance:

$$\boldsymbol{\mu}_j = \frac{1}{N_j}\sum_{k=1}^{N}\delta_{jk}\boldsymbol{x}_k \tag{56}$$

$$V_j^2 = \frac{1}{N_j}\sum_{k=1}^{N}\delta_{jk}(\boldsymbol{x}_k - \boldsymbol{\mu}_j)(\boldsymbol{x}_k - \boldsymbol{\mu}_j)^T$$

where N_j is the number of samples that correspond to the jth process condition, and δ_{jk} is a delta function defined as follows:

$$\delta_{jk} = \begin{cases} 1 & if\ c(\boldsymbol{x}_k) = c_j \\ 0 & if\ c(\boldsymbol{x}_k) \neq c_j \end{cases} \tag{57}$$

The performance of the statistical pattern recognition methods depends on the probability distribution of the samples. It has been shown that if the probability distribution is Gaussian or close to Gaussian, the Bayes estimation is the optimal classification and the nearest neighbor method also performs well. However, if the probability distribution is not close to Gaussian, then the distribution-free methods are preferred.

The distribution-free pattern recognition methods are based on the similarity between a sample $\boldsymbol{x}$ and the patterns. From a geometrical point of view, the signal features span into an m-dimensional space. In this space, each class is characterized by a vector (pattern) $\boldsymbol{p}_j = [p_{1j}\ p_{2j}\ \ldots\ p_{nj}]^T$. On the other hand, the sample $\boldsymbol{x}$ is also a vector in the space. Hence, the similarity between a pattern and a sample can be measured by the distance between them. As shown in Fig. 11, the distance between the sample and pattern $\boldsymbol{p}_1$ is d_1, and the distance between the sample and pattern $\boldsymbol{p}_2$ is d_2. The minimum distance would indicate the resemblance and hence can be used for classification.

There are a number of ways to define patterns and distances. This results in various distribution-free pattern recognition methods. Commonly used methods include Mahalanobis's method, the linear discrimination method, and Fisher's method. In Mahalanobis's method, the patterns are the means of the learning samples (i.e., $\boldsymbol{p}_j = \boldsymbol{\mu}_j, j = 1, 2, \ldots, m$), and the distance is defined as

$$q_j(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{p}_j)^T V_j (\boldsymbol{x} - \boldsymbol{p}_j) \tag{58}$$

The linear discriminate method, also called the K-mean algorithm, uses the same pattern, but the distance is defined as

$$q_j(\boldsymbol{x}_k) = \sum_{i=1}^{m} w_{ij}[x(k,i) - c_{ij}]^2 \tag{59}$$

where w_{ij} and c_{ij} are the weights and centers of the patterns, respectively. They can be determined by minimizing:

$$J = \sum_{k=1}^{N}\sum_{j=1}^{n}\delta_{jk}q_j(\boldsymbol{x}_k) \tag{60}$$

Similarly, Fisher's method uses the same patterns, but the distance is defined as

$$q_j(\boldsymbol{x}) = \beta_j^T \boldsymbol{x} \tag{61}$$

where β_j is determined by maximizing:

$$J = \sum_{j=1}^{n}\beta_j^T V_j \beta_j \tag{62}$$

In the classification phase, the distribution-free methods are similar to the statistical methods. They use the minimum distance:

$$j^* = \arg\min_j \left(q_j(\boldsymbol{x})\right) \tag{63}$$

Pattern recognition methods are typical weighted methods. Their effectiveness depends not only on the discriminate function, but also the distribution of the learning samples as well as the definition of the classes. In practice, many faults are "fuzzy" in nature. For example, in the diagnosis of large rotating machinery, the rotor unbalance may be in many states, from minor to severe. Hence, the diagnosis is often not what the fault is, but to what degree the fault is. This question may be better answered by fuzzy logic.

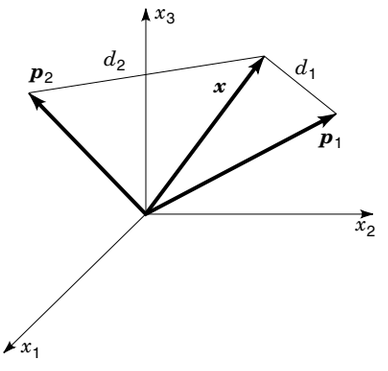

Figure 11. The distribution-free pattern recognition methods.

Fuzzy Logic Method

Details of fuzzy logic and fuzzy systems are discussed in FUZZY LOGIC SYSTEMS. Under the fuzzy concept, uncertain events are described by means of fuzzy degrees (also called relationship functions, possibility functions, or membership functions). Briefly, if A is an uncertain event defined in the universal set U, then A can be described by

$$A = \{x|\mu_A(x)\} \tag{64}$$

where $x \in U$ is the value of A, and $\mu_A(x)$ is the fuzzy degree. The fuzzy degree $\mu_A(x)$ is a monotonous function, $0 \leq \mu_A(x) \leq 1$, while 0 means certainly no and 1 implies certainly yes. The

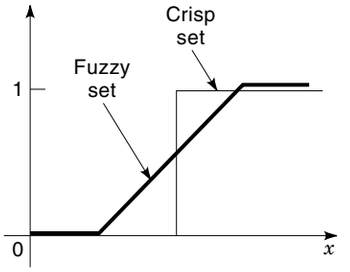

Figure 12. An example of fuzzy membership function.

difference between a fuzzy concept and a certain concept is illustrated in Fig. 12.

An often confused issue is the difference between the fuzzy degree and the probability. The fuzzy degree represents the imprecision of an event (e.g., how similar A is to another event B) while the probability describes the occurrence frequency of A (e.g., how likely it is that A will occur). Based on fuzzy logic, a number of classification methods have been developed. These include the fuzzy C-mean method and the fuzzy linear equation method.

The fuzzy C-mean method was first introduced by Bezdek (8). It uses a cluster center, $\boldsymbol{V} = [v(j, i)]$, and a fuzzy degree, $\boldsymbol{U} = [u(k, j)]$, for classification. In the leaning phase, the cluster center and the fuzzy degree are determined by minimizing:

$$\boldsymbol{J}(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{X}) = \sum_{k=1}^{N} \sum_{j=1}^{n} \sum_{i=1}^{m} u(k, j)^v \|x(k, i) - j, i)\|^v \tag{65}$$

subject to

$$\boldsymbol{M} = \left\{ [u(k, j), v(j, i) \Big/ \sum_{j=1}^{n} u(k, j) = 1, \forall k = 1, 2, \ldots \right\} \tag{66}$$

where v is a positive number that controls the shape of the fuzzy degree (usually $v = 2$ is used), $\|\cdot\|$ represents the norm, and M represents the feasible solution sets. It has been shown (8) that the necessary condition for solving Eq. (66) is

$$u(k, j) = \frac{1}{\sum_{i=1}^{m} \sum_{\alpha=1}^{m} \left(\dfrac{\|x(k, i) - v(j, i)\|}{\|x(k, i) - v(\alpha, i)\|} \right)^{1/v-1}} \tag{67}$$

$$v(j, i) = \frac{\sum_{k=1}^{N} u^v(k, j) x(k, i)}{\sum_{k=1}^{N} u^v(k, j)} \tag{68}$$

Equations (67) and (68) cannot be solved analytically but can be solved by iterations. Once the cluster center is found, the correlation of a new sample, $\boldsymbol{x}$, to the classes can be evaluated based on its fuzzy degrees, $u(\boldsymbol{x}, j), j = 1, 2, \ldots, n$, calculated using Eq. (67). Furthermore, its estimated class is the one that has the maximum fuzzy degree:

$$j^* = \arg\max_j \big(u(\boldsymbol{x}, j)\big) \tag{69}$$

The fuzzy linear equation method is first introduced in Ref. 9. It is assumed that the relationship between the signal features and the classes, as shown in Fig. 8, can be described by a fuzzy linear equation:

$$\boldsymbol{r} = \boldsymbol{Q} \circ \boldsymbol{p} \tag{70}$$

where $\boldsymbol{r}$ represents the fuzzy degree of the signal features, $\boldsymbol{p}$ represents the fuzzy degree of the classes, $\boldsymbol{Q}$ is the fuzzy relationship function, and the symbol "$\circ$" is a fuzzy operator (10). Rewriting Eq. (71) in matrix form,

$$\begin{bmatrix} r_1 \\ r_2 \\ \cdots \\ r_m \end{bmatrix} = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1n} \\ q_{21} & q_{22} & \cdots & q_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ q_{m1} & q_{m2} & \cdots & q_{mn} \end{bmatrix} \circ \begin{bmatrix} p_1 \\ p_2 \\ \cdots \\ p_n \end{bmatrix} \tag{71}$$

For each row, we have

$$r_i = q_{i1} \otimes p_1 \oplus q_{i2} \otimes p_2 \oplus \ldots \oplus q_{in} p_n \tag{72}$$

where $\otimes$ denotes the fuzzy multiplication and $\oplus$ denotes the fuzzy addition. The element q_{ij} is the fuzzy relationship that relates the ith signal feature to the jth class. In the learning phase, the relationship is determined by the occurrence frequency and the strength of support of the learning samples. Let $S_i = \{x(1, i), x(2, i), \ldots, x(N, i)\}$, which is the set that contains the ith signal features of all learning samples, and let

$$x_{i,\max} = \max_i \{x(1, i), x(2, i), \ldots, x(N, i)\} \tag{73}$$

$$x_{i,\min} = \min_i \{x(1, i), x(2, i), \ldots, x(N, i)\} \tag{74}$$

Furthermore, dividing the interval between $x_{i,\max}$ and $x_{i,\min}$ into L evenly distributed subintervals (in practice, $L = N/10 \sim N/15$ is recommended so that there will be enough samples in each interval). Each subinterval, denoted by $v(i, k)$, $k = 1, 2, \ldots, L$, is defined as follows:

$$v(i, k) = [x_{i,\min} + (k - 1)\Delta x_i, x_{i,\min} + k\Delta x_i] \tag{75}$$

where

$$\Delta x = \frac{x_{i,\max} - x_{i,\min}}{L}$$

Then q_{ij} can be represented by a set with L elements:

$$q_{ij} = \{v(i, k) | q(i, j, k), k = 1, 2, \ldots, L\} \tag{76}$$

where the fuzzy degree, $q(i, j, k)$, is determined by the occurrence frequency and the strength of support of the learning samples as defined in the following equation:

$$q(i, j, k) = \alpha \frac{C_{ijk}}{C_{ik}} + (1 - \alpha) \frac{C_{ijk}}{C_{ij}} \tag{77}$$

where C_{ijk} is the number of training samples that correspond to jth class and located inside the kth subinterval, C_{ik} is the number of samples that are located inside the kth subinterval, C_{ij} is the number of samples in S_i that correspond to the jth process condition, and $0 \le \alpha \le 1$ is a constant.

Once the relationship function, $\boldsymbol{Q}$, is found, and a new sample, $\boldsymbol{x}$, is given, the classification is done in two steps. First, since each element, q_{ij}, of the fuzzy relationship function is a set, it is necessary to determine which element of the set should be used. Such an element is called the value of the fuzzy relationship and is denoted by $\boldsymbol{Q}^v$. It is determined based on the sample: Suppose the value of the ith feature of the sample is located inside the kth subinterval, $v(i, k)$; then

$$q_{ij}^v = q(i, j, k) \tag{78}$$

By so doing, the fuzzy relationship $\boldsymbol{Q}$ is reduced to a $m \times n$ matrix $\boldsymbol{Q}^v$. The second step is to solve the linear fuzzy equation. A commonly used solution is the max-min solution defined in the following equation (10):

$$p_j = \max_i \min\{q_{ij}^v, r_i\} \tag{79}$$

An often better performed solution is the one proposed in Ref. 9:

$$p_j = \sum_{i=1}^{m} \min\{q_{ij}^v, r_i\} \tag{80}$$

Based on the preceding solutions, the corresponding class of the new sample is the one that has the maximum fuzzy degree; that is,

$$j^* = \arg\max_j \{p_j\} \tag{81}$$

Artificial Neural Network

Since its rediscovery in the 1980s, the ANN has quickly become one of the most commonly used methods for fault diagnosis. The reader can find a detailed discussion on ANN in NEURAL NET ARCHITECTURE. In short, as shown in Fig. 13, from a mathematical point of view, an ANN can be considered as a nonlinear mapping function, which maps a set of signal features $\boldsymbol{x}$ (input of the ANN) to a pattern $\boldsymbol{z}$ (output of the ANN). From the figure, we also see that a typical feedforward ANN consists of an input layer, a hidden layer, and an output layer. The nodes in the hidden layer can be described by

$$y_k = F\left(\sum_{i=1}^{n} x_i w_{ik} + \theta_k\right) \tag{82}$$

where $k = 1, 2, \ldots, h$ is used to index the nodes in the hidden layer (h is the number of nodes in the hidden layer), x_i is the inputs, w_{ik} is the weights, θ_k is the thresholds, and $F(\cdot)$ is a nonlinear function. $F(\cdot)$ may be in various forms and one of them is defined as follows:

$$F(t) = \frac{1}{1 + e^{-t}} \tag{83}$$

Similarly, the output nodes of the network can be described by

$$z_k = F\left(\sum_{i=1}^{h} y_i g_{ik} + \rho_k\right) \tag{84}$$

where $k = 1, 2, \ldots, m$ is used to index the output nodes of the ANN, g_{ik} is the weights, and ρ_k is the thresholds.

In the learning phase, building an ANN involves (1) designing the architecture of the ANN (namely, select number of layers and number of nodes in each layer), (2) assigning desirable or target outputs of the ANN, denoted by $\boldsymbol{d} = (d_1, d_2, \ldots, d_m)$, and (3) applying a training algorithm to find the weights and the thresholds of the ANN, $\{w_{ik}, \theta_k, g_{ik}, \text{ and } \rho_k\}$ that minimize the error:

$$E = \sum_{j=1}^{N} (d_j - z_j)^2 \tag{85}$$

where $\boldsymbol{z}_j$ is the actual output corresponding to sample $\boldsymbol{x}_j$.

Regarding the structure design, it has been agreed that one hidden layer is usually sufficient. It is also known that the number of nodes in the hidden layer must be sufficient. However, if too many nodes are used, the network may capture and memorize insignificant patterns or noises in the training samples. As a result, its ability to reason is reduced. The optimal number of nodes can be found based on the fact that the best ANN is the one most similar to the training samples. The similarity can be defined in a number of different ways, and one of them is as follows:

$$S = \sum_{i=1}^{N} \sum_{j=1}^{N} a_{ij} \ln \frac{a_{ij}}{b_{ij}} + b_{ij} \ln \frac{b_{ij}}{a_{ij}} + a_{ij} \ln \frac{a_{ij}}{c_{ij}} + c_{ij} \ln \frac{c_{ij}}{a_{ij}} \tag{86}$$

where $a_{ij} = \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$, $b_{ij} = \|\boldsymbol{y}_i - \boldsymbol{y}_j\|$, and $c_{ij} = \|\boldsymbol{z}_i - \boldsymbol{z}_j\|$. Note that the similarity is a function of h (the number of nodes in the hidden layer); that is, $S = S(h)$. Accordingly, the optimal number of nodes can be found by minimizing the total similarity:

$$h^* = \arg\min\{S(h)\} \tag{87}$$

where h^* is the optimal number of nodes in the hidden layer.

There are two ways of assigning target outputs. The first one is the so-called 0–1 assignment. It assigns a one to the

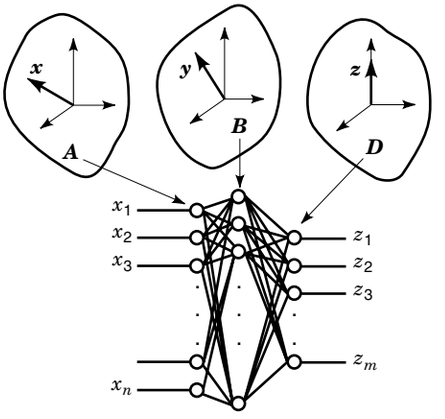

Figure 13. Nonlinear mapping in an ANN.

output corresponding to the class of the sample and zero to the others. For example, if $c(\boldsymbol{x}) = c_1$, then $d(\boldsymbol{x}) = [1, 0, \ldots, 0]$. The other one is based on the similarity. That is, the target outputs shall be similar to the patterns of the training samples. Obviously, the most similar assignment is the training samples themselves. However, this assignment will force the ANN to follow a large number of unorganized patterns so that the ANN becomes very complicated and, more important, loses its ability to reason. The second most similar assignment is the mean of each class of the training samples. That is,

$$d_i = \boldsymbol{x}_i = \frac{1}{NC_i} \sum_{j=1}^{N} \delta_{if} \boldsymbol{x}_k \tag{88}$$

where NC_i is the number of training samples that correspond to the jth class, c_i, $i = 1, 2, \ldots, m$.

When the structure is designed and the target outputs are assigned, we can then train the ANN. There are a number of ANN training methods. Unarguably, the most commonly used method is the back propagation (BP) algorithm. It is a set of iteration equations used to determine the coefficients w_{ik}, θ_k, g_{ik}, and ρ_k that minimize the estimation error defined in Eq. (84), and these equations can be found in ART NEURAL NETS (11). Whereas the ANN is trained and a new sample is presented, the corresponding class of the new sample can be estimated by calculating the output of the ANN and comparing the output to the target output. This is similar to the pattern recognition method and the fuzzy logic method discussed in the previous sections.

Decision Trees

The classification methods described previously are all weighting methods, in which decisions are made by weighting the signal features. For example, the pattern recognition methods use linear (K-mean algorithm) or quadratic (Fisher's algorithm) weighting functions; the fuzzy logic methods use fuzzy degrees, and the neural networks use nonlinear mapping functions. As point out earlier, classifications can also be done by decomposing the signal features.

The most effective way of decomposition is the use of decision trees. The decision tree can be built by partitioning the training samples. For simplicity, let us consider how to build a binary tree. It starts from the root of the tree, at which a signal feature and a threshold are selected to partition the training samples, $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$, into two sets: $\boldsymbol{X} = \boldsymbol{X}_1 + \boldsymbol{X}_2$. Each set contains mutually exclusive patterns. Then, two nodes are built following the root of the tree. Then, at Node 1, the training sample set $\boldsymbol{X}_1$ is further partitioned into two subsets (i.e., $\boldsymbol{X}_1 = \boldsymbol{X}_{11} + \boldsymbol{X}_{12}$); and at Node 2, the training sample set $\boldsymbol{X}_2$ is partitioned into two subsets (i.e., $\boldsymbol{X}_2 = \boldsymbol{X}_{21} + \boldsymbol{X}_{22}$). Such a partition process continues until all the training samples are grouped according to their corresponding classes. This process is illustrated in Fig. 14.

There are many different ways to partition the training samples. The optimal partition can be obtained by finding all the possible partitions and choosing the one that minimizes a given objective function. However, as mentioned earlier, this leads to a so-called NP complete problem just like the traveling salesperson problem. It requires an exponential computation load and cannot be solved when the number of samples

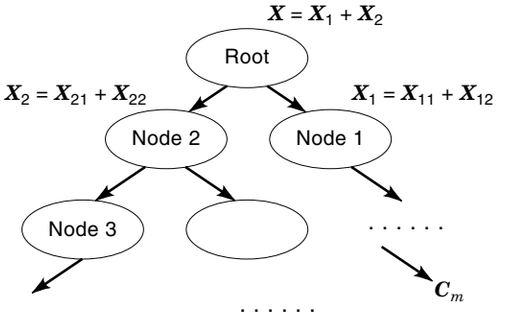

Figure 14. Building a decision tree.

N is large, regardless of how powerful a computer may be. Hence, the best we can do is to find a suboptimal partition. A number of methods have been developed to find suboptimal partitions, and one of the effective ones is an algorithm called ID3 (Iterating Dichotomizer Three).

Algorithm ID3 was first introduced by Quinlan (12). It uses the minimum entropy gain to direct the search of the partition. Suppose at a node of a tree, there are $S (S \leq N)$ samples, $\boldsymbol{X}_S = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_S,\}$, to be partitioned. The partition is associated to the entropy determined by the distribution of the samples. Let NC_k, $k = 1, 2, \ldots, m$, be the number of samples in $\boldsymbol{X}_S$ that correspond to class c_k, and

$$P_{SC_k} = \frac{NC_k}{S} \tag{89}$$

Then the entropy, denoted by $I(\boldsymbol{X}_S)$, is defined as follows:

$$I(\boldsymbol{X}_S) = \sum_{k=1}^{m} P_{SC_k} \log_2(P_{SC_k}) \tag{90}$$

The partition is to decompose the training samples into two subsets: $\boldsymbol{X}_S = \boldsymbol{X}_{S1} + \boldsymbol{X}_{S2}$, where $\boldsymbol{X}_{S1}$ and $\boldsymbol{X}_{S2}$ have S_1 and S_2 samples, respectively; and $S = S_1 + S_2$. Suppose, furthermore, that the jth signal feature, X_j, is used as the pivot of the partition. Then the entropy of the partition is

$$E(\boldsymbol{X}_S, X_j) = \frac{S_1}{S} I(\boldsymbol{X}_{S1}) + \frac{S_2}{S} I(\boldsymbol{X}_{S2}) \tag{91}$$

and the entropy gain of the partition is

$$G(\boldsymbol{X}_S, X_j) = I(\boldsymbol{X}_S) - E(\boldsymbol{X}_S, X_j) \tag{92}$$

Note that for each signal feature, there may be a large number of possible partitions. However, only a few could result in a small entropy gains and, hence, provide desirable classification. These are the partitions that make $\boldsymbol{X}_{S1}$ and $\boldsymbol{X}_{S2}$ contain mutually exclusive patterns. For example, $\boldsymbol{X}_{S1}$ contains only the samples from a certain class: c_1 or c_2, . . ., or from two classes: $c_1 + c_2$, $c_1 + c_3$, and so on; and $\boldsymbol{X}_{S2}$ contains the complement. In particular, there are only m partitions that make $\boldsymbol{X}_{S1}$ contain the samples from one class, which can be found easily by sorting the data. Note that there may not exist a signal feature that is capable of completely separating one class from the others. In this case, the partitions that make $\boldsymbol{X}_{S1}$ contain most samples from a certain class will be

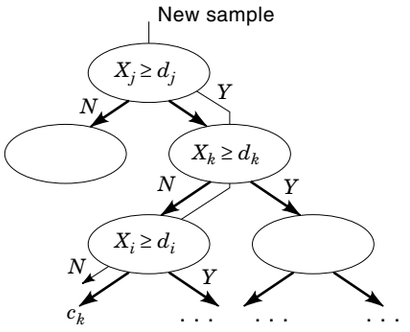

Figure 15. Decision tree method.

used. When a partition is determined, the threshold of the partition is the arithmetic mean of the two closest points in $\boldsymbol{X}_{S1}$ and $\boldsymbol{X}_{S2}$.

In the ID3 algorithm, the aforementioned partitioning process starts at the root of the decision tree, where all the samples are to be partitioned. It examines all the partitions of every signal feature and selects the partition that has minimum entropy gain (if there is more than one partition having the same minimum entropy gain, then the one that has largest difference between the two sets $\boldsymbol{X}_{S1}$ and $\boldsymbol{X}_{S2}$ shall be selected). Then the partitioned samples are partitioned again. The process ends when all partitioned samples are properly classified (i.e., each partitioned subset contains only the samples from the same class), resulting a binary decision tree.

When a decision tree is built, the classification can be done by searching through the tree. Since most decision trees are rather simple, a binary search is usually sufficient. As shown in Fig. 15, given a new sample, $\boldsymbol{x} = \{x_1, x_2, \ldots, x_n\}$, the search starts at the root: If $x_j > d_j$, then the search is directed to the right. Next, assuming that $x_k < d_k$, the search is directed to left, and so on. Finally, the corresponding class of the new sample, c_k, is found, as indicated by a leaf of the tree.

The decision tree method often works very well. According to the simulation study by Quinlan (12), if the training samples cover 50% of the problem space, the success rate is 75%. If the training samples cover 85% of the problem space, then the success rate may reach as high as 95%. Another interesting feature of the decision tree method is that it may not use all the signal features. In fact, the unused signal features are the less effective features and, hence, can be disregarded.

However, the decision tree method learns only from the training samples and cannot capture any faults that are not in the training samples, no matter how simple they may be. For example, sensor malfunction is a common problem in fault diagnosis and can be easily captured since it is always associated with either no signal or a saturated signal. Nevertheless, this cannot be recognized unless extensive training samples are provided during the training of the decision tree. To solve this problem and hence add flexibility to the fault diagnosis, we can use the expert systems method.

Expert Systems

Expert systems are discussed in the article EXPERT SYSTEMS. In general, expert systems consist of three basic components: an interface, an inference engine, and a knowledge base. The interface is the window of communication between the user and the computer. The inference engine is used to manipulate the knowledge. Regardless the applications, the basic functions of the interface and inference engine are the same, and hence expert system shells are developed. As a result, the main effort of using expert systems for fault diagnosis is to develop the knowledge base. The knowledge base may be in various forms; the most commonly used form is the rule base, where the knowledge is represented in terms of rules: "If . . ., then". Although there may be cases in which multiple rules may apply and different applicable rules lead to contrary results, the expert system shell usually manages to deliver good results. Therefore, the main task in developing the knowledge base is to develop the rules. This is called knowledge acquisition.

There are several knowledge acquisition methods: (1) machine learning, (2) system modeling and simulation, and (3) domain experts consultation. The decision tree method described previously is a typical example of machine learning. It is called learning from samples. The samples may be obtained from historical records (operation records and maintenance records of the system and other similar or related systems). Many systems, such as power generation stations, large turbine machinery sets, and automobile assembly lines, kept extensive historical records ranging from quality control charts to maintenance service records. These records and the records of other similar or related systems are important knowledge sources for fault diagnosis. From a theoretical point of view, historical records and samples represent specific instances of faults. Learning from samples is a generalization process that constructs diagnosis rules from these samples. Because of the incompleteness of the learning samples (in practice, there are always new samples that are different from the learning samples), the learned decision rules are often partial. The accuracy of the learned rules can be evaluated by error, which includes bias and variation. For example, upon obtaining a new sample, we can calculate the new mean and new variance of a class. If the new mean is almost the same as the old mean, then we say that the estimation is unbiased. If, furthermore, the new variance does not change, then we say the variation is small. Also, there may be a large number of samples from a same class (e.g., the normal class). This is called redundant information. Using redundant information can improve classification accuracy as well.

Computer modeling and simulation allow us to look into the inside a system under various working conditions and hence are excellent tools of knowledge acquisition. Depending on the applications, various system models can be used, such as dynamic models or finite element (or finite difference) models. The dynamic models are often used for fault diagnosis. This is due to the fact that most engineering systems are dynamic systems and the dynamic features, such as natural frequencies and damping ratios, are effective features for fault diagnosis. The use of the finite element model (FEM) is based on the consideration that engineering systems are, in fact, distributed-parameter systems. Hence, it is important to examine not only the behavior of the system as a whole but also the behavior of the system at particular areas. Models are simplified representations of systems, and the accuracy of a model depends greatly on the formation of the model as well as on the key parameters (such as system parameters, material constants, and friction coefficients). When checking a handbook, it is not unusual to find that these parameters vary over a wide range. As a result, the system model may

behave differently. To improve the accuracy of the model, we can use the sensor signals to fine-tune the key parameters. Based on computer models and simulation, various system faults can be simulated. Since the simulation costs no more than the computation cost, it is arguably the cheapest method of knowledge acquisition.

Domain experts are those people who know how. They may include the people who research, design, manufacture, operate, and maintain the system and/or similar and related systems. They know the system from different aspects and often possess the knowledge in-substitutable. Acquiring knowledge from the domain experts involves interviewing them and organizing knowledge. Interviewing the domain experts should be subjective and specific. *Subjective* means not leading the questions and adding opinions. *Specific* means focusing on the issue. The following lists are a selection of questions.

For system operators and maintenance workers:

1. When you see [a system failure], what else you also see [or hear]?
2. When you see [a system failure], what do you do?
3. Last time you saw [a system failure], what was the difference/similarity to this one?

For system designers and manufacturers:

1. When [a system failure] occurs, what do you think could also happen?
2. When [a system failure] occurs, what do you think should be done?

Often the knowledge acquired from domain experts is vague, incomplete, and controversial. Therefore, knowledge organization is necessary. Knowledge can be organized by several different forms, such as rules and events with the use of belief functions and fuzzy degrees. The reader is referred to EXPERT SYSTEMS for details.

Finally, fault diagnosis is a task involving the entire system life cycle. Whenever new knowledge and/or information become available, we should update or upgrade the diagnosis rules so that the diagnosis expert systems can self-improve throughout their course of application.

Remarks on Using Signal Classification Methods

In summary, the following rules are recommended for choosing signal classification methods for fault diagnosis:

1. Since most of the faults are fuzzy in nature (in terms of their identity, severity, and correlation to other faults), we should use fuzzy logic methods for signal classification.
2. To diagnose complicated systems with many different faults (more than six) and signal features (more than eight), we should use the decision tree method because it can effectively decompose a complicated problem into several smaller problems.
3. Signal classification requires a learning process. Most signal classification methods described in this article learn from samples. To accommodate other information and knowledge, we should use expert systems.

The final and perhaps the most important recommendation for an engineer who needs to conduct fault diagnosis is to understand the system. After all, the system faults occur within the system. Without a good understanding of the system, it would be difficult to understand what are the faults and what may cause the faults. Consequently, it would be pointless to use the fault diagnosis tools described in this article or any other tools. Fortunately, for most engineering systems, there is usually abundant literature including product user manuals, trade magazine articles, case study reports, monographs and academic journals, and conference papers. These provide all kinds of information necessary for fault diagnosis. For example, for fault diagnosis of large rotating machinery, one may refer to Refs. 14 and 15. In particular, Ref. 14 presents some 54 practical cases with detailed fault patterns and correction methods. Following these works, we minimize the fault diagnosis errors and, hence, optimize the operations of the engineering systems.

BIBLIOGRAPHY

1. R. V. Hogg and A. T. Craig, *Introduction to Mathematical Statistics,* 4th ed., New York: Macmillan, 1978.
2. T. Grandke, Interpolation algorithm for discrete Fourier transforms of weighted signals, *IEEE Trans. Image Process.,* **IM-32**: 350–355, 1983.
3. H. Choi and W. J. Williams, Improved time-frequency representation of multicomponent signals using exponential kernels, *IEEE Trans. Acoust. Speech Signal Process.,* **37**: 862–871, 1989.
4. S. G. Mallat, Multifrequency channel decomposition of images and wavelet models, *IEEE Trans. Acoust. Speech Signal Process.,* **37**: 2091–2110, 1989.
5. Y. Wu and R. Du, Feature extraction and assessment using wavelet packets for tool condition monitoring, *Mech. Syst. Signal Process.,* **10** (1): 29–53, 1996.
6. R. Du, Y. D. Chen, and Y. B. Chen, Four dimensional holospectrum—A new method for analyzing force distributions, *Trans. ASME, J. Manufacturing Eng. Sci.,* **119** (1): 95–104, 1996.
7. C. L. Nikias and M. R. Raghuveer, Bispectrum estimation: A digital signal processing framework, *Proc. IEEE,* **75**: 869–891, 1987.
8. J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithm,* New York: Plenum Press, 1981.
9. R. Du, M. A. Elbestawi, and S. Li, Tool condition monitoring in turning using fuzzy set theory, *Int. J. Mach. Tools Manuf.,* **32** (6): 781–796, 1992.
10. J. G. Klir and A. T. Folger, *Fuzzy Sets, Uncertainty, and Information,* Englewood Cliffs, NJ: Prentice Hall, 1988.
11. D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition,* Vol. 1 and 2, Boston: MIT Press, 1988.
12. J. R. Quinlan, Induction of decision trees, *Machine Learning,* **1**: 81–106, 1986.
13. L. A. Zadeh, Outline of a new approach to the analysis of complex systems and decision processes, *IEEE Trans. Syst. Man Cybern.,* **SMC-3**: 28, 1973.
14. R. C. Eisenmann, Sr. and R. C. Eisenmann, Jr., *Machinery Malfunction Diagnosis and Correction,* Hewlett-Packard Professional Books, Englewood Cliffs, NJ: Prentice-Hall PTR, 1997.
15. HP Application Note 243, *Fundamentals of Signal Analysis,* and HP Application Note 243-1, *Effective Machinery Measurements Using Dynamic Signal Analyzers,* Palo Alto, CA: Hewlett-Packard, 1994.

RUXU DU
University of Windsor

FAULT DIAGNOSIS. See FAULT LOCATION.