

ONLINE DIAGNOSTICS OF ROTATING MACHINERY

by

Kim Alan Lovejoy

President

Lovejoy Controls Corporation

Waukesha, Wisconsin



Kim Alan Lovejoy is currently President of Lovejoy Controls Corporation of Waukesha, Wisconsin. He obtained B.S.M.E./B.S.E.E. degrees from the University of Wisconsin.

His current duties include managing development and implementation of custom control and instrumentation systems for clients worldwide. Mr. Lovejoy's areas of specialization include steam turbine speed control systems and vibration monitoring networks.

Previous related employment was with the Westinghouse Electric Corporation Power Generation Service Division, as a Controls and Vibration Field Specialist from 1974 to 1978.

Currently, Mr. Lovejoy is a member of the Vibration Institute and the Instrument Society of America.

ABSTRACT

Has the industrial standardization of the IBM PC/AT as a building block for dedicated instrument computer systems brought costs down to the point where sophisticated monitoring systems are not only possible, but make good economic sense?

Are today's plant engineering departments more-and-more knowledgeable of both rotating machinery phenomena and computer applications to the point of surpassing most instrument vendors?

Has the vibration monitor manufacturing industry as a whole been guilty of over selling the application of FFT (Fast Fourier Transform) analyzers in machinery diagnostics?

Can an end user design and implement a better machinery diagnostic system than can be bought turn-key, and at far less cost?

Are most so-called "Predictive" or "Preventative Maintenance" programs really only "Failure Detection" and "Failure Management" programs?

Material presented herein shall give reason to answer "YES" to all the above.

INTRODUCTION

The first practical machinery diagnostics were performed by equipment operators in the first decade of this century, up to the mid 1960s (and to present at low-budget facilities). The methods employed relied upon an operator's sensory inputs of feel, sight, and sound with the occasional employment of a noise conductive dowel. Before finding humor with such an arrangement, it must be pointed out that this system can be quite effective and surprisingly accurate.

Human sense of hearing has a typical frequency bandwidth of 25 Hz to 18,000 Hz, with exact ranges varying among individuals. Human ability to "feel" flooring vibration or pulsations trans-

mitted through a wood dowel to the ear bone structure lower this perceived minimum to several hertz. Human hearing by itself has remarkable analyzer ability. The basilar membrane of the inner ear responds to frequency components of complex sounds by stimulating auditory nerve membranes along its spiral length. The ear's ability to differentiate sound patterns rivals the best acoustical sensors. Oldtime equipment operators, unlike their control room housed modern counterparts, sat at desks or consoles within feet of operating equipment. After several months of absorbing equipment sounds and vibrations, these operators held in memory cumulative background noise/vibration patterns which represented a "normal" machine condition. Any slight deviation in the vibration or noise pattern, whether intensity (amplitude) or pitch (frequency) would alert them of a potential problem. The operators did not, however, just sit all day and listen for pattern changes, as other duties commanded their immediate attention. These duties included checking gauges, feeling bearing drains, recording log information, changing loads, and also fulfilling basic human requirements of eating, drinking, and communicating with others. Noise/vibration pattern change recognition was a *background function*. The operator's attention would jerk to a change in pattern when, say, a clattering noise suddenly rose from a turbine. The steps in "analysis" of the old-time operator are basic to any machinery diagnostic system.

The first reaction of the operator was to scan his memory for a history of a similar noise/vibration patterns and the related cause of the disturbance. If the intensity of the clatter demanded more than a curious examination, an immediate decision was also made whether or not to trip the machine and exit the scene. Assuming in the example, that the clatter was mild, and that the operator had no previous experience to identify it, he would then proceed to the machine (with wood dowel to, *gather more data*).

Upon approaching the local source of the clatter, the operator would employ his primary senses, to deduce either the apparent cause, or the need for trained assistance. If the cause was discernable, say, turbine lagging slapping a case, due to a loose bolt, the bolt was tightened, and the *pattern cause* was memorized. Perhaps not after one, but certainly after several such incidents, the operator's memory would hold a "loose lagging" pattern recognition diagnostic, which would continuously be functioning in a *background* mode to other mental duties. This cycle of events is illustrated in Figure 1.

Another point may belabor the example, but is needed to establish a fundamental difference between an "online diagnostic" system which is dealt with herein, and an "expert system," which is quite different. The operator in the loose lagging example neither knew, nor was required to know the exact physics of the noise pattern generation. Similarly, the online diagnostic system relies on pattern changes from sensor inputs, and does not necessarily need "expert" knowledge of the rotordynamics at hand to properly diagnose a common problem. By defining the scope of an online diagnostic system to identify *common* machinery problems, rather than *extraordinary* problems, the services of the friendly "expert" so necessary in expert system design are not required. This is not to say that knowledge of

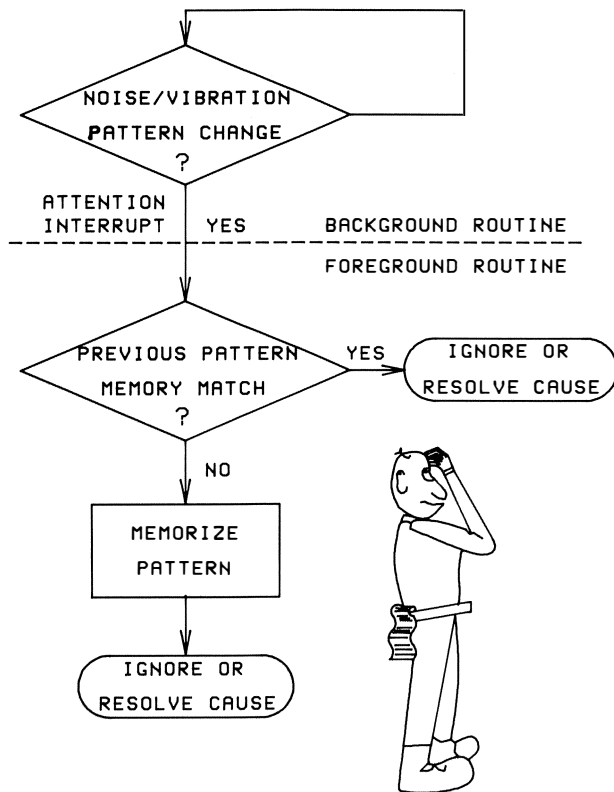


Figure 1. Oldtime Operator Online Diagnostic Cycle.

rotordynamics is unnecessary; in fact, all possible "expert" help will be used to establish recognizable problems. However, once these patterns are established, knowledge of the transfer functions between problem and sensor pattern is strictly academic.

ANALYSIS INSTRUMENTS, AND "PREDICTIVE MAINTENANCE" PROGRAMS

Probably the first vibration analysis instrument, if one ignores the reed tachometer, was the vibrometer. The vibrometer was a levered stylus which was mechanically coupled to a shaft rider. Shaft rider motion was amplified via the lever to the point of the stylus, which scribed an acetate tape strip drawn past the stylus by a windup spring mechanism. The result was an early version analog signal recorder. With this type of instrument, and little else, basic knowledge of the manifestation of a variety of machinery phenomena as component frequency elements were established by pioneering engineers.

With the commercial availability of semiconductor circuits in the late 1960s and early 1970s, a trend began to develop spawned by instrument manufacturers to use portable equipment for periodic recording of vibration parameters. The parameters were then examined by a trained technician or engineer for problem pattern recognition. Dozens of instrument manufacturers have presented this approach in mildly varying form as "predictive" or "preventative" maintenance programs to date. Sales literature for the portable equipment enthusiastically proclaims that one can "diagnose failures before they occur" and "avoid costly shutdowns by predicting time-to-failure." The sales departments of the manufacturers could hardly contain themselves in explaining all the benefits of their programs, using their instrument.

Examining the substance of these ads, one finds the manufacturers are now claiming predictive powers for what has histori-

cally been vibograph problem diagnostics. Charts and tables published since the 1930s have pointed out that a $2 \times$ frequency component (twice running speed frequency) commonly indicates misalignment of shafts. A component of $\frac{1}{2} \times$ or slightly less, indicates natural frequency excitation by rubs in many rotors. A very high frequency present on antifriction bearing units has been known for decades to indicate bearing damage. There have, of course, been some new diagnostics based upon elaborate studies with FFT analyzers, notably in areas of rotor crack and torsional vibration detection, but as a whole, the performance has not equaled the hype.

Why have these "walk by" programs with FFTs been disappointing? Two very basic reasons are suggested. First, a walk by program by definition does not provide continuous monitoring, an essential element in clearly defining problem patterns. When a machine is visited only once a day or week for data acquisition with temporary instrumentation, a damaging failure often occurs between samplings. The earliest possible detection of loss of rotating mass or antifriction bearing failure is essential in minimizing the extent of damages. It is not implied that any "prediction" is to be made here—the failure has already occurred (leave the predicting to instrument manufacturer advertisement writers). The value of early detection is found in having to replace bearings in lieu of an entire rotor, or a shroud in lieu of an entire blade row. Claiming "prediction" in such cases is akin to labeling the low oil pressure light on an auto as "predictive" in that if you do not heed its warning, you shall in time ruin your engine.

The second reason for lack of success lies in the very use of an FFT as a machinery vibration analysis device. The FFT is far from a perfect frequency resolver, and is by no means the last word in analyzing machinery vibration data. The major flaw of the FFT is traceable to its Fourier transform parent. The Fourier transform "resolves" a raw signal by summing sineoidal functions until the raw function is approximated within a given tolerance or acceptable error magnitude. In other words, the Fourier transform converts a time-domain signal into a frequency-domain signal. The transform is mathematically equivalent to equating a function to a series of a given number of terms of the summation of sineoidal components,

$$f(t) = \sum_{n=1}^{\text{no. terms}} C_n \cos(n\omega t - \theta) \quad (1)$$

If the raw signal is indeed composed of only sineoidal elements, the Fourier transform will be precisely accurate when the number of terms equals the number of components of the raw signal, and all would be fine. As is often the case, however, a fly swims in this ointment. While real world machinery vibration signals are *generally* sineoidal in composition, they are seldom void of several discontinuities, and never, in practice, the pure addition of simple harmonic motions. Discontinuities of signal result from both physical and electrical disturbances. The principle offenders are listed in Table 1. The effect of these discontinuities in a signal analyzed by Fourier method is the generation of false sidebands, or harmonic components which exist not in the signal analyzed, but as a result of the failure of the Fourier transform to accurately model nonsineoidal variations. In short, the Fourier transform attempts to resolve the discontinuities by *creating* whole-fractional and integer-multiple false harmonics of the principle frequency. Termed the *Gibb's Phenomena*, this effect is best illustrated by the FFT's spectral presentation of the one-sixth duty cycle rectangular wave function $f(2)$, shown in Figure 2. Such impulse functions result from the phenomena shown in Table 1.

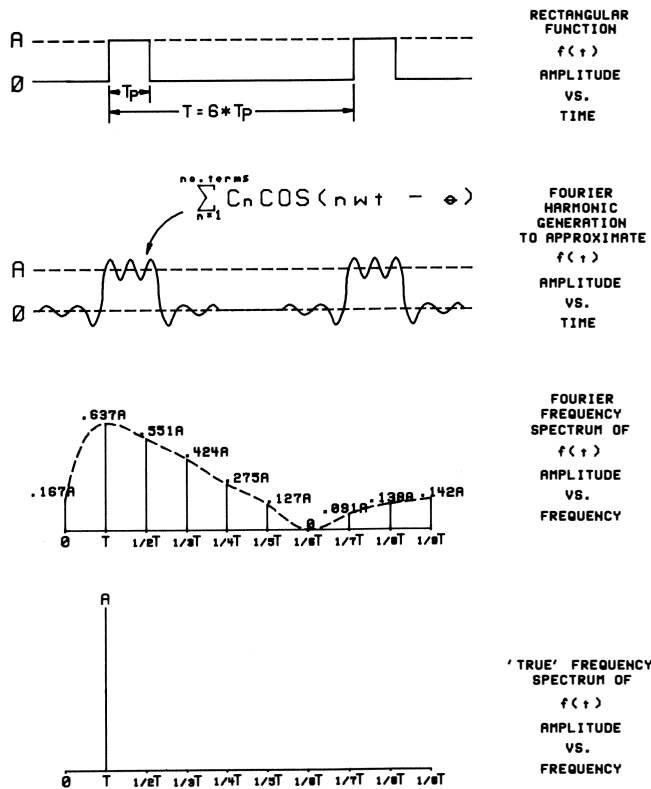


Figure 2. FFT Analysis of a Rectangular Wave.

Table 1. Sources of Signal Discontinuities.

SOURCES OF DISCONTINUITIES
IN
ANALOG VIBRATION SIGNALS

- EXCITATION POWER SUPPLY RIPPLE
- DENTED SHAFTS
- PROXIMITY PROBE ELECTRICAL RUN-OUT
- RF NOISE IN ENVIRONMENT
- POOR ANALOG CABLE SHIELDING
- THERMAL NOISE OF AMPLIFIERS
- BEARING-IMPACTING
- ROTOR-IMPACTING
- ANALOG "RUNOUT-SUBTRACTORS"

This shortcoming is far from unknown to instrument designers. The fix employed is usually an attempt at filtering (flat-top, Hanning, Kaiser-Bessel, etc.) the discontinuities from the raw signal before Fourier analysis. None of the filters used are perfect, and each distorts both the resolved amplitude and the frequency domain of the signal. Very high quality lab grade FFT instruments with dedicated circuitry per channel and sophisticated background "white noise filtering" minimize these errors to insignificance, but the devices marketed as machinery vibration analyzers are far from this quality and price. The lab grade instruments tend to be fragile, do not survive the harsh life of walkby monitoring well, and often require elaborate calibration techniques and equipment.

A telltale of a harmonic-confused FFT analyzer is the generation of false sidebands of six, eight, ten or more times the running speed frequency of the machine under study. Curiously, this type of "spectrum" is often featured on sales literature for the instruments (perhaps a mundane single or two-component plot would appear less impressive). If one reviews the sensor type and mounting to verify the signal is true rotor motion, an interesting paradox is revealed with respect to the *energy* of the vibration. While running speed components represent a static rotor deflection, higher harmonics dictate oscillatory rotor motion at the particular frequency. If the energy necessary to bend the rotor at each of these frequencies to the indicated amplitudes of the "spectrum" are summed, one often exceeds the nameplate rating of the prime mover!

Returning attention to the task of machinery problem diagnostics, almost all frequency patterns for common machinery problems occur at whole-fraction or integer-multiple harmonics, or just where the FFT analyzers produce false data. Attempting to discern subtle trends in harmonic components by algorithm with such instruments requires a parallel faith in astrology.

DESIGNING AN ONLINE
MACHINERY DIAGNOSTIC SYSTEM

The first step in designing an online diagnostic system is defining the system goals. Restraint should be practiced in initially defining goals to avoid over complication and a project which grows into a monster. Because of specifying computer equipment and automated diagnostics, the urge will develop to add a multitude of spinoff features. Interested parties will request computer database ties to parts inventories, offsite modems, remote data terminals throughout the plant, and sundry other features, which will distract the designer from the basic system function.

Realistic goals address the desire to detect, during operation of a machine, problems experienced in the past or likely to be experienced. The system designed will be a concise computer network which mimicks the diagnostic process of the oldtime operator. For a working example, a small steam turbine will be used. The particular turbine has had several historical problems which would be highly advantageous to detect as they occur. These are well known phenomena, the dynamics of which have been clearly defined, if not an obvious early detection. The example phenomena are:

- loss of rotating mass (blades, shroud),
- internal rub,
- bearing wipe,
- misalignment.

The system goals will be to detect these phenomena at the earliest possible moment so as to minimize damage.

The system to be designed is *diagnostic*, not *supervisory*. The turbine already has a functioning *supervisory* system, consisting of vibration, thrust wear, bearing oil pressure and vacuum

monitoring with alarm, and in some cases, trip connections. The difference between the two is worthy of elaboration.

A supervisory system monitors simple variables and compares levels with alarm setpoints either in real time (analog systems) or quite frequently (digital/multiplexed). The purpose of the supervisory systems is to alert operators and/or shut down the machine if basic parameters exceed acceptable operating levels. Other than simple setpoint comparisons, no analyzing is performed. An online diagnostic system *may* assume duties of a supervisory system (if it must), but this will significantly impact the system design by requiring parameter level reviews at short time intervals as a foreground function. Such a system involves techniques beyond the scope of this study. Unburdened by supervisory chores, the online diagnostic system design is much easier.

Although the supervisory and online diagnostic systems are independent in function, they very well may share common sensors and pickups. If the particular supervisory sensor is properly located for phenomena detection data, its output may be shared by both systems. In the example, as in most cases, some new sensors will be required. Supervisory systems do not typically phase reference vibration signals, which is an absolute necessity for online diagnostics, so a once-per-revolution keyphase proximity switch will be added. The supervisory vibration pickups will be utilized. Additional forward-shoe thrust bearing metal and journal bearing metal thermocouples will also be added, since it is not recommended to "share outputs" of very low level signals such as thermocouples. The additional sensors do not represent an appreciable cost or installation difficulty. The sensor layout is outlined in Figure 3.

Today's plant instrumentation engineer is blessed with building block hardware of low cost, versatility, and ease of application. The main ingredient of the example system will be the

workhorse "industrial PC," the industrial hardened and improved versions of the Intel 80286 microcomputer-based personal computer family. In addition to the base computers, analog-to-digital input boards, network communications boards, and a host of interface hardware is now available at reasonable cost from a number of manufacturers.

Combining the available hardware with the advent of "structured" high level program languages, it is found that engineers familiar with BASIC can now write relocatable code which compiles to execute at speeds approaching dedicated machine language. The result is the economical development of inhouse systems.

The next step is the specification of digitization hardware. Each sensor provides an analog DC or AC signal proportional to the measured parameter. Analog signals, unfortunately, are influenced by number of "noise" sources, many of which have been previously introduced as causing discontinuities (Table 1). Designing the digitization section with these in mind will preclude numerous problems later. Since most instrument power supplies available for sensor excitation will exhibit between 5.0 and 10.0 millivolts ripple, and most proximity probe circuits operate over a 10.0 volts direct current (vdc) full scale range, the ideal analog-to-digital (A/D) conversion precision should be not be under 0.1 percent, or about 10 bit (1024-step) A/D conversion. Multichannel A/D boards available for the PC bus fall into eight or twelve bit conversion precisions, so a 12-bit A/D will be used, and the two least significant bits of results will be masked. The inclusion of all bits only serves to concentrate downstream analysis on the power supply ripple rather than the vibration signal. A further reduction of precision may be necessary and will be discussed later.

A/D boards commercially available typically handle up to 16 multiplexed input channels, which can be scanned up to 100,000 readings per second and placed directly in computer memory via DMA (Direct Memory Access) control. For the example system, one board will be more than adequate. Between the raw sensor outputs and the A/D multiplexer signal conditioning will be required.

The input signals represent three different analog scales. The thermocouples are best signal conditioned with a cold junction and linearization module, the output of which will match the full scale A/D range (usually 10 vdc). The proximity keyphase switch may be excited with a dedicated supply and directly applied as an A/D multiplexer input if *no other system will be sharing its output*. The proximity sensors, already shared with the supervisory system, must have electrical isolation along with amplification to the desired range. For a 20-mil peak-to-peak vibration range, a 200 mV/mil proximator (5.0 vdc gap), and a 10-volt A/D, this gain will be

$$G = \frac{10 \text{ volts}}{20 \text{ mils} \times .200 \text{ volts/mil}} = 2.5 \quad (2)$$

The analog isolator and amplifier may be installed as one signal conditioning module. Isolators employing optical techniques will low-pass filter data more so than those using transformer couplings. In the example, the peak frequency of vibration data ever expected would be about four times maximum running speed, or 400 Hz for a 6000 rpm rated drive turbine. Isolating signal conditioners must therefore at least pass 400 Hz signals. This value may surprise users of commercial analyzers, familiar with examining 10 KHz bandwidths. The fact is that journal bearing rotors simply do not exhibit high frequency phenomena. Only where solid evidence exists that higher frequencies of concern are present, will we worry about greater than four-times running frequency bandwidths.

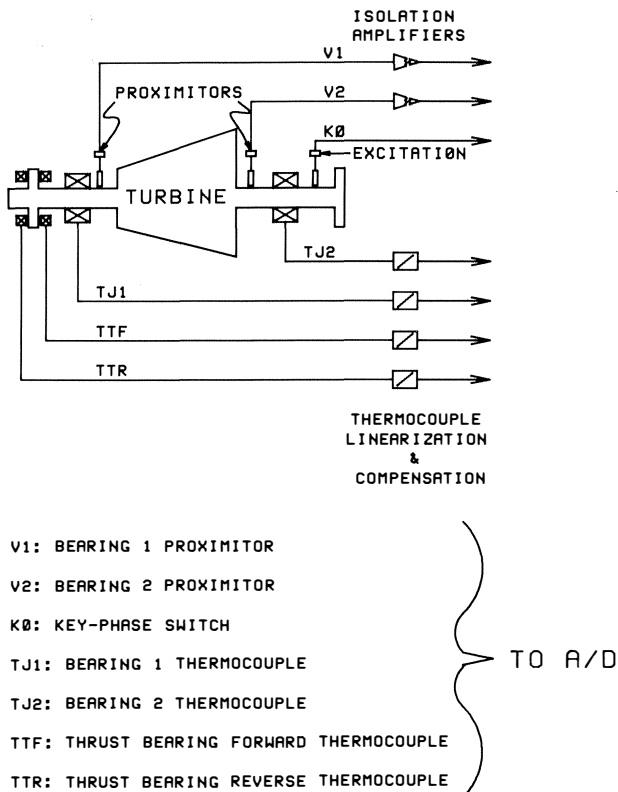


Figure 3. Online Diagnostic Sensor Diagram.

The digitization system will be packaged in an environmentally sealed (and controlled, if necessary) enclosure as close to the sensors as possible, to minimize noise in analog channels. The digitization computer will be programmed to collect and analyze the sensor input to the degree necessary to detect the desired phenomena. The digitizer hardware layout is diagrammed in Figure 4. A condensed data stream will then be transmitted via an RS-485 serial communications link to the host computer in a lab or office environment where it will be stored periodically, displayed, and automatically analyzed for pattern recognition.

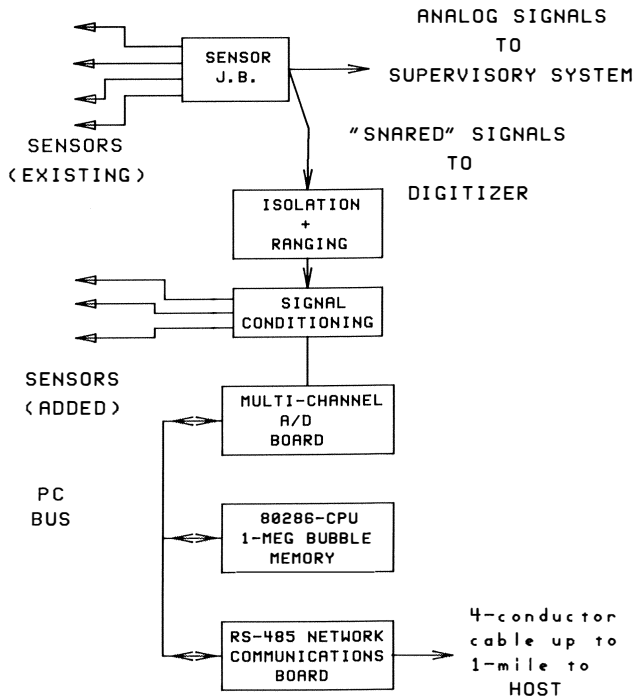


Figure 4. Digitizer Block Diagram.

The digitizer software in the example will not detect the on-line diagnostics—the routine will not have “time” for that. But it must acquire all the data that the host will need. For the example online diagnostics, this will include:

- the root mean squared (RMS) amplitude and phase angle of the vibration sensor inputs at least once every five seconds.
- the turbine speed with each vibration/phase data set.
- the bearing temperatures every five seconds.
- the principal vibration frequency (usually running speed and the principal fractional and integer multiple harmonics about every ten minutes).

Manufacturers of the A/D boards provide software “drivers” which enable control from the structured BASIC program, including channel, sampling rate, and sampling sequence control. For the thermocouple inputs, only a bits-to-degrees conversion is necessary. For the remainder, work a little more. The designer is often faced with a hardware vs software solutions decision. Such is the case with vibration RMS determination. The RMS amplitude may be calculated strictly by software examining a time span of the raw vibration signal (as the frequency-resolution algorithms will), or by hardware “RMS detectors.” In the use of the latter, the raw vibration signal is branched as shown in Figure 5 both directly to a multiplexer channel and indirectly through an RMS detector.

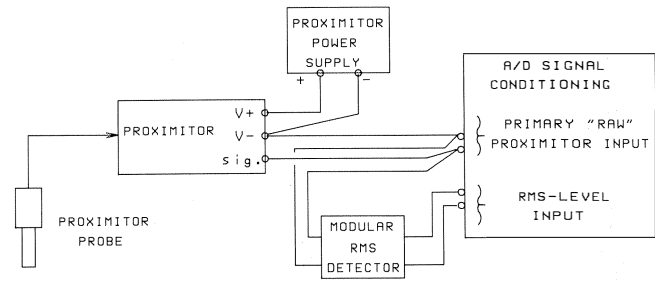


Figure 5. RMS Detection.

The latter method is preferred in this case, to reduce the software overhead and to “speed up” the acquisition process.

In order to accomplish the processing of the five second interval data in a timely manner, a timer interrupt is used to trigger a priority routine which cycles the A/D (via driver calls) through its data inputs. Also in this priority routine, the relative phase angle of each vibration input will be measured by starting a DMA transfer with the keyphase negative transition and sequentially loading until the next keyphase negative transition. The sample count times the sample period will very accurately define the rotor period, the inverse of which is the current running speed frequency. Each vibration phase may be determined by the ratio of sample counts between keyphase transitions and point where the raw vibration signal equals its RMS counterpart multiplied by 360-degrees. Although this phase angle is not necessarily the equal to the running speed frequency phase angle, the difference is minute in all but extreme cases. The last task of the digitizer software is the frequency resolution of principal components.

Because the online diagnostic algorithms in the host computer require a “hard” spectrum to properly recognize data patterns, an FFT approach with the resulting “soft” spectrum solutions will not be used. A “hard” spectrum is one without ambiguity. An indicated line on a hard spectrum represents a very real and present discreet vibration component which is not necessarily a pure sineusoidal waveform. A good example of a hard spectrum is the result of passing visible light through a diffraction grating. Each component light wavelength (and frequency) presents itself as a clean spectral line, without sidebands or sloping lobes. Since it is not desired to dedicate a lab grade dedicated channel analyzer network next to the turbine for cost, durability, and convenience reasons, a frequency will be written that will resolve algorithms to be resident in the digitizer computer program and function as a *background* routine.

This is not as monumental a task as one might think. To make this process easier for plant instrument engineers, the algorithms have been included, written in structured BASIC, which form a very accurate vibration frequency resolver producing hard spectra from DMA-loaded data blocks on industrial PCs. The BASIC listings are presented in the APPENDIX. These routines are declared public domain software through this publication, and may be used by any end user for inhouse systems. The routines may not be resold for value. All routines listed are currently in use by online diagnostic systems.

Algorithmic routines titled DATALOG, MXSORT, BEAT, PHASER, and DEMOD comprise the frequency resolver. The technique employed is by no means the only way to resolve vibration frequencies, and experimentation is encouraged for anyone with an idea of a new approach. This method mimics in software processes compatible to BASIC the function of a phase locked loop demodulator. It will helpful in understanding the use of the algorithms to first review the function of the analog hardware equivalent.

A phase locked loop (PLL) circuit is a hybrid analog/digital device with the remarkable ability to lock an internally generated square wave exactly in phase with a raw composite signal component, provided the component lies within a preset bandwidth of the PLL, known as its "capture range." PLLs have revolutionized radio tuning circuits, radar receivers, and other applications where high accuracy *demodulation* is needed. To determine the amplitude of the component under lock, the squarewave output of the PLL is multiplied times the raw composite waveform with the result integrated over a fixed time period. Both the hardware circuit and the resulting waveform multiplication and integral are illustrated in Figure 6. If the PLL succeeds in locking on a component, the resulting continuous multiplication will align the positive half cycle of the square wave with the positive half cycle of the component. The negative half cycle will then align with the negative half cycle of the component, with the resulting multiplication over one cycle yielding a predominance of positive products. The integral of these positive products will continuously increase over multiple cycles, reaching a level proportional to the component amplitude when sufficient cycles have been run to reduce low frequency component error. The circuit of Figure 6 is quite practical, and in fact is in current application by the author's company for specific types of frequency resolvers. It's only real drawback is the hardware intensity of the design, and the need of careful trimming of the analog multiplier to insure accuracy. To eliminate the hardware complexity, the software equivalent was developed. Each routine works to duplicate the PLL demodulator process, with special attention given to resolving the most likely components.

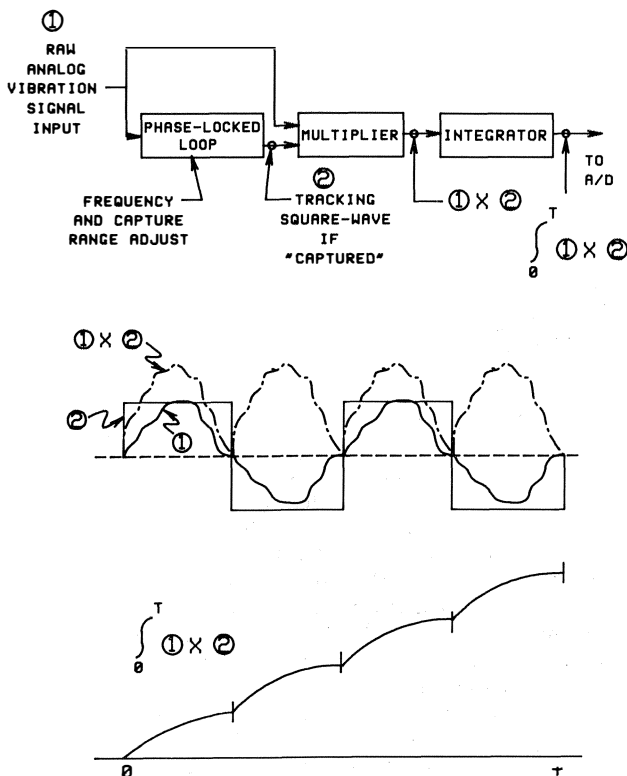


Figure 6. The Phase Locked Loop Demodulator.

The DATALOG routine converts the DMA loaded A/D values into BASIC integer array elements masked to 8, 9, 10, 11, or 12 (no mask) bits according to the optimum data precision. A software filter may be employed automatically (to be discussed)

which selects how many successive samples are to be averaged into one "reading." A running sample value is calculated, and sample counts which define positive function zero crossings are recorded in an array. The immediate slopes, i.e., the difference between successive readings is recorded in an array. The scan loop of the raw data continues until the eleventh function zero crossing is reached. The array of slope values is now examined by the MXSORT routine to sort out sample counts to function local maxima. Exiting the MXSORT loop, the background routine calculates the average gap between function positive zero crossings (ZGA percent), and the average gap between local maxima samplings (MGA percent).

If the ZGA percent value is equal to the MGA percent value, and the individual local maxima lie consistently one-quarter cycle after the positive zero crossing points, assume that the signal contains but one frequency component, and go on to examine the next vibration signal. If the former is true, but the latter is false, assume a pair of closely spaced "beat frequencies" exist, and execute the BEAT routine to determine the actual periods. This is quite a signal analysis accomplishment, as closely spaced beat frequencies are quite unresolvable by most analyzers. If either equality fails, the fundamental fractional and multiple harmonics of both the ZGA percent period and the MGA percent period must be proven present or absent. At this level of the analysis, "suspect periods" have been determined, which may or may not define component elements of the raw signal. The PHASER routine is next run to eliminate all but the most likely of these periods, and, like the PLL, lock phase with the component period. Since PHASER is a sort of quick test to confirm absence, but not presence, of a component, the resulting survivors are placed in a potential period array along with their phase offsets. The final routine in the algorithm is DEMOD, which performs both the analog multiplier and the integrator of the PLL demodulator circuit. Locking a software \pm unity valued square wave function in multiplication with the eleven zero crossing span of readings from DATALOG, integral values are generated at each zero crossing point. Using a derivation of the criterion for convergence developed by Augustin-Louis Cauchy, the French mathematician, in 1821, the demodulation need not be extended further. The validity of the demodulated amplitude is proven by an increasing value of the "negative blocking factor" or the percentage of positive multiplications (integral samples) taken at zero crossing points. If the negative blocking factor increases after nine or ten crossings, a component of the demodulator period is present in the raw signal.

The execution time of the frequency-resolver algorithm is a function of the signal complexity. Single component waveforms will resolve in seconds. Complex harmonics and beat patterns can take several minutes. As a background task, these execution times are generally acceptable. The solution array of periods and amplitudes completed by DEMOD will be form a solid, hard spectrum for pattern recognition.

A specific area of concern in this and any analysis technique is maintaining analog signal quality. Although there is limited exposure to noise sources by almost immediately digitizing data, there still exists potential for intrusion by 60-cycle and other predominant environmental frequencies. These will be passed by the frequency resolver as spectral lines, although false harmonics will not be generated. To harden an online diagnostic system against such noise, one may use an active system of automatically controlled digital filtering. Three digital filter factors are automatically adjusted. The first, the run averaging of data defined by the variable "TL3" in DATALOG, permits adjustment by the host computer of the data averaging count, or how many samples either side of a given sample are summed and averaged for that particular reading. The second filter factor is defined by the variable "SP2" in DATALOG, and is the mask to

reduce the 12-bit A/D data precision to 11, 10, 9, or 8, to avoid background "noise processing." The third filter factor is found as "TL2" in the MXSORT routine, where it establishes the slope characteristics of an acceptable local maximum.

All three filter factors are adjusted automatically by the feedback loop depicted in Figure 7. To "monitor" background noise levels, an instrumentation amplifier with about the same gain as the resident sensor is placed in each analog junction box. The instrument amp uses the sensor power supply through a voltage divider as a locally "fixed" reference. The amplifier output is wired to the digitizer in the same path as the sensor wires and acquires the equivalent background noise.

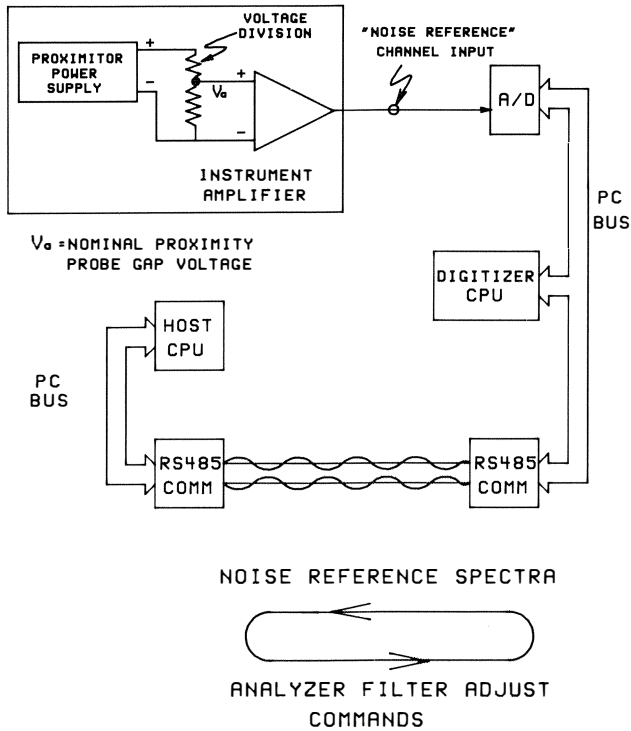


Figure 7. Filter Feedback Loop.

By using this noise reference channel, the results of executing the frequency resolving algorithm may be used to restrict or lax the three filter factors. The noise reference channel is scanned on an hourly basis or by special host request.

HOST COMPUTER SYSTEM

Thus far our turbine online diagnostic system has been assembled from industrial PC components and some "cookbook" software routines. The system will now benefit from the plant experience in preparing the on-line diagnostic routines of the host program.

Routines for detecting loss of rotating mass are based upon the assumption that all normal vibration vector changes will occur gradually, with speed or load change, unlike a loss of rotor mass which will step-change the vectoral positions just like a balance move. To recognize this pattern, we keep track of the cartesian coordinates of each turbine vibration input, and trigger our loss-of-mass diagnostic if the absolute vector position changes over 1-mil (0.001 in) between five-second scans. On machines which suffer large, rapid load changes the position change limit may have to be raised. Mathematically, the cartesian conversion of the data may be represented as

$$x = M \times \sin(\theta) \quad y = M \times \cos(\theta) \quad (3)$$

- (x,y) = cartesian coordinates in mils
- M = vibration magnitude in mils
- θ = vibration phase angle

The calculation of the absolute change in vectoral position is:

$$Z = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (4)$$

A loss-of-mass reaction (Figure 8) occurred at Commonwealth Edison Company's Zion Unit 1 low pressure "B" turbine in 1984. Note that by "crossing zero," the net change in *magnitude* was very small, although a large change in *vectoral* position resulted. The BASIC code equivalent of this can be found as the MASSLOSS routine in the APPENDIX.

The online diagnostic for rotor rubs has been published for years as a two effect pattern. First, the vibration vector pattern begins a counter rotational closed path loop pattern on a Bodé plot. Secondly, since the rotor is being excited, the natural frequency component of the rotor (usually about 1/2 \times or slightly less) will appear on the hard spectrum. To recognize the Bodé plot pattern, continuously log the vibration cartesian coordinate individual changes with incoming data. The algorithm proceeds through a series of "confirmations" which occur when the (x,y) changes indicate a proper antirotational curve. If the current vectoral direction is the same as the last, there is no status change in the algorithm. If the direction indicates a counterrotational turn, the confidence level is incremented. If a confidence level of four (antirotation "turns") is reached, the rub pattern is recognized as complete. Vectoral turns with rotation reset the algorithm. The APPENDIX listing of RUB provides the BASIC code. A final confirmation of the online diagnostic is performed by locating the natural frequency component from the hard spectrum.

The misalignment pattern to be recognized will be a twice running speed component and/or a deviation from normal journal bearing metal temperatures.

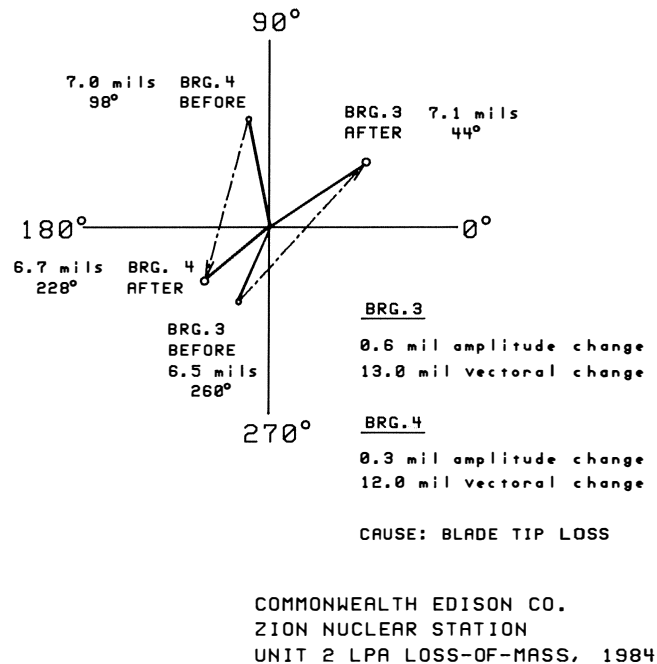


Figure 8. Zion Mass Loss

The bearing wipe pattern to be recognized is a short term high temperature excursion followed by a slightly cooler running bearing (due to increased oil flow).

Now having satisfied the design goals, additional data handling utilities may be added to the host program. The two final features to be discussed are calendar memories and file triggers.

Calendar memories in BASIC are data arrays containing data values taken at a given interval. The author has found that both a short term calendar of 60 points taken ten seconds apart (spanning a ten minute period) and a long term calendar of sixty points taken at ten minute intervals (spanning ten hours) are both valuable. It is also useful to record the *range* of the data parameter during the interval, rather than an instantaneous value. Calendars always hold the last 60 interval values, with the newest data overwriting the oldest. For machinery under special study, such as attempting to define a new online diagnostic pattern, a trigger technique can be used to save the short term calendars when a change occurs in key data.

A trigger routine may be initiated by a high parameter level, a vectoral shift, or any combination of data position. Following initialization, the trigger routine proceeds to load another one-half of the short term calendar, then write the calendar to a hard or floppy disk file. Through this method detailed, close spaced data both before and after an event are saved in condensed format for later evaluation of patterns.

CONCLUSIONS

The availability of sophisticated, low cost data acquisition hardware for the industrial PC computer bus, along with structured BASIC languages has opened the door for inhouse machinery online diagnostic systems design.

Hardware and software techniques can be employed in frequency resolution which are superior to commercial instruments in both accuracy of data and the ability to implement new diagnostics based upon operating knowledge of machinery problems.

A functional online diagnostic system provides early warning of impending problems and relieves technicians of the time consuming tasks of manual data gathering and analysis.

The question becomes—what is the goal of a machinery monitoring program?

- To generate reams of data for engineers to study?

or

- To warn at the earliest possible moment of a machinery problem?

If the answer is the latter, surely online systems are superior.

APPENDIX

Software provided in this paper is released as public domain software, and may be applied as desired by end users. The routines presented may not be commercially sold for any purpose.

Routines are written in structured BASIC language designed to be compiled. Some routines may be useful in interpreted BASIC, but will execute very slowly. Data identified as "External" is declared in the outer program structure.

1. DATALOG ROUTINE DIGITIZER RESIDENT

DATALOG serves as a background routine and main program structure. It is periodically interrupted by the five-second timer interrupt call for the A/D scans using the A/D manufacturer's driver. The code is presented in a format compatible to most structured BASIC's, however, some modification in the area of variable declarations and procedure calls in inevitable.

DECLARATIONS . . .

```

STRING ARRAY(50): SR$(200) (ALL RESULTS WRITTEN HERE)
BYTE ARRAY(30001): D1[7000H:0H]
' . . . Segment which A/D writes data to via DMA
INTEGER ARRAY (5001,2): VD%
' . . . Array to hold converted readings and "mean-to-reading"
INTEGER ARRAY(2001,2): A%
' . . . Array to hold local maxima location and "gap-from-previous"
INTEGER ARRAY (2001,2): B%
' . . . Array to hold zero crossing positive locations and gaps
INTEGER ARRAY(5001): S%
' . . . Array to hold slopes or d(VD%)/d(reading)
INTEGER ARRAY(100,2): PP%
' . . . Array to hold potential periods and phase offsets
INTEGER ARRAY(100,2): DP%
' . . . Array to hold results of detected periods and amplitudes
INTEGER: MXPT%, ZGA%, MGA%, IEND%, N, I, J, K, L, M, C
REAL: SUM, PER, MEAN
BYTE: AFLAG, CLICK, SP2, TL2, TL3
INTEGER: K1, K2, K3, K4, K5, K6, K7, K8, K9, K10
' . . . Assumes initializations for hardware completed . . .
' . . . J = channel to resolve of sequence from A/D 100 SUM=0
' . . . The first ten A/D channel values are converted to integers
' . . . K1 thru K10. MEAN(J) values are equivalent to the proximity
' . . . probe gap value in bits, determined on the previous scan by line
' . . . number 460, SP2 is the precision mask.
110 K10=(16*D1(J+55)+INT((D1(J+54) AND SP2)/16))-MEAN(J)
120 K1=(16*D1(J+1)+INT((D1(J) AND SP2)/16))-MEAN(J)
130 K2=(16*D1(J+7)+INT((D1(J+6) AND SP2)/16))-MEAN(J)
140 K3=(16*D1(J+13)+INT((D1(J+12) AND SP2)/16))-MEAN(J)
150 K4=(16*D1(J+19)+INT((D1(J+18) AND SP2)/16))-MEAN(J)
160 K5=(16*D1(J+25)+INT((D1(J+24) AND SP2)/16))-MEAN(J)
170 K6=(16*D1(J+31)+INT((D1(J+30) AND SP2)/16))-MEAN(J)
180 K7=(16*D1(J+37)+INT((D1(J+36) AND SP2)/16))-MEAN(J)
190 K8=(16*D1(J+43)+INT((D1(J+42) AND SP2)/16))-MEAN(J)
200 K9=(16*D1(J+49)+INT((D1(J+48) AND SP2)/16))-MEAN(J)
' . . . The raw data is now scanned in the following loop to locate
' . . . ζεροψηροσσηγ ποιητσ ανδ ασσηγν σλοπησ
210 N=J+54:L=0:B%(L,0)=0:R=0
220 WHILE R < 4990
230 N=N+6
240 R=INT((N-J)/6)+1
250 K1=K2:K2=K3:K3=K4:K4=K5:K5=K6:K6=K7:K7=K8:
K8=K9:K9=K10
260 K10=(16*K1(N+1)+INT((D1(N) AND SP2)/16))-MEAN(J)
270 ON TL3 GOTO 280,290,300,310,320,330,340,350,360,370
280 VD%(R-5,0)=K5:GOTO 380
290 VD%(R-5,0)=(K4+K5)/2:GOTO 380
300 VD%(R-5,0)=(K4+K5+K6)/3:GOTO 380
310 VD%(R-5,0)=(K3+K4+K5+K6)/4:GOTO 380
320 VD%(R-5,0)=(K3+K4+K5+K6+K7)/5:GOTO 380
330 VD%(R-5,0)=(K2+K3+K4+K5+K6+K7)/6:GOTO 380
340 VD%(R-5,0)=(K2+K3+K4+K5+K6+K7+K8)/7:GOTO 380
350 VD%(R-5,0)=(K2+K3+K4+K5+K6+K7+K8+K9)/8:
GOTO 380
360 VD%(R-5,0)=(K1+K2+K3+K4+K5+K6+K7+K8+K9)/
9:GOTO 380
370 VD%(R-5,0)=(K1+K2+K3+K4+K5+K6+K7+K8+K9+
K10)/10
380 S%(R-5)=VD%(R-5,0)-VD%(R-6,0)
390 SUM=SUM+VD%(R-5,0):VD%(R-5,1)=SUM/(R-5)
400 IF R=3 THEN GOTO 450
410 IF VD%(R-5,0)>=0 AND VD%(R-6,0)<=0 THEN
GOTO 420 ELSE GOTO 450
420 IF R-B%(L,0)<10 THEN GOTO 450' . . . high freq. cutoff
430 L=L+1:B%(L,0)=R-5
440 B%(L,1)=B%(L,0)-B%(L-1,0)
450 REPEAT IF L<11
460 MEAN(J)=VD%(R-5,1)
' . . . The slope array will now be examined to locate local
' . . . maxima within the data set.
470 IEND%=R:ZCP%=L:N=6:AFLAG=0:MXPT%=0

```



```

480 WHILE N<R-1 AND MXPT%<1000
    490 N=N+1
    500 MXSORT
510 REPEAT IF AFLAG=0
' . . . Waveform analysis is next . . .
'
520 IF ZCP%<4 OR MXPT%<3 THEN GOTO 530 ELSE GOTO 540
530 SR$(J)="Low Frequency Signal Below Bandwidth":EXIT
' . . . The above is executed when insufficient signal
' variation is detected.
540 I=0
550 FOR N=2 TO MXPT%
    560 I=I+A%(N,1)
570 NEXT N
580 MGA%=I/(MXPT%-2):' . . . Local Maxima Gap Average
590 I=0:CLICK=0
600 R=B%(3,1)
610 FOR N=2 TO ZCP%-1
    620 I=I+B%(N,1)
    630 IF ABS(R-B%(N,1))>17 THEN CLICK=1:' . . . Beat Freq.
        Present
620 NEXT N
630 ZGA%=I/(ZCP%-2):' . . . Zero crossing gap average
640 PCT%=0:' Potential period counter
'
' . . . the routine 'PHASER' is now repeatedly called to establish
' potential periods based upon ZGA% and MGA% following
' emulation of the phase-locked loop
'
650 PER=ZGA%:PHASER
660 PER=ZGA%/2:PHASER
670 PER=ZGA%/3:PHASER
680 PER=ZGA%/4:PHASER
690 PER=ZGA%*2:PHASER
700 PER=ZGA%*3:PHASER
710 PER=ZGA%*4:PHASER
720 IF ZGA%=MGA% THEN GOTO 740
730 PER=MGA%:PHASER
740 H=0
750 IF CLICK=1 THEN BEAT:' . . . if present, resolve beat
760 DEMOD:' . . . run phase-locked loop emulating demodulator
' . . . Results are now used to update SR$ string array . . .
770 SR$(J)='''':' . . . pre-clear results
780 FOR N=1 TO H
    790 APPEND SR$(j), '(' + STR$(DP%(N,0))'' ',' +STR$(DP%
        (N,1))+'''
800 NEXT N
'

```

The string elements of SR\$(J) then contain the hard spectrum of the vibration point in the form;

(period, amplitude)*(period, amplitude)* . . . (period, amplitude)
. . . With a pair of values assigned for each positively demodulated frequency component.

2. MXSORT ROUTINE DIGITIZER RESIDENT

MXSORT is called by DATALOG, and locates the time-domain positions of the raw signal local maxima based on TL2 values of consecutive rise/fall slopes. All external variables are defined in DATALOG.

```

INTEGER: I,J,K,L,M
EXTERNAL: AFLAG,S%( ),MXPT%,A%( ),IEND%,TL2
50 IF S%(N)>0 AND S%(N+1)<0 THEN GOTO 52 ELSE GOTO 60
52 IF TL2=1 THEN GOTO 80
53 IF TL2=3 THEN GOTO 70
54 IF S%(N-1)>0 AND S%(N+2)<0 THEN GOTO 80 ELSE GOTO
    210
60 IF S%(N)>0 AND S%(N+1)=0 THEN GOTO 62 ELSE GOTO 210
62 IF TL2=1 THEN GOTO 130
64 IF TL2=3 THEN GOTO 120
66 IF S%(N-1)>0 THEN GOTO 130 ELSE GOTO 210
70 IF S%(N-1)>0 AND S%(N-2)>0 THEN GOTO 130 ELSE
    GOTO 210

```

```

80 MXPT%=MXPT%+1
90 A%(MXPT%,0)=N
100 A%(MXPT%,1)+A%(MXPT%,0)-A%(MXPT%-1,0)
110 GOTO 210
120 IF S%(N-1)>0 AND S%(N-2)>0 THEN GOTO 130 ELSE
    GOTO 210
130 I=1
140 I=I+1:IF N+5+I>IEND% THEN AFLAG=2:GOTO 210
150 IF I>50 THEN AFLAG=1:GOTO 210
160 IF S%(N+1)>0 THEN GOTO 210
170 IF S%(N+1)<0 THEN GOTO 175 ELSE GOTO 140
172 IF TL2=1 THEN GOTO 180
173 IF TL2=3 THEN GOTO 175
174 IF S%(N+1+1)<0 THEN GOTO 180 ELSE GOTO 210
175 IF S%(N+1+1)<=0 AND S%(N+1+2)<0 THEN GOTO 180
    ELSE GOTO 210
180 MXPT%=MXPT%+1
190 A%(MXPT%,0)=N+(I/2)
200 A%(MXPT%,1)=A%(MXPT%,0)-A%(MXPT%-1,0)
210 ' EXIT PROCEDURE

```

3. BEAT ROUTINE DIGITIZER RESIDENT

BEAT resolves closely-spaced frequencies by analyzing the set of local maxima locations for the beat period. BEAT is called by DATALOG, where all external variables referenced are declared.

```

INTEGER: I,J,K,L,M,AMIN,AMAX
REAL#: X,Y,Z . . . NOTE DOUBLE-PRECISION!
EXTERNAL: VD%( ),A%( ),MXPT%,MGA%,ZGA%,PP%( ),PCT%,
    PER,PHASER

```

```

60 FOR N=1 TO MXPT%
    70 K=VD%(A%(N,0),0)
    80 S%(N)=K-M
    90 M=K
100 NEXT N
110 L=0:N=1
120 WHILE N<MXPT%
    130 IF S%(N)>0 AND S%(N+1)>0 THEN GOTO 500
    140 IF S%(N)<0 AND S%(N+1)<0 THEN GOTO 500
    150 IF S%(N)>0 AND S%(N+1)<0 THEN GOTO 200
    160 IF S%(N)>0 AND S%(N+1)=0 THEN GOTO 300
    170 IF S%(N)<0 AND S%(N+1)>0 THEN GOTO 400
    180 IF S%(N)<0 AND S%(N+1)=0 THEN GOTO 440
    200 L=L+1
    210 AMAX=A%(N,0)
    220 GOTO 500
    300 I=0
    310 I=I+1:IF N+1>MXPT% THEN GOTO 500
    320 IF S%(N+1)<0 THEN GOTO 330
    325 IF S%(N+1)>0 THEN GOTO 500 ELSE GOTO 310
    330 AMAX=A%(N+1/2,0)
    340 L=L+1
    350 GOTO 500
    400 L=L+1
    405 AMIN=A%(N,0)
    410 GOTO 500
    440 I=0
    450 I=I+1:IF N+I>=MXPT% THEN GOTO 500
    460 IF S%(N+I)>0 THEN GOTO 470
    465 IF S%(N+I)<0 THEN GOTO 500 ELSE GOTO 450
    470 L=L+1
    480 AMIN=A%(N+I/2,0)
500 REPEAT IF L<2
560 X=ABS(AMAX-AMIN):IF X=0 THEN GOTO 600
570 PER=MGA%/(1-MGA%/X)
580 PHASER
600 ' EXIT PROCEDURE

```

4. PHASER ROUTINE DIGITIZER RESIDENT

PHASER examines a possible component period (PER). If the probability is not zero for the component, the PP%() array is amended to include the period and its phase offset. All external variables are declared in DATALOG.

```

INTEGER: N,I,J,K,L,DET,POL,S,BEG,LST,M
REAL#V,W,X,Y,Z,TOG,PHA
REALARRAY(10000):BF/X . . . USUALLY
EXTENDED MEMORY

```

```
EXTERNAL: VD%( ),PP%( ),PER,PCT%
```

```

50 TOG=PER/2:PHA=0
55 M=PER/50
56 IF M<1 THEN M=1
60 FOR J=0 TO 8
  70 PHA=PHA+PER/8
  80 K=PHA-PER/8
  90 N=0:DET=0
  100 WHILE N<PER+1
    110 K=K+M:N=N+M
    120 IF K>=2*TOG THEN K=0
    130 IF K=TOG THEN POL=0
    140 IF K>TOG THEN POL=1
    150 IF K<TOG THEN POL=-1
    160 X=POL*(VD%(N,0)-VD%(PER,1))
    170 IF X<0 THEN DET=DET+1
  180 REPEAT
  190 BF(J)=1-DET/(PER/M)
200 NEXT J
205 Y=0
210 FOR J=1 TO 8
  215 IF BF(J)>Y THEN L=J
220 NEXT J
230 BF(L)=0:Y=0
240 FOR J=1 TO 8
  250 IF BF(J)>=Y THEN S=J
260 NEXT J
270 IF S>L THEN N=S:S=L:L=N
280 BEG=S*PER/8:LST=L*PER/8
290 FOR J=BEG TO LST STEP M
  300 K=J-1
  310 N=0:DET=0:X=0
  320 WHILE N<PER+1
    330 K=K+M
    340 N=N+M
    350 IF K>=2*TOG THEN K=0:POL=+1
    360 IF K=TOG THEN POL=0
    370 IF K>TOG THEN POL=1
    380 IF K<TOG THEN POL=-1
    390 X=POL*(VD%(N,0)-VD%(PER,1))
    400 IF X<0 THEN DET=DET+1
  410 REPEAT
  420 BF(J)=1-DET/(PER/M)
430 NEXT J
440 I=0:W=0
450 FOR J=BEG TO LST STEP M
  460 IF BF(J)>W THEN I=J:W=BF(J)
470 NEXT J
480 IF BF(I)<.5 THEN GOTO 600
490 PCT%=PCT%+1
500 PP%(PCT%,0)=PER
510 PP%(PCT%,1)=1
600' EXIT PROCEDURE

```

5. DEMOD ROUTINE DIGITIZER-RESIDENT

DEMOD emulates a phase-locked loop demodulator circuit to resolve the amplitudes of periods found to potentially be present by the PHASER routine tests. Results are placed in the DP% array with the first element being the period, the second the resolved amplitude. The H integer maintains the count of resolved components.

```

INTEGER: I,J,K,L,M,N,POL,PHA,PER,S
REAL# W,X,Y,Z,TOG,MEAN,RSUM,DET
REAL# ARRAY(100):BF
EXTERNAL: H,PCT%,PP%( ),VD%( ),DP%( )
60 FOR J=1 TO PCT%
  70 PER=PP%(J,0)
  80 IF PER<9 THEN GOTO 390
  90 IF PER>2000 THEN GOTO 390

```

REQUIRES

```

100 PHA=PP%(J,1)
110 TOG=PER/2
120 I=4991
125 M=PER/50
126 IF M<1 THEN M=1
130 X=0:Y=0:LST=0:RSUM=0
140 K=PHA-1:N=0:L=1:S=0
150 WHILE N+M<I
  160 K=K+M
  170 N=N+M
  180 IF K>2*TOG THEN K=0
  190 IF K=TOG THEN POL=0
  200 IF K>TOG THEN POL=1
  210 IF K<TOG THEN POL=-1
  220 RSUM=POL*(VD%(N,0)-VD%(L*PER,1))
  230 X=X+RSUM
  240 IF RSUM<0 THEN DET=DET+1
  250 IF N=L*PER THEN GOTO 260 ELSE GOTO 300
  260 LST=LST+1-DET/(N/M)
  270 IF X/(N/M)<0 THEN S=S+1
  280 L=L+1:BF(L)=1-DET/(N/M)
  290 IF BF(L)<.60 THEN S=4
  300 REPEAT IF L<11 AND S<=3
  310 IF S>3 THEN GOTO 390
  320 IF BF(L)>=LST/L-.03 THEN GOTO 350
  330 IF BF(L-1)>=LST/L-.03 THEN GOTO 350
  340 IF BF(L-2)>LST/L-.03 THEN GOTO 350
  350 IF X/(N/M)<10 THEN GOTO 390
  355 H=H+1
  360 DP%(H,0)=PP%(J,0)
  370 DP%(H,1)=X/(N/M)
390 NEXT J
400' EXIT PROCEDURE

```

6. MASSLOSS CALCULATION HOST RESIDENT

The BASIC code lines for loss of rotating mass pattern recognition are given as:

```

REAL: X1,X2,Y1,Y2,M1,M2,P1,P2,SHIFT
' M1 and M2 are successive point magnitudes
' P1 and P2 are successive point phases
' SHIFT is the net vectoral change in position, unit-dependent
100 X1=M1*SIN(P1*.0174533)
110 Y1=M1*COS(P1*.0174533)
120 X2=M2*SIN(P2*.0174533)
130 Y2=M2*COS(P2*.0174533)
140 SHIFT=SQRT((X1-X2)^2+(Y1-Y2)^2)

```

"SHIFT" is the compared to an allowable load-change induced vectoral position change between samples.

7. RUB ALGORITHM HOST RESIDENT

Using the same declarations as MASSLOSS, AND ADDING; REAL ARRAY (30): QUAD INTEGER: CONF

```

200 IF (X2-X1)>0 AND (Y2-Y1)>0 THEN QUAD(N)=4
210 IF (X2-X1)<0 AND (Y2-Y1)>0 THEN QUAD(N)=1
220 IF (X2-X1)<0 AND (Y2-Y1)<0 THEN QUAD(N)=2
230 IF (X2-X1)>0 AND (Y2-Y1)<0 THEN QUAD(N)=3
240 WHILE R<=N-1
  250 IF QUAD(R)=1 AND QUAD(R+1)=1 THEN GOTO 500
  260 IF QUAD(R)=1 AND QUAD(R+1)=2 THEN CONF=CONF+1:GOTO 500
  270 CONF=0:GOTO 500
  270 IF QUAD(R)=2 AND QUAD(R+1)=2 THEN GOTO 500
  280 IF QUAD(R)=2 AND QUAD(R+1)=3 THEN CONF=CONF+1:GOTO 500
  290 CONF=0:GOTO 500
  300 IF QUAD(R)=3 AND QUAD(R+1)=3 THEN GOTO 500
  310 IF QUAD(R)=3 AND QUAD(R+1)=4 THEN CONF=CONF+1:GOTO 500
  320 CONF=0:GOTO 500
  330 IF QUAD(R)=4 AND QUAD(R+1)=4 THEN GOTO 500
  340 IF QUAD(R)=4 AND QUAD(R+1)=1 THEN CONF=

```

```
CONF+1:GOTO 500
350 CONF=0
500 NEXT R
510 IF CONF=0 THEN N=1
520 IF CONF=4 THEN . . . PATTERN RECOGNIZED, EXIT
ALGORITHM
```

BIBLIOGRAPHY

- Cheng, D. K., "Analysis of Linear Systems," Reading, Massachusetts: Addison-Wesley (1959).
- Eshleman, R. L., "Vibration Analysis of Low Speed Equipment," Vibration Institute (1986).
- Gillett, Philip, "Linear Mathematics," Boston, Massachusetts: Pringle, Weber and Schmidt (1970).
- Harris, F. J., "On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform," IEEE Proceedings, 66 (1)(1978).
- Saunders, F. A., "Physics and Music," Scientific American (1948).

