To the Graduate Council:

I am submitting herewith a thesis written by Curtis S. Jones entitled "A Study of Classical and Parametric Spectral Techniques for Tube Leak Detection Using a Visual Programming Language." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

Daniel B Koch

Dr. Daniel B. Koch, Major Professor

We have read this thesis and recommend its acceptance:

Donald W. Bouldin

Accepted for the Council:

CWMinkel

Associate Vice Chancellor
and Dean of The Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at the University of Tennessee, Knoxville, I agree that the Library shall make it available to borrowers under the rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgement of the source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in his absence, by the Head of Interlibrary Services when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature Curtis Jones

Date 7/22/92

A STUDY OF CLASSICAL AND PARAMETRIC SPECTRAL TECHNIQUES FOR TUBE LEAK DETECTION USING A VISUAL PROGRAMMING LANGUAGE

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Curtis S. Jones

August 1992

ACKNOWLEDGEMENTS

This research was conducted on the campus of the University of Tennessee in Knoxville (UTK) during the period January 1991 through May 1992. The project was funded by a research grant provided by the Tennessee Valley Authority (TVA).

This thesis research was made possible because many people gave their time, advice, and encouragement. I want to thank Dr. Daniel B. Koch, my major advisor, for the guidance, encouragement, and trust he provided me throughout this research project. I also want to thank Keith W. Morris at TVA for his patience and enthusiasm for this project. The members of my thesis committee also have my gratitude for giving their time and experience.

I certainly want to thank my family, especially my parents Clyde and Georgine, for all the help and support they have provided me. Without their encouragement I would have never begun graduate school. Bill and Dorothy Mullican have also been extremely helpful in many ways, and I will always be in debt to their kindness. I would also like to say thank you to my wife Kris for all the love, encouragement, and understanding she has patiently shown throughout this time. Last, I would like to thank God for the love and faith he has given me that makes life worthwhile.

ABSTRACT

Detecting incipient tube leaks in fossil-fueled boilers consistently and reliably has proven to be difficult. Traditionally, boiler operators have relied on indicators such as fluctuation in pressure or obvious abnormal noise during furnace operation. These methods are ineffective in that they often do not provide early enough detection to avoid a forced outage.

An acoustic leak detection system was developed in Europe in the 1970s and introduced to U.S. utilities in the 1980s. Most of these systems are hardware-intensive and employ primarily time-domain signal processing in the detection method. This research represents the approach, analysis, and conclusions drawn by applying classical and parametric spectral techniques to the leak detection problem. It is shown that a software-intensive system utilizing spectral techniques holds great promise for earlier, more reliable leak detection over the current hardware-intensive system. Contrasts and comparisons between the classical and parametric techniques are examined. The techniques are programmed with a visual programming language, which has proven to be a powerful and flexible tool that has greatly reduced development time for program changes over traditional text-based programming environments.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

In fossil-fueled boilers, water running in tubes or pipes along the boiler wall is heated by a large flame until it is converted into superheated steam. The steam supplied by the boiler is then converted into usable electric energy by other power plant components. The tubes however, sometimes develop small leaks that can interrupt the power generation process. These leaks expand over time due to the extreme heat and pressure brought to bear on a weakened tube wall. If a leak becomes large enough, the boiler operator can no longer maintain adequate pressure and power levels, and a forced shut-down to repair the leak damage must occur.

Traditionally, boiler operators have relied on indicators such as fluctuation in pressure or obvious abnormal noise during furnace operation. These methods are ineffective in that they often do not provide early enough detection to avoid a forced outage. This also proves costly. Bringing a unit down when there is a heavy power demand, or being unable to meet fully the power demand because of low efficiency, usually means the utility must buy power elsewhere at high prices.

The desire is to develop a reliable, automated system to detect a pinhole leak in its earliest stages, alert an operator to its existence, approximate its location, size, rate of growth, and whether it causes any other leaks. With this information, an operator could schedule an outage for investigation and repair during a weekend when replacement power is cheap or when other units could cover the demand.

It is this desire to detect leaks early that led U.S. utilities to investigate the success certain European utilities were having in early leak detection. The Central Electricity Generating Board (CEGB) of the United Kingdom was successful in developing an acoustic leak detection system during the 1970s that was introduced to U.S. utilities in the early 1980s. This system employs acoustic sensors, or microphones, installed in metal waveguides mounted in strategic locations in the furnace wall. The sensors each send a low level signal back to a central processing unit that filters out the dominant combustion noise, and determines whether the mean-square value of the acoustic power in the pass-band of each filtered signal exceeds a preset threshold. If a sensor's signal does exceed this threshold for a certain amount of time, a leak is declared to be near that sensor.

The Tennessee Valley Authority (TVA) has been interested in the development and implementation of an early, reliable leak detection system, as well as other power plant component diagnostic monitoring systems. In 1985, TVA installed the Online Diagnostic Monitoring System (OLDMS) at its Kingston Fossil Plant. The primary purpose of the OLDMS is to provide a test environment for a number of real-time diagnostic and performance monitoring systems. The Kingston Plant consists of nine boiler units that use pulverized bitimous coal for fuel. Unit 5 consists of twin furnaces rated at 200 MW each, and is the host unit for the OLDMS [1].

Performance monitoring systems were developed to calculate unit-heat-rate, as well as to monitor thermal efficiency of the turbine, feedwater heaters, boiler and condenser. Among the diagnostic tools are vibration monitors, water chemistry monitors, slag monitors to optimize soot-blowing, and an acoustic tube leak detector. These diagnostic systems were purchased and placed in the OLDMS to determine whether each can alert plant personnel to incipient problems that might require either preventative measures or immediate corrective action [1].

The present acoustic tube leak detector is the OLM-100 Acoustic Boiler Tube Leak Detection System purchased from NDT International, Incorporated of West Chester, Pennsylvania. It utilizes the same basic principles that the aforementioned CEGB system employs. Each sensor's signal passes through a local high-pass filter and preamplifier and is routed to a central processing station. At this processing station, NDT's Fluid Flow Detection (FFD) hardware modules pass each signal through a low-pass filter and compute a mean-square value for the sensor signal. The front-end high-pass filter passes signal frequencies above 1,000 Hertz (Hz) because combustion roar dominates the lower frequencies. The low-pass filter on the FFD module passes signal frequencies below 10,000 Hz to eliminate consideration of any high-frequency transients. Thus, the mean-square value calculated to determine leak presence considers signal content only between 1,000 and 10,000 Hz. Each sensor has threshold values set to activate an alarm if the mean-square value of a sensor's signal exceeds this threshold for a specified period of time [2].

Figure 1.1 and Figure 1.2 show the placement of acoustic sensors in the superheat and reheat furnaces, respectively, of Unit 5. Each sensor was strategically located to an area of the furnace known to be a tube leak trouble area. Each sensor is expected to monitor the area covered in a 25-foot radius. Figure 1.1 shows that sensors are concentrated along the superheat furnace's waterwalls and near the economizer section; whereas, Figure 1.2 shows that sensors are concentrated more in the upper areas of the reheat boiler and less along the furnace waterwalls. In both cases, 12 sensors monitor a furnace approximately 12 stories high [3].

Figure 1.3 shows the shape and placement of the metal waveguide in the boiler wall. The acoustic sensor is housed in the rear of the waveguide, but insulated from the waveguide structure so as to respond only to airborne acoustics. The preamplifier and filter assembly is located as close to the sensor as possible. The downward slope and negative acoustic boiler pressure normally keep the tube free of flyash accumulation; however, a test-hole was drilled in the waveguide to provide easy access for plant personnel to clean the waveguide by injecting compressed air.

In January of 1991, TVA started efforts to enhance boiler tube leak detection in a cooperative effort with the University of

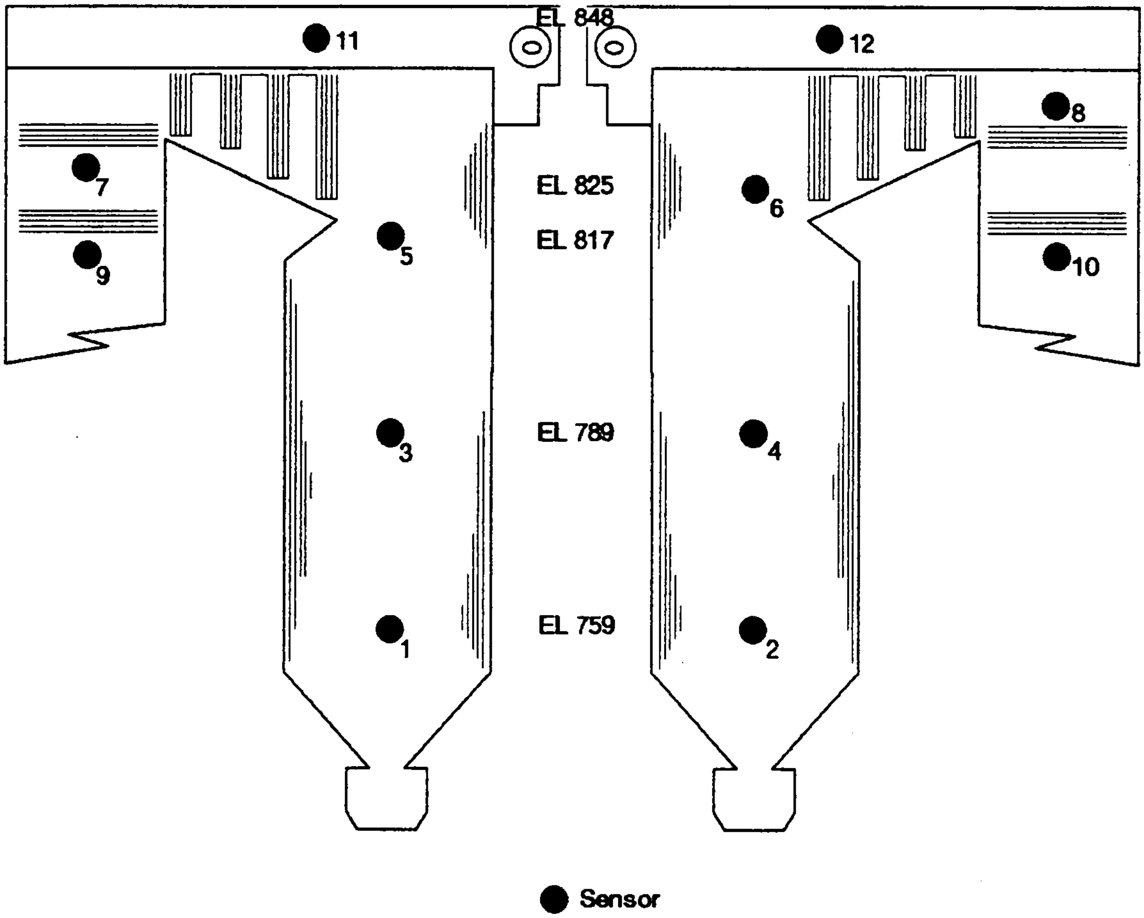


Figure 1.1. Location of Acoustic Sensors in Kingston Fossil Plant, Unit 5, Superheat Boiler.

Adapted from: R. D. Moss and R. L. Keyser, "An Introduction to the Tennessee Valley Authority Online Diagnostic Monitoring System Project," TVA publication TVA/PBO/R&D-89/8, Knoxville, TN, May 1989.

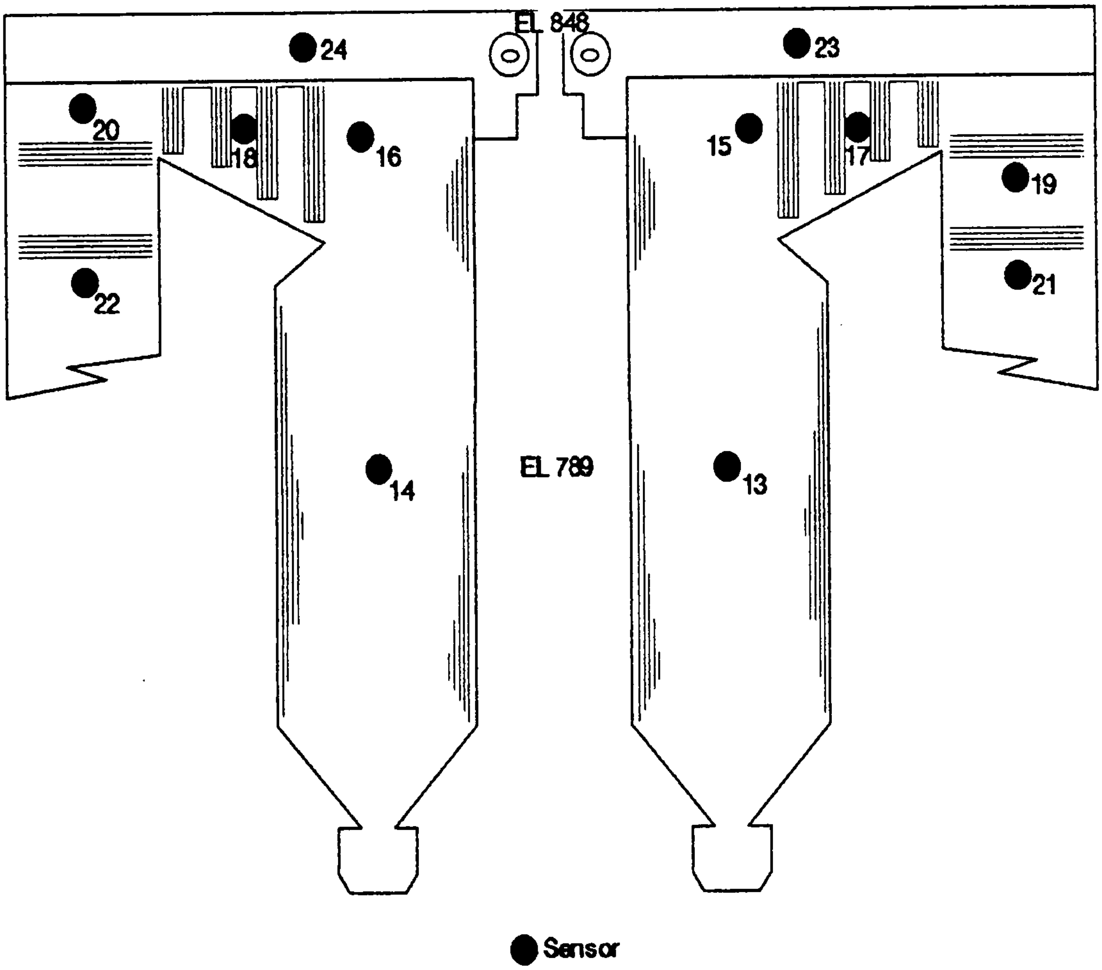


Figure 1.2. Location of Acoustic Sensors in Kingston Fossil Plant, Unit 5, Reheat Boiler.

Adapted from: R. D. Moss and R. L. Keyser, "An Introduction to the Tennessee Valley Authority Online Diagnostic Monitoring System Project," TVA publication TVA/PBO/R&D-89/8, Knoxville, TN, May 1989.

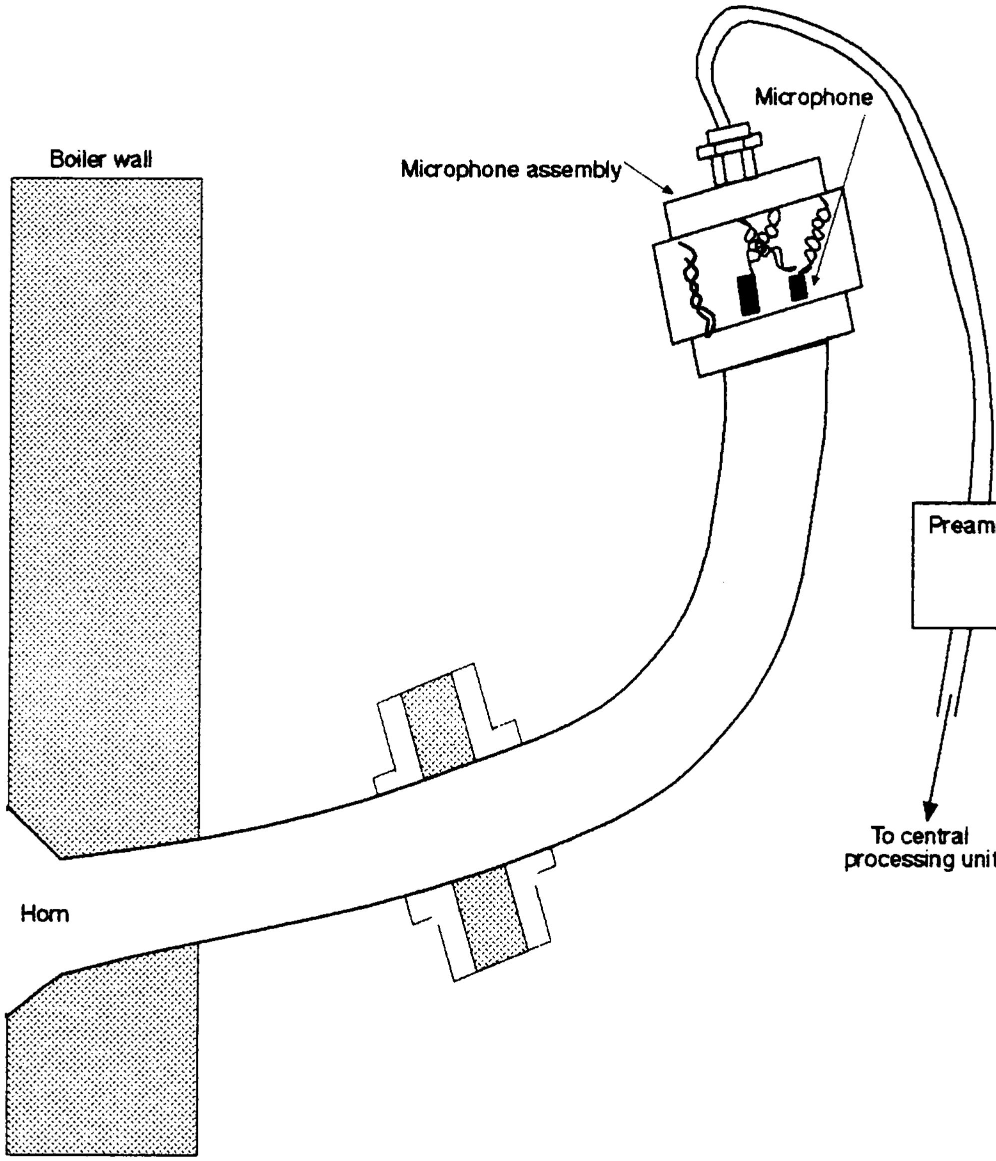


Figure 1.3. Diagram of Waveguide, Sensor, and Preamp Assembly.

Adapted from: "Technical Manual for the OLM-100 Acoustic Boiler Tube Leak Detection System", NDT International, Inc., West Chester, PA, January 1986.

Tennessee's Communications, Information, and Signal Processing (CISP) Group. It was believed that a more consistent, reliable, and less expensive leak detection system could be developed by investigating spectral analysis approaches. These approaches would be characterized as more software-intensive than the current hardware-dependent leak detection system. The ability to identify the earliest beginnings of a leak by capturing its spectral signature would represent significant savings over the present system.

This thesis represents the research, approach, analysis, and conclusions drawn thus far in the joint effort between the CISP Group and TVA. This cooperative effort between the CISP Group and TVA is referred to throughout this paper as the Boiler Acoustic Monitor (BAM) project.

The approach taken in the BAM project was to record each raw acoustic sensor signal on a Sony PC-108M, a digital tape recorder. The only raw acoustic signal readily accessible, though, was that which had been filtered at the preamplifier; therefore, each acoustic signal was intercepted and recorded before being processed by NDT's FFD module. The PC-108M employed automatically adjusting anti-aliasing filters to limit the processed band to 25,000 Hz when two channels were recorded simultaneously.

Tapes were replayed at the CISP Group Laboratory at the University of Tennessee and imported into data files on an Apple Macintosh IIx computer. A data acquisition board, the Lab-NB by National Instruments, sampled an analog signal regenerated by the

PC-108M at a sampling rate of approximately 41,667 Hz. Files were recorded in blocks of 4096 sampled points, creating a maximum spectral resolution of approximately 10 Hz.

LabVIEW, a Visual Programming Language (VPL), was chosen for program development primarily because of its unique Graphical User Interface (GUI), and its block diagram approach to program development. LabVIEW was developed specifically for the Apple Macintosh platform by the National Instruments Corporation, and seemed uniquely suited for the graphical challenge presented by the BAM project to locate spectral leak signatures. Its smart, mouse-driven GUI affords a friendly control and display for the operator. The self-documenting feature of its block diagram programming environment, along with many signal processing tools provided as built-in block diagram library routines, frees the programmer for rapid program development.

Both classical and parametric Power Spectral Density (PSD) estimation techniques were developed for the BAM project in LabVIEW to analyze the acoustic sensor data. Classical techniques are the industry standard primarily because most engineers and scientists are familiar with them, and because the processing requirements associated with them are not too demanding. The heart of popular classical techniques is the Fast Fourier Transform (FFT), coupled with the use of an explicit data window required to mitigate spectral instability. This instability is caused by an implicit data window produced whenever data is taken as a block from an infinite, random data sequence for processing.

The data outside the window is implicitly assumed to be zero by the very nature of block data acquisition. This is almost always an unrealistic assumption for random processes. Parametric techniques however, attempt to make a better assumption about the unobserved data by constructing filter models. These models, when driven by white noise, produce an output with the statistical characteristics of the original data sequence. This can lead to PSD estimates with better resolution, less distortion, and greater peak amplification properties.

Section II of this thesis is a literature review of important reports in the area of tube leak detection in fossil-fueled boilers. Section III gives background and a discussion of the classical PSD estimation methods utilized in the BAM project. Section IV presents background and a discussion of the parametric PSD estimation methods developed for the BAM project. Section V presents the analysis of the boiler acoustic sensor data for the determination of a leak detection technique, a comparison of what each classical and parametric method's relative performance might be as part of the leak detection algorithm, and a proposed leak detection system. Finally, Section VI presents conclusions derived from the analysis, and possible future directions for the continuance of the BAM project.

II. LITERATURE REVIEW

This section presents an overview of references that have contributed significantly to this research either in background, concept, or results. First, information on the acoustic signal characteristics of boiler noise sources is summarized. Next, work which contributed to necessary background information on leak occurrences and consequences is cited. A summary is then given of present leak detection systems available to U.S. utilities. Finally, an explanation is given as to how the literature review affected the scope and direction of this research.

A. Boiler Acoustic Noise Sources

Putnam and Faulkner [4] showed that combustion roar produces a smooth noise spectrum related to the flame reaction chemistry and turbulence level of the combustion region. The spectrum is broadband with no predominant frequencies. The maximum amplitude region of the spectrum typically occurs between 100 and 1000 Hz, and appears to be related to the chemically controlled reaction rate per unit volume of fuel.

When a flame is inside an enclosure, broad peaks appear in the spectrum at the natural frequencies of the enclosure. A furnace's internal shape and the acoustical properties of the furnace lining amplify these peaks, as Figure 2.1 illustrates. However, Putnam and

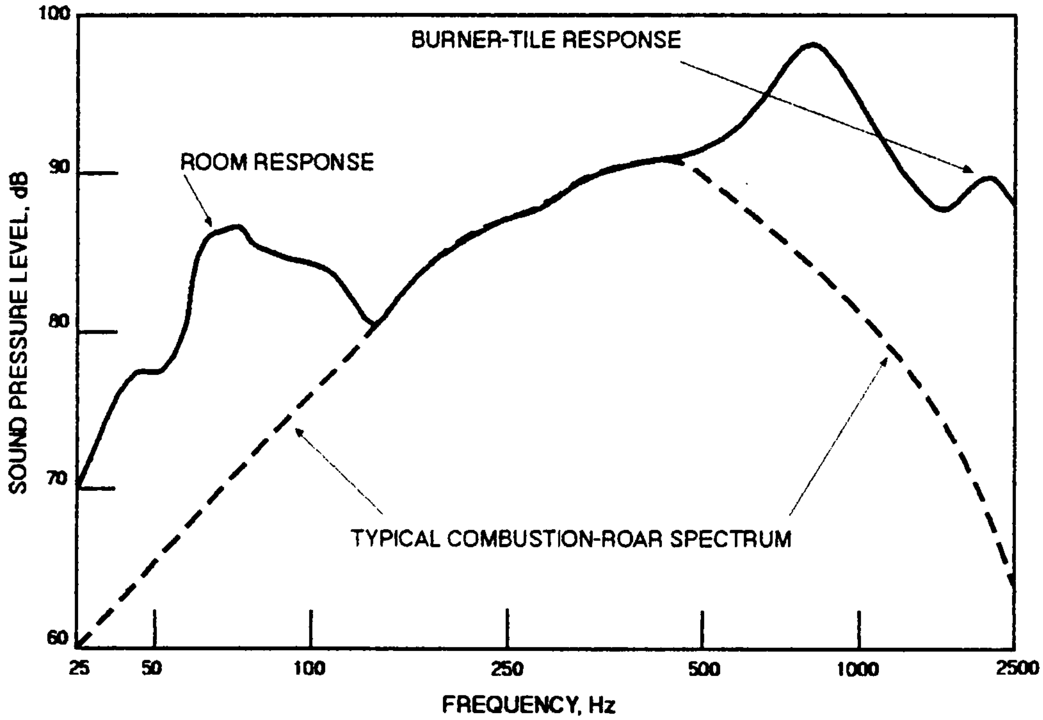


Figure 2.1. Amplification of Peak Frequencies Caused by Boiler Enclosure.

Adapted from: A. A. Putnam and L. Faulkner, "An Overview of Combustion Noise," Journal of Energy, November-December 1983, pps. 458-469.

Faulkner note that from a practical point of view, fan or blower noise can be of greater magnitude than the combustion roar.

The findings of Putnam and Faulkner may explain the similarities and differences in characteristic spectrum between sensor channels found during the BAM project. Figure 2.2 shows a typical spectrum for Sensor 4 and Sensor 8 of the superheat furnace of Unit 5. The top graph shows several peaks between 1,500 and 6,000 Hz that are typical of Sensor 4; whereas, the bottom graph shows typical peaks for Sensor 8 centered at different frequencies. The shape of the waveguide that houses the sensors may contribute to the spectral similarities between the two sensors, while the different locations in the furnace may contribute to the differences in spectra.

It becomes necessary then, to characterize what a leak might look like, and where it might appear in the power spectrum. One EPRI commissioned report by Stulen [5] stands out as a definitive work on trying to characterize tube leaks. According to Stulen, when a leak develops in a tube wall, either pressurized water or steam shoots out of the hole as a steam jet. The jet's turbulence generates a continuous random sound or vibration that is broadband and travels through the air in the boiler. This noise is added to the boiler's ambient noise.

Stulen states three important characteristics of a jet that should be considered in any leak detection system. First, the acoustic power associated with a leak-jet increases with the mass flow rate and

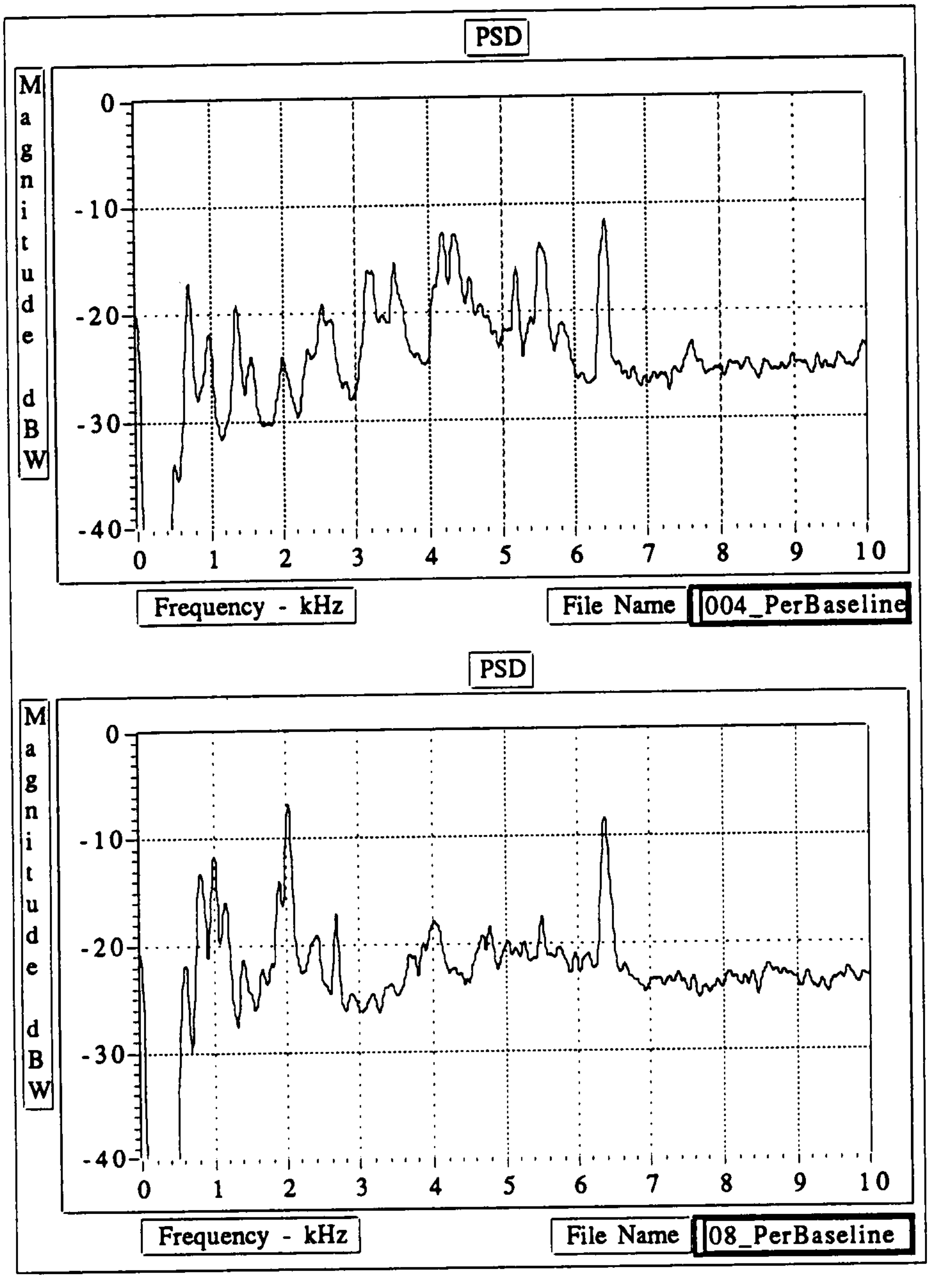


Figure 2.2. Comparison of Typical Power Spectra of Sensor 4 (Upper Graph) and Sensor 8 (Lower Graph).

velocity of the jet. Second, the noise generated by the jet has a peak amplitude in the power spectrum, yet because it is random it contributes power over the entire spectrum. Last, the noise radiated from the jet has direction and thus is not uniform.

Acoustic power is generated by the conversion of some of the mechanical energy inherent in the leak-jet. This leads to a mathematical expression for the sound power level of a jet measured in decibels (dB),

$$L_w = 10 \log_{10}\left(\frac{1}{2}\, \dot{m}\, V^2 \xi\, 10^{12}\right) , \tag{2.1}$$

where $\dot{m}$ is the mass flow rate, V is the exit velocity of the jet, and ξ is the efficiency constant representing the mechanical to acoustic power conversion. The constant 10^{12} is a scale factor for conversion to the logarithmic sound scale. The above equation shows the proportionality between leak power and mechanical power, $0.5\dot{m}\, V^2$, in watts.

The mathematical expression Stulen presents for the leak noise power's center, or peak frequency, f_p, is especially useful. It relates V to the leak hole diameter, D, as follows:

$$f_p = \frac{S_t\, V}{D} , \tag{2.2}$$

where S_t is the Strouhal number (ratio of inertial and vibrational forces found in most cases to be a constant value of 0.2). This mathematical expression underscores what may be intuitive: that the peak frequency of the leak gets lower as the size of the hole increases (assuming a reasonably constant pressure).

Since the leak is random, all of its power is not centered at f_p, but falls off around this peak sharply when the leak is small, and more gradually as the hole diameter grows. Likewise, it is not uniformly radiated; rather, it has direction. Experimental data suggest that most of the sound pressure radiates from the leak at an angle between 30 to 45 degrees from the horizontal leak axis [5].

The leak's airborne vibrational power is attenuated, or muffled, dramatically as its distance from the source gets larger. This is due to classical absorption of the sound energy by the furnace gas, as well as the refraction of the sound waves as they travel through the flyash and turbulent environment of the boiler.

B. Consequences of Tube Leaks

An EPRI journal article [6] reported a number of significant facts about boiler tube leaks: the leading cause of availability loss at U.S. power plants are boiler tube leaks, accounting for six percent of availability loss. Failures occurred in all significant boiler areas including economizers, waterwalls, superheaters, and reheaters. More than 80 percent of the reported tube leaks resulted in a forced shutdown of the boiler, where a typical down-time for repair lasting three days could cost a utility $750,000 for replacement power on a 400 MW or larger unit.

Without acoustic leak detection systems, operators must monitor make-up water rate directly, observe the fluctuation in furnace pressure, and listen for obvious noise during furnace operation. The problem with these methods are that they do not provide early detection, can only detect moderate to large leaks, and cannot precisely locate the leak [7]. The desire to reduce outage time for repair and therefore limit the cost associated with tube leaks led EPRI to research the apparent success certain European utilities were experiencing in early leak detection.

C. Leak Detection Systems

Most acoustic leak detection systems place pressure sensitive sensors at strategic locations throughout the boiler. Each sensor is installed in the back of a metal waveguide which is mounted directly on the boiler wall. These sensors capture the airborne soundwaves of boiler operation. Each sensor should have a signature, or average baseline mean-square value during normal operation. When a leak develops near the sensor, the mean-square value should exceed a threshold value set just comfortably above the baseline of that sensor.

In order to improve detection performance, most systems use bandpass filtering to isolate the power spectrum to between 1,500 Hz and 10,000 Hz. This enables the system to ignore the combustion noise, usually centered around a peak frequency (f_p) of 750 Hz, and any high-frequency transients.

These simple systems have enjoyed marked success since their introduction to the U.S. market in 1984 [8]. In most cases, where accurate records have been kept, acoustic monitoring systems can detect pinhole leaks as much as two to three days before regular methods, and tube ruptures as much as several hours before. The earlier the detection, the better the chance is in limiting impingement of adjacent tubes. This impingement, sometimes called "washing", is especially destructive since the superheated jet of steam and water can so degrade the integrity of other surrounding tubes that the impinged tubes are discovered and repaired while the original pinhole leak remains undetected [6].

Snow and Ball of the CEGB described the development of acoustic leak detection systems in the United Kingdom [7]. First installed in 1974, they have undergone much evolution but remain basically the same in underlying technique.

The instrument's sensor is mounted peripherally in a probe tube called a waveguide. Probes must be just long enough to reduce the heat the sensor is exposed to but not compromise the integrity of the sound waves entering it. The horn shape of the waveguide reduces the standing waves and enhances the signal in relation to structural or external noise.

Some placements of probes in the boiler, such as in the economizer region, provide a sufficient vacuum to keep the guides free of dust and fly-ash. Other areas supply an unreliable vacuum, so regular sensor maintenance includes periodic clearing of the waveguides.

The typical acoustic boiler leak detection system begins with the aforementioned sensor, which connects to a local preamplifier. Each sensor has a detection range of approximately 30 feet. The preamplifier filters and amplifies the signal for transmission to the signal processing unit located in a central room. One signal processing unit usually contains individual channels for 10 to 12 sensors located at trouble spots in the boiler.

The signal processing usually amounts to further bandpass filtering of the signal followed by a root-mean-square (RMS) voltage computation. This RMS value is compared to a threshold level experimentally calibrated for each sensor channel. An alarm will alert the operator should the value exceed this threshold for a given period of time. The time lag before alarm is to protect the operator from false alarms due to intermittent soot-blower noise or bangs and pops in combustion noise [9].

Another boiler leak detection system involves acoustic emission (AE), which monitors structure-borne sound; whereas, acoustic leak detection systems monitor air-borne sound. The AE system uses sensors made with piezoelectric elements mounted on an acoustic probe attached to the boiler wall [10]. Signal processing of the AE signal also involves calculating the RMS values of each sensor and comparing to individually preset thresholds [5].

D. Conclusions

Research indicated that little documentation existed that showed actual spectra of sensor channels of an acoustic leak detection system. All the leak detection systems studied processed acoustic signals in the time-domain; therefore, most graphical representations of tube leaks were time-domain plots. A fundamental assumption of this project was that a small, slowly expanding tube leak should have a distinguishable spectral signature a significant period of time before it becomes apparent in the time-domain.

Since little was known about a typical acoustic channel power spectrum, finding an appropriate and optimum leak detection algorithm for the BAM project posed special challenges for software application development. It was anticipated that initial leak detection algorithms would be constantly modified and often restructured. A VPL was chosen, therefore, to ease the burden and frustration of developing and modifying programs that so often accompanies programming in traditional text-based languages such as FORTRAN or C.

LabVIEW is a VPL developed by National Instruments that provides a graphical programming environment with a library of data acquisition and DSP analysis functions for use in application development [11]. A LabVIEW application program is called a Virtual Instrument (VI), and consists of a front panel, block diagram, and icon/connector. The front panel is a GUI with a library of knobs, slides, switches, graphs, and strip charts. These controls and indicators enable the programmer to develop custom GUI's that mimic the time proven interface of hardware instrumentation.

Each control or indicator on the front panel can be sized and positioned, and has individual options. For instance, graphs can have legends where unique line styles and colors can be selected via a menu palette for single or overlaid plots. Axis values on a graph can be dynamically changed to view small or large blocks of plotted data. Controls for any data type can be constructed and specified for default value, range, and dimension [12].

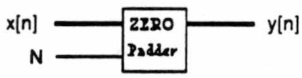
Engineers and scientists often initially develop programs by sketching a block diagram; however, the ideas expressed in the block diagrams inevitably must be converted into cryptic code. With LabVIEW, the block diagram is the program. The programmer passes data between block functions and terminals with graphical wires. Terminals represent front panel controls or indicators and appear on the block diagram to route raw or processed data to and from the GUI. Each function is represented as a graphically unique icon; therefore, a LabVIEW program is self-documenting. This feature enables rapid initial development of VIs, and relatively easy modification and adaptation of existing VIs. In fact, the programmer can develop custom VIs for use as subVIs in the block diagram of a higher order VI. SubVIs are analogous to subroutines in conventional text-based languages [12].

The block diagram program is data driven. That is, a function is executed only when all the inputs are available at the particular function's node, then the function's outputs are generated. This allows for simultaneous execution of instructions, a type of parallel processing [13]. Of course, execution order must be enforced in some instances. Thus LabVIEW is equipped with familiar programming structures such as For Loops, While Loops, and Case Statements that force sequential order, repetitive operations, and branching routines. These structures appear as graphical borders that contain the particular icons whose execution must be controlled [12].

Figure 2.3 shows the icon/connector, front panel, and block diagram of the "Padder2" VI developed in LabVIEW for this project. This VI was designed to be used as a SubVI; it appends a specified number of zeros to an array for use in FFT techniques in a calling VI. The connector pane at the top of the figure illustrates how "Padder2" would connect to the block diagram of a calling VI. The front panel in the middle of the figure shows the two inputs, $x[n]$ and N, and the output, $y[n]$. N is the array size the user desires $y[n]$ to be after execution.

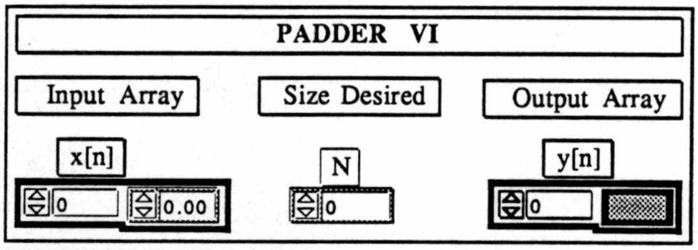
Reading from left to right in the block diagram at the bottom of the figure, $x[n]$ is passed into the graphic labeled "Case Structure". It is also passed into the icon labeled "Array Size". The size of $x[n]$ is compared to N by the block icon "=". If N is equal to the size of $x[n]$, the block icon "=" passes a logical "true" to the case structure control, "?", otherwise it passes a logical "false". If the case selected is "false",

Connector Pane



PADDER2

Front Panel



Block Diagram

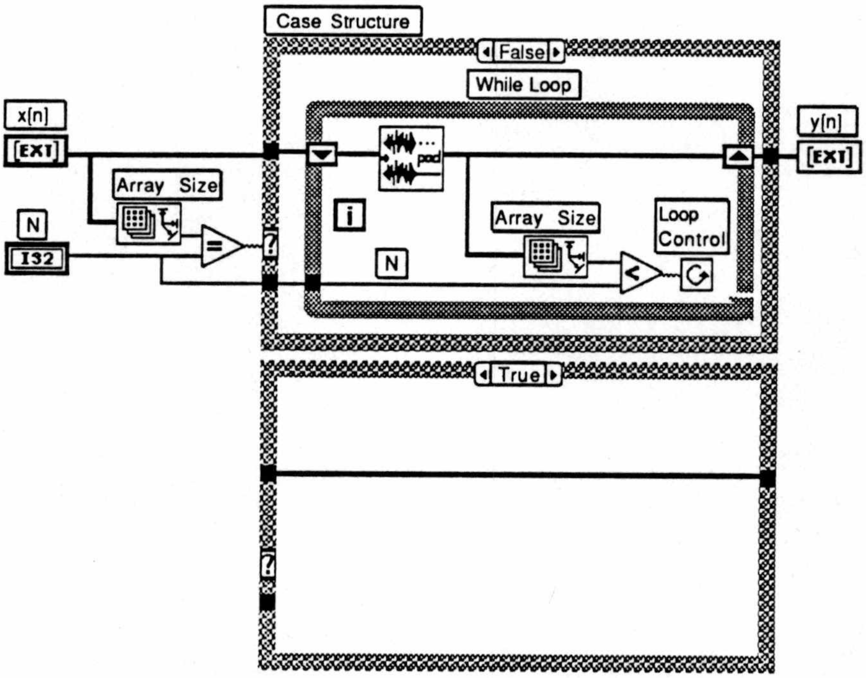


Figure 2.3. Connector Pane, Front Panel, and Block Diagram of Padder2 VI.

then zeros are appended to $x[n]$ utilizing the graphical structure labeled "While Loop"; however, if the case selected is "true", then $x[n]$ is passed directly through to $y[n]$ as the normally hidden "true" case structure indicates. The while loop continually pads $x[n]$ with zeros to the next power of two in the block labeled "pad". Each time it is padded, the resulting array size is compared to N to determine if the while loop should continue executing. When the block icon "<" passes a logical "false" to the icon labeled "Loop Control", the loop stops executing and the resulting padded array is passed to $y[n]$.

The acoustic monitoring systems studied employed hardware-intensive, time-domain signal processing methods to detect boiler tube leaks. The versatility and power afforded by LabVIEW enabled this project to analyze and develop software-intensive, frequency domain DSP methods for detecting the spectral signature of a boiler tube leak. The next two sections review the theory of both classical and parametric DSP techniques utilized for this project.

III. CLASSICAL POWER SPECTRAL DENSITY ESTIMATION

The PSD of a random signal relates how the power of that signal is distributed along the frequency domain axis. However, to precisely represent a random signal would require an infinite-length data-set set of observations ranging over a continuum of time values. This, of course, is not practical. Time demands and processing capabilities limit any characterization to one of estimation using finite length records. The correlogram method and periodogram method are the two principle approaches that have been used over the years to reliably estimate the PSD. They have come to be known as the classical methods.

A. Correlogram Methods

It has been shown [14-18] that the PSD function, $P_{xx}(f)$, can be defined as the Discrete-Time Fourier Transform (DFT) of the autocorrelation sequence:

$$P_{xx}(f) = T \sum_{m=-\infty}^{\infty} R_{xx}[m]\exp(-j2\pi m \frac{f}{f_s}) \quad , \quad \frac{-f_s}{2} < f < \frac{f_s}{2} \tag{3.1}$$

where T is the time between samples, f is frequency, f_s is $1/T$, m is the time-lag index, and the autocorrelation, $R_{xx}[m]$, is defined as the limit of a time average

$$R_{xx}[m] = \lim_{M\to\infty}\left\{\left(\frac{1}{2M+1}\right)\sum_{n=-M}^{M} x[n+m]x^*[n]\right\} \tag{3.2}$$

where $x[n]$ is the sampled random signal in the time domain and $x^*[n]$ is its complex conjugate.

Since an infinite data record is not available, a precise autocorrelation cannot be realized. However, Blackman and Tukey published a treatise in 1958 [20] that described in detail a practical procedure for estimating the PSD using autocorrelation estimates of finite time sequences. Their procedure became known as the Blackman-Tukey method for PSD estimation. As their procedure spawned similar procedures, the whole class of procedures based on autocorrelation or cross-correlation estimates became known as correlogram methods of PSD estimation.

The basic correlogram method starts with acquiring a reliable estimate of the autocorrelation. Then the estimate data is smoothed by multiplying it by a window function. This windowed data is then transformed into the frequency domain via a Fourier Transform.

If the random signal $x[n]$ is truncated to N samples indexed from n=0 to n=N-1, then the autocorrelation estimate, $\hat{R}_{xx}[m]$, can be written

$$\hat{R}_{xx}[m] = \begin{cases} \frac{1}{N-|m|} \sum_{n=0}^{N-|m|-1} x^*[n+|m|]x[n] & , \ 0 \le |m| \le N \\ 0 & , \text{ elsewhere} . \end{cases} \quad (3.3)$$

The mean of this autocorrelation estimate can be shown to be the true autocorrelation [14]:

$$\begin{aligned} \mathcal{E}\{\hat{R}_{xx}[m]\} &= \frac{1}{N-|m|} \sum_{n=0}^{N-|m|-1} \mathcal{E}\{x^*[n+|m|]x[n]\} \\ &= R_{xx}[m] , \end{aligned} \quad (3.4)$$

where $\mathcal{E}\{\}$ is the expectation operation. However, Jenkins and Watts [17] have shown that the variance of the autocorrelation estimate of a Gaussian process as a function of the number of data samples is approximately

$$\text{var}\{\hat{R}_{xx}[m]\} = \frac{N}{N-|m|} \sum_{n=-\infty}^{\infty} \left(R_{xx}^2[m] + R_{xx}[k+m]R_{xx}[k-m]\right), \quad m << N \quad (3.5)$$

This formula shows that the variance increases as the lag index increases.

This increase in variance of $\hat{R}_{xx}[m]$ as m approaches N becomes obvious with study of Equation 3.3. If N=100 samples of $x[n]$ are observed, and the unobserved data is assumed zero, then $\hat{R}_{xx}[m=0]$ consists of 100 sums. $\hat{R}_{xx}[m=85]$, however, consists of only N-m=15 sums. Correlation estimates, therefore, are more uncertain for higher lags since there are fewer samples left to average. Blackman and Tukey suggested a maximum lag index $L \leq (N/10)$ to avoid this instability for higher lags. They also employed a window function to reduce bias and leakage when the autocorrelation estimate is transformed via a Fourier Transform.

The window function, $w[m]$, is a multiplicative weighting function applied to $\hat{R}_{xx}[m]$ to reduce the spectral leakage that occurs when transforming truncated data sets. The window is a $(2L+1)$-point sequence that spans the interval $-L \leq m \leq L$ and is normalized so that $w[0]=1$. Thus the PSD estimate will be scaled properly.

When $\hat{R}_{xx}[m]$ is truncated to L samples, it implies $\hat{R}_{xx}[m]=0$ for $L<m<N$. The implied window which multiplies $\hat{R}_{xx}[m]$ is a rectangular one where

$$w_r[m] = \begin{cases} 1 \ , & -L \leq m \leq L \\ 0 \ , & \text{elsewhere} \ . \end{cases} \tag{3.6}$$

This creates a sharp discontinuity between $\hat{R}_{xx}[L]$ and $\hat{R}_{xx}[L+1]$. After the truncated $\hat{R}_{xx}[m]$ is transformed, these discontinuities cause a distorted spectrum.

The cause of this distortion can be seen mathematically by considering the PSD estimate, $\hat{P}_{xx}[f]$, and calculating its mean value

$$
\begin{aligned}
\mathfrak{E}\left\{\hat{P}_{xx}[m]\right\} &= T \sum_{m=-N}^{N} w[m]\mathfrak{E}\left\{\hat{R}_{xx}[m]\right\}\exp(-j2\pi m\frac{f}{f_s}) \ , \\
&= T \sum_{m=-L}^{L} R_{xx}[m]\exp(-j2\pi m\frac{f}{f_s}) \ , \\
&= W_r[f] * P_{xx}[f] \ , \qquad (3.7)
\end{aligned}
$$

where * indicates convolution. Thus the mean value of the PSD estimate can be understood as the convolution of the true PSD with the transform of the rectangular window function, known as the digital sinc or Dirichlet kernel. Its transform is

$$
W_r[f] = T\exp(-j2\pi ft[L-1])\frac{\sin(\pi fTL)}{\sin(\pi fT)} \ . \qquad (3.8)
$$

When convolved with $P_{xx}(f)$, the relatively high sidelobes of $W_r[f]$ will appear as smaller tones in $\hat{P}_{xx}(f)$ next to an adjacent larger tone. This distortion of the magnitude of adjacent frequencies can cause an actual, smaller adjacent tone in $P_{xx}(f)$ to be masked in $\hat{P}_{xx}(f)$. Convolution also causes a loss in resolution, or smearing, in $\hat{P}_{xx}(f)$ due to the width of the mainlobe of $W_r[f]$. This can result in two distinct, adjacent tones being smeared together and appearing as one single tone (see Figure 3.1).

The perfect window would obviously be one whose transform is the delta function. It has an infinitely small mainlobe width and no sidelobes; therefore, any function convolved with it remains effectively unchanged. Its realization is impossible, of course, but it

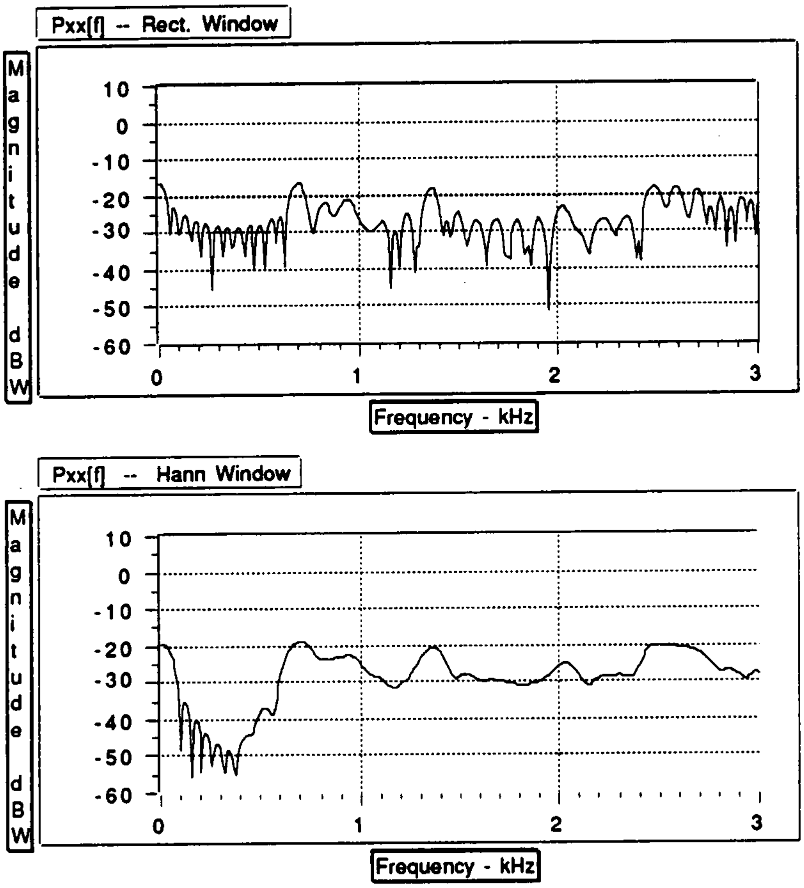


Figure 3.1. Smearing of Adjacent Sinusoidal Tones.

establishes a picture of the ideal window by which any other window's effectiveness can be judged pictorially. When compared with the delta function, the rectangular window's transform compares well in resolution since its mainlobe width is relatively narrow; however, it fails with regard to leakage suppression because of its high sidelobes. Choosing a window demonstrates the classic tradeoff between stability and resolution. A window which is more effective at leakage suppression (i.e., lower sidelobes) would result in lower resolution (i.e., wider mainlobe). Conversely, a window which would achieve better resolution (i.e., narrower mainlobe) is less effective at leakage suppression (i.e., higher sidelobes).

A window function whose transform has low sidelobes and a wide mainlobe is one whose time domain representation has rounded edges and slopes that dissolve gradually to 0 at $w[L]$ and $w[-L]$. When it then multiplies the truncated sequence, it smooths the discontinuity between $\hat{R}_{xx}[L]$ and $\hat{R}_{xx}[L+1]$. This results in a smoother, more stable, and less biased spectrum, but with a corresponding loss of resolution due to the width of the transform's mainlobe.

The Blackman-Tukey correlogram PSD estimator, $\hat{P}_{BT}[f]$, is defined as

$$\hat{P}_{BT}(f) = T \sum_{m=-L}^{L} w[m]\hat{R}_{xx}[m]\exp(-j2\pi m \frac{f}{f_s}) \quad , \quad \frac{-f_s}{2} < f < \frac{f_s}{2} \tag{3.9}$$

where the choice of $w[m]$ is selected by the user. There are two principle places where the user can exert control over the tradeoffs among resolution, stability, and leakage suppression. One is the by varying the maximum lag indice L, and the other is by choice of the window function, $w[m]$.

The relationship between the choice of the maximum lag index L and the window function $w[m]$, and their corresponding effect on resolution and stability can be clearly seen upon examination of the statistical time-bandwidth product (STBP). The STBP for random signals is a modified version of the time-bandwidth product (TBP) for deterministic signals.

It has been shown [14] that the TBP is $T_e B_e = 1$, where T_e is the equivalent time width in seconds and B_e is the equivalent bandwidth in Hertz (Hz). In general, T_e is taken to be the data record interval (i.e., $T_e = NT$), so that the energy spectral density of a finite length deterministic signal has a resolution no better than $B_e = 1/(NT)$ Hz. However, the STBP must take into account the fluctuations associated with random signals. Therefore, a formula for determining the relative stability of the PSD estimate is the quality ratio Q, defined as the ratio of the variance of the PSD estimate to its squared mean

$$Q = \frac{\mathrm{var}\{\hat{P}_{xx}[f]\}}{\left(\mathcal{E}\{\hat{P}_{xx}[f]\}\right)^2} \quad . \tag{3.10}$$

Values of Q << 1 would represent small variations from the mean, meaning a relatively smooth and stable PSD estimate; whereas, values of Q >> 1 would represent large variations from the mean, meaning a unstable PSD estimate. As L approaches the total data record length N, the variance will increase yielding unstable and unreliable estimates.

Instead of B_e, the STBP uses the effective statistical bandwidth B_s [20] as a rough measure of resolution. B_s is defined as

$$B_s = \frac{\left[\int_{-\frac{1}{2T}}^{\frac{1}{2T}} W(f)df\right]^2}{\int_{-\frac{1}{2T}}^{\frac{1}{2T}} W^2(f)df} \tag{3.11}$$

where $W(f)$ is the frequency domain representation of the window function $w[m]$. $W(f)$ has alternating positive and negative sidelobes, while $W^2(f)$ has only positive values (see Figure 3.2). Integrating $W(f)$ would result in the alternating positive and negative sidelobes cancelling each other, while integrating $W^2(f)$ would result in them being added together. Thus B_s can be regarded as the area of the window function's mainlobe divided by its total area (mainlobe and sidelobes). Therefore, resolution increases (B_s becomes small) if $W(f)$ has a narrow mainlobe and high sidelobes, while it decreases (B_s becomes large) if $W(f)$ has a wide mainlobe and low sidelobes.

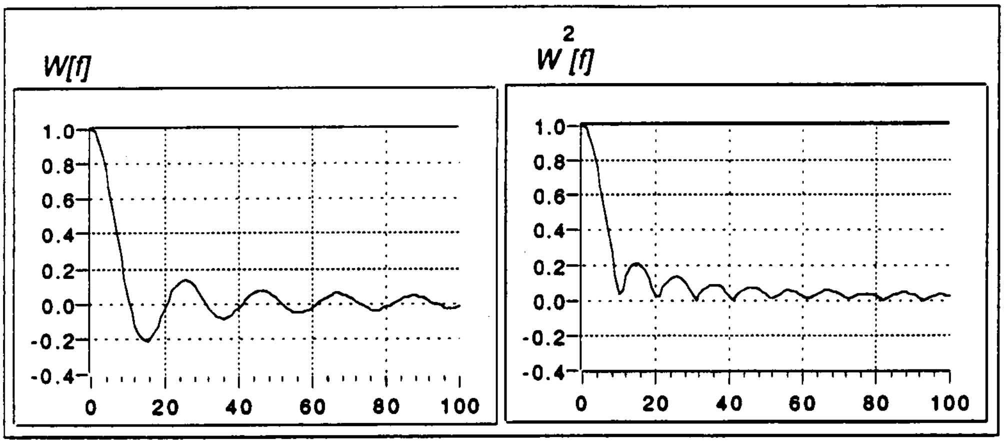


Figure 3.2. Comparison of $W(f)$ and $W^2(f)$.

The product of Q, B_s, and T_e is known as the STBP, where

$$QB_sT_e \geq 1 \qquad (3.12)$$

for most Gaussian processes. The tradeoff between resolution and stability can easily be demonstrated by this equation. It is not possible to decrease the value of Q (increase stability) without correspondingly increasing the value of B_s (decrease resolution). Likewise, it is not possible to increase the value of Q without decreasing B_s.

It is possible to increase L and not simultaneously increase the variance, but only by using a biased autocorrelation function, $\ddot{\hat{R}}_{xx}[m]$, defined as

$$\ddot{\hat{R}}_{xx}[m] = \frac{1}{N}\sum_{n=0}^{L} \hat{x}^*[n + |m|]\hat{x}[n] \tag{3.13}$$

where

$$\ddot{\hat{R}}_{xx}[m] = \frac{N-|m|}{N}\hat{R}_{xx}[m] \ . \tag{3.14}$$

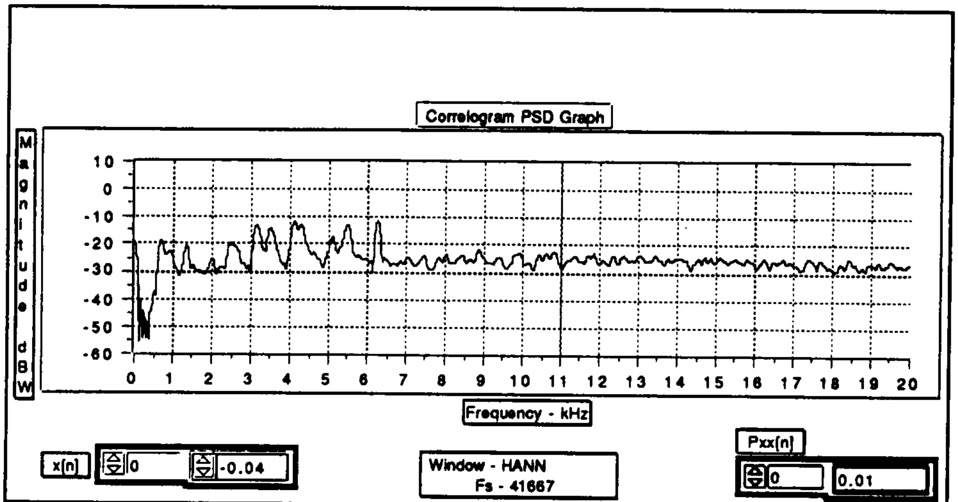
Using $\ddot{\hat{R}}_{xx}[m]$ to determine the PSD estimate means $\hat{R}_{xx}[m]$ is in effect being multiplied by a triangular window. This window is not very effective at leakage suppression, but using $\ddot{\hat{R}}_{xx}[m]$ does insure that $\ddot{\hat{R}}_{xx}[0] \geq \ddot{\hat{R}}_{xx}[m]$, a condition required for autocorrelation functions that is not always met by the unbiased $\hat{R}_{xx}[m]$. Since $\ddot{\hat{R}}_{xx}[0] = \hat{R}_{xx}[0]$, the power is preserved in the measured signal.

Figure 3.3 shows the connector pane and block diagram of the correlogram PSD estimator VI created in LabVIEW 2 to analyze the boiler sensor data for this project. The connector pane at the top of the figure shows how this VI (or subroutine) would be connected (or which parameters should be passed) in the block diagram (or program) of another VI (or calling program). The block diagram at the bottom of the figure is used to calculate the biased correlogram PSD estimator, $\ddot{\hat{P}}_{xx}[f]$. Reading from left to right on the block diagram, data is passed as an extended precision floating point array ([EXT]) into the VI's input node labeled "$x[n]$". The array $x[n]$ is then passed to the block labeled "AutoCorrelation", which is a VI provided by LabVIEW to calculate the autocorrelation. This autocorrelation

Connector Pane

CORRELPSDVI

Front Panel



Block Diagram

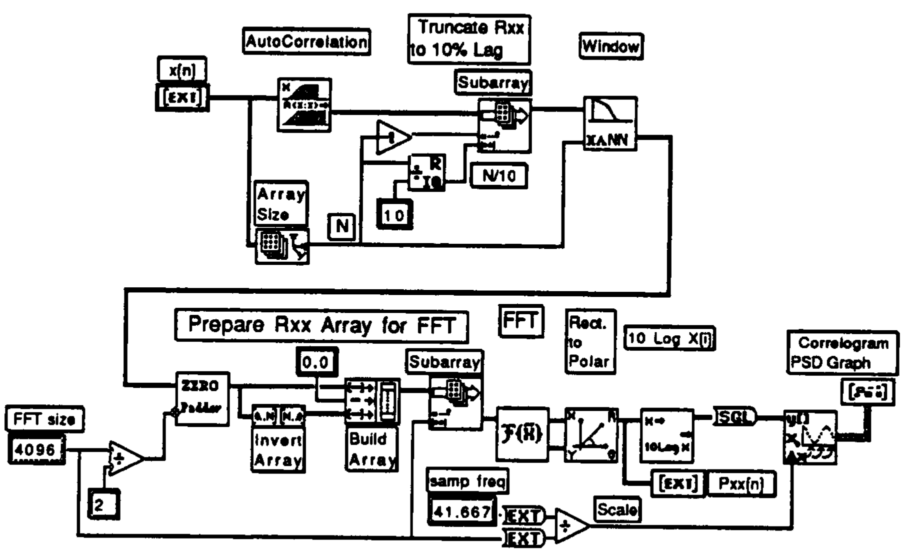


Figure 3.3. Connector Pane, Front Panel, and Block Diagram of Correlogram PSD VI.

array, $\hat{R}_{xx}[m]$, is then truncated to ten percent of N. Next, under the text "Window", $\hat{R}_{xx}[m]$ is multiplied by a Hann window for smoothing. Then, under the text "Prepare R_{xx} Array for FFT", it is padded with zeros to an array size of 2048, then appended with a mirror image of itself (since $R^*_{xx}[m] = R_{xx}[-m]$). In the remaining blocks a Fast Fourier Transform (FFT) is performed on $(w[m]\ddot{\hat{R}}_{xx}[m])$ to yield $\hat{P}_{BT}[f]$, which is then converted to polar coordinates, then to decibels, and finally plotted on the front panel graph labeled "Correlogram PSD Graph".

B. Periodogram Methods

The correlogram methods discussed earlier are called "indirect" methods since they obtain the PSD estimate indirectly from the data set via an autocorrelation estimate. However, the periodogram methods are called "direct" methods since they use an alternate definition of the PSD that obtains the estimate by operating directly on the data set.

This alternate definition of the PSD is based on ergodicity. If the random process being measured is ergodic, then the PSD can be defined

$$P_{xx}[f] = \lim_{N\to\infty} \mathrm{E}\left\{ \frac{1}{(2N+1)T} \left| T \sum_{n=-N}^{N} x[n]\exp(-j2\pi n \frac{f}{f_s}) \right|^2 \right\} , \frac{f_s}{2} < f < \frac{f_s}{2} , \qquad (3.15)$$

where $\mathfrak{E}\{\}$ is the expectation operation. Since an infinite data set would be impractical, a periodogram PSD estimate has the form

$$\hat{P}_{xx}[f] = \mathfrak{E}\left\{ \frac{T}{N} \left| \sum_{n=0}^{N-1} x[n]\exp(-j2\pi fnT) \right|^2 \right\} \ , \ -\frac{f_s}{2} < f < \frac{f_s}{2} \ . \qquad (3.16)$$

If the expectation operation is ignored, this definition will yield only unstable PSD estimates [1-3]. However, if the definition is modified to average several estimates of $\hat{P}_{xx}[f]$, then a stable, smooth PSD estimate can result.

Welch [19] developed a periodogram PSD estimation method that averages several spectra by dividing the total, finite data array into overlapping segments. Each segment is multiplied by a window function (see Figure 3.4) and then transformed to the frequency domain via a Fourier Transform. The squared magnitude of each transformed segment is calculated, after which all of the transformed segments are averaged together to yield the final PSD estimate.

If $x[n]$ is truncated to N samples, indexed from n=0 to n=(N-1), then it is possible to divide $x[n]$ into $NSEG$ segments of $NSAMP$ samples each, with a shift of $NSHIFT$ samples between segments that controls the amount of overlap ($NSAMP$-$NSHIFT$). $NSEG$ is calculated by

$$NSEG = \mathrm{INT}\left(\frac{N-NSAMP}{NSHIFT}\right) + 1 \qquad (3.17)$$

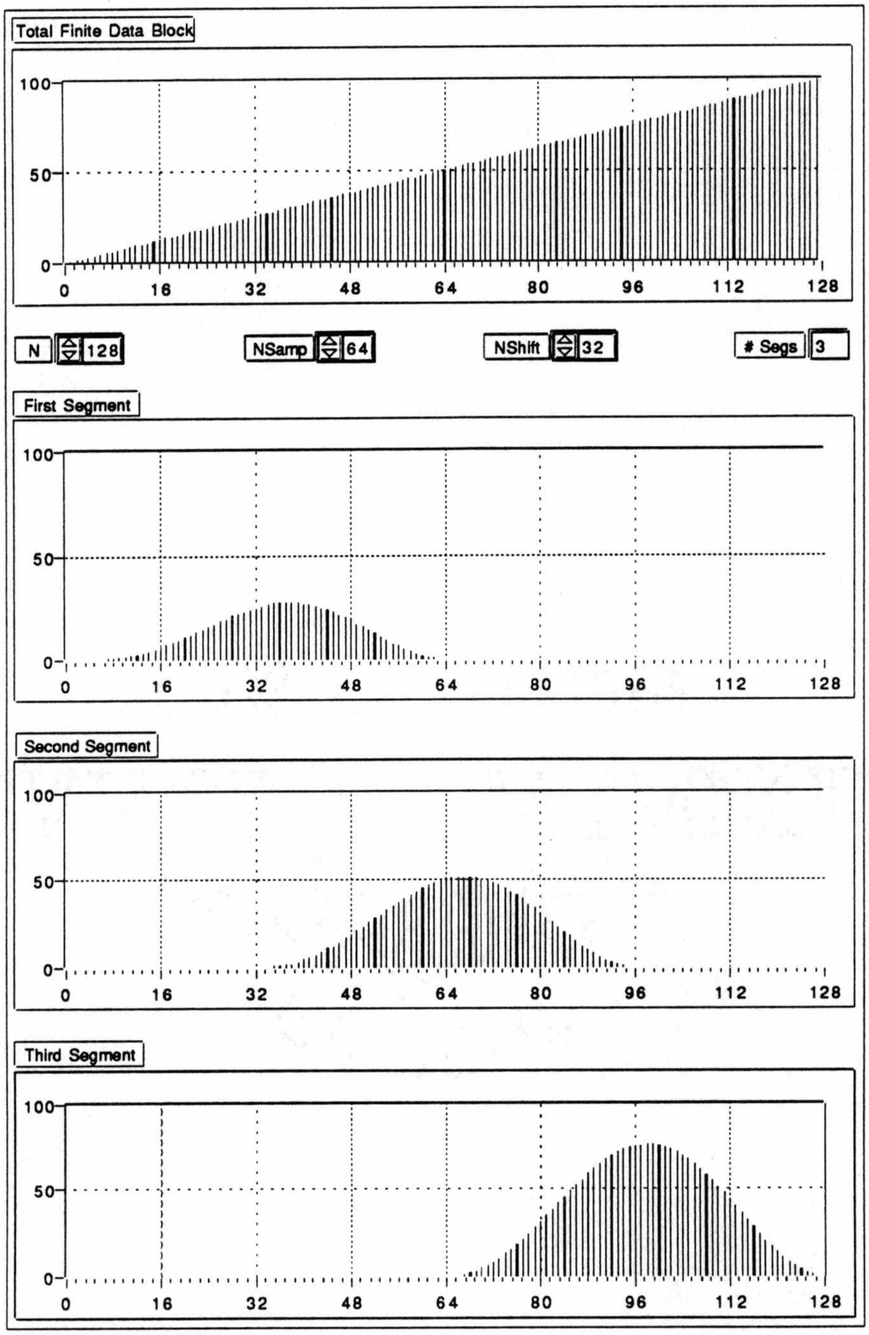


Figure 3.4. Windowed, Overlapping Segments.

where INT{} is the integer operator that truncates any decimal remainder. The k-th segment is defined as

$$x^{(k)}[n] = x[n + k(NSHIFT)] \tag{3.18}$$

where $0 \le n \le (NSAMP\text{-}1)$ and $0 \le k \le (NSEG\text{-}1)$. For instance, consider a block of data of length N=128 shown in Figure 3.4. If the segment size is $NSAMP$=64 and the amount of shift is $NSHIFT$=32, then the number of segments calculated is $NSEG$=3.

The sample spectrum of the windowed k-th segment will be

$$\hat{P}_{xx}^{(k)} = \frac{1}{Uf_s NSAMP}\left|\sum_{n=0}^{NSAMP\text{-}1} w[n]x^{(k)}[n]\exp(\text{-j}2\pi fnT)\right|^2, \quad \text{-}\frac{f_s}{2} < f < \frac{f_s}{2}, \tag{3.19}$$

where U is the discrete-time window energy

$$U = \frac{1}{NSAMP}\sum_{n=0}^{NSAMP\text{-}1} w^2[n] \quad . \tag{3.20}$$

Dividing by U in Equation 3.19 removes the effect of the window energy bias in each segment estimate. The average of the windowed segment periodograms is the Welch periodogram PSD estimate,

$$\hat{P}_W[f] = \frac{1}{NSEG}\sum_{n=0}^{NSEG\text{-}1} \hat{P}_{xx}^{(k)}[f] \quad . \tag{3.21}$$

The mean of this PSD estimate can be shown [14] to be the convolution of the true PSD with the squared magnitude of the window function's transform divided by the window's energy correction factor U ,

$$\mathbf{E}\{\hat{P}_W[f]\} = \frac{1}{P}\sum_{k=0}^{P-1} \mathbf{E}\{\hat{P}_{xx}^{(k)}[f]\}$$

$$= P_{xx}[f] * \frac{|W[f]|^2}{U} \quad . \qquad (3.22)$$

Welch suggested [19] using 50% overlap of the segments in order to give equal weighting to all of the samples. For instance, every sample that is weighted heavily by the window in one segment, will receive less weight in the next segment, except the $NSAMP/2$ samples on each end of the original N-point sequence. The variance of the Welch periodogram is given [14] as

$$\mathrm{var}\{\hat{P}_W[f]\} \propto \frac{(P_{xx}[f])^2}{NSEG} \quad . \qquad (3.23)$$

Thus the variance decreases as the number of segments increase.

The resolution and stability of the periodogram PSD estimates are controlled primarily by the user's selection of both the segment length $NSAMP$, and the amount of shift between segments $NSHIFT$. As $NSAMP$ and $NSHIFT$ approach the total record length N, the resolution of the PSD estimate increases, but the variance and instability also increase. As $NSAMP$ and $NSHIFT$ decrease, stability increases, but at the price of decreased resolution. As discussed previously in reference to the correlogram methods, window selection plays a part in leakage suppression with a corresponding effect in resolution.

Figure 3.5 shows the connector pane, front panel, and block diagram used to calculate the periodogram PSD estimator, $\hat{P}_W[f]$. Reading from left to right on the block diagram, data is passed in as an extended precision floating point array ([EXT]) into the VI's input node labeled "$x[n]$". The selected segment length $NSAMP$, and the selected shift between segments $NSHIFT$, are also passed to their corresponding nodes on the block diagram. Variables $x[n]$, $NSAMP$, and $NSHIFT$ are then passed through the structure labeled "PERIODOGRAM" to the VI labeled "Segment Shifter". The structure is the LabVIEW "while loop". This loop executes all of the functions contained within its boundaries once, then re-executes them again and again until the block under the text "Loop Condition" is passed a false boolean value. The box labeled "i" is an integer counter that counts each loop execution starting from i=0. The number of times this loop is executed is controlled by comparing the number of segments, $NSEG$, to the value of i. Thus the loop is executed $NSEG$ times. After calculating the number of segments $NSEG$, the Segment Shifter VI divides $x[n]$ into $NSEG$ segments by the Segment Shifter VI. Each segment is passed through the Hann window VI, padded with zeros to length 2048 by the Zero Padder VI, and transformed to

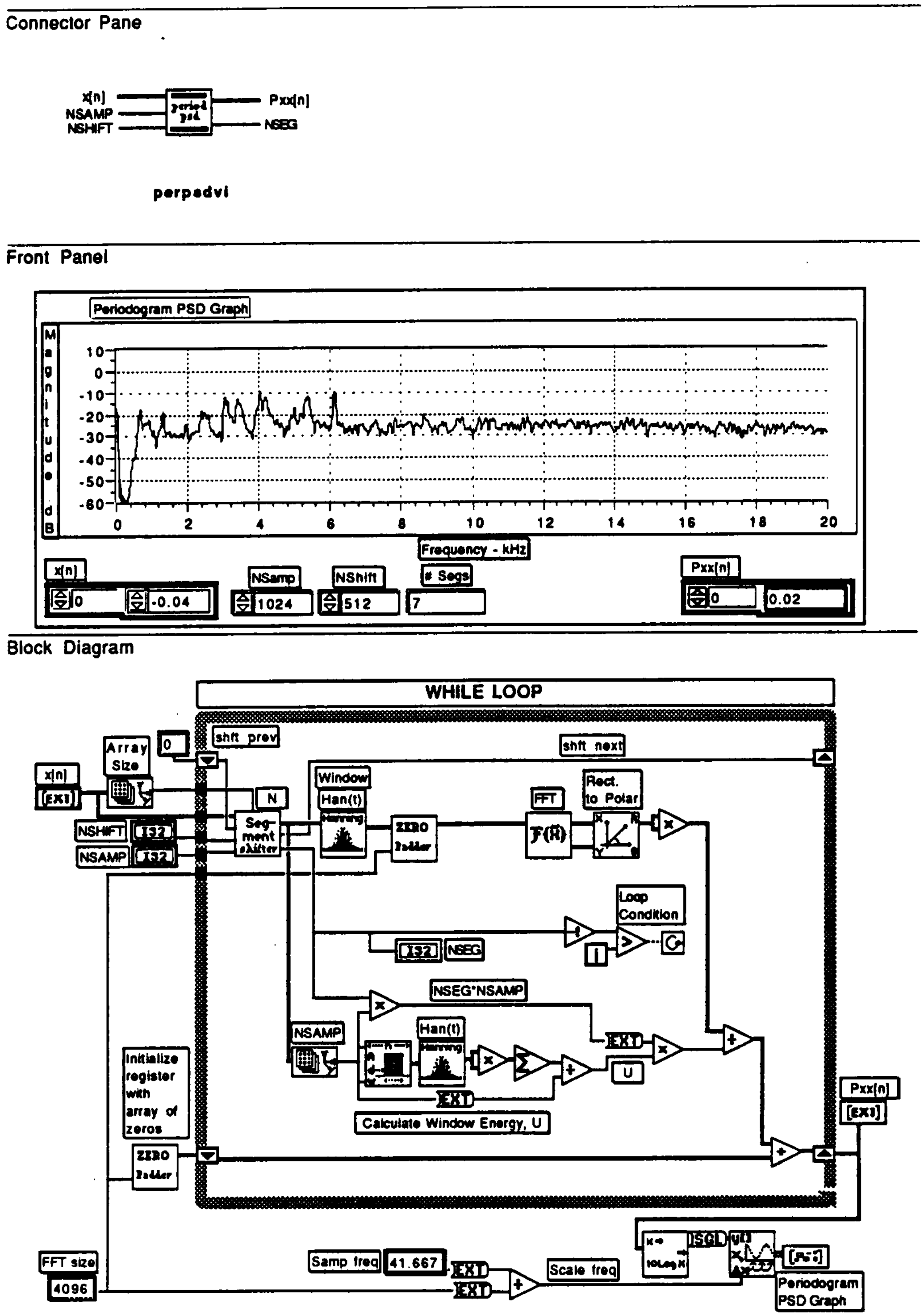


Figure 3.5. Connector Pane, Front Panel, and Block Diagram of Periodogram PSD VI.

the frequency domain via the FFT VI. Next, the magnitude is squared via the Rectangular to Polar VI and multiplier function, then divided by the energy correction factor (*U NSEG NSAMP*). At the end of each loop, each transformed segment is added to all previously transformed segments yielding a PSD estimate at the bottom, right-hand side of the loop structure. This estimate is converted to decibel units and plotted on the front panel graph labeled "Periodogram PSD Graph".

Both of the classical methods described in this section employ a window function to estimate the PSD. Window functions, however, implicitly assume that the data outside the window is zero. Methods will be introduced in the following section that make better assumptions about the data outside the window, and thus can produce much better resolution.

IV. PARAMETRIC POWER SPECTRAL DENSITY ESTIMATION

The principal limitation in calculating the true PSD of a random process is that the signal and its autocorrelation must be truncated to a finite length by a window function. The classical methods of PSD estimation implicitly assume that the data outside the window is zero. This is usually an unrealistic assumption that leads to a distorted PSD estimate. Parametric methods, however, use a time-series model to approximate the process that generated the finite set of data. The parametric PSD estimate then, is a function of the model parameters. These models can make better assumptions about the unobserved data leading to higher resolution and accuracy in the PSD estimate, if an appropriate model is used. The autoregressive (AR), moving average (MA), and autoregressive-moving average (ARMA) process models are essentially filters driven by white noise. Methods will be presented in this section that estimate the filter parameters and white noise variance from a finite data record for each model type.

A. Parametric Random Process Models

If $x[n]$ is a random process, then a time-series model to approximate it is given by

$$x[n] = -\sum_{k=1}^{p} a[k]x[n-k] + \sum_{k=0}^{q} b[k]d[n-k] \tag{4.1}$$

where $a[k]$ and $b[k]$ are parameters of a causal filter $h[k]$, and $d[n]$ is the driving input sequence. The filter $h[k]$ can be found by applying z-transform theory to Equation 4.1:

$$\begin{aligned} X[z] &= -X[z]\sum_{k=1}^{p} a[k]z^{-k} + D[z]\sum_{k=0}^{q} b[k]z^{-k} \\ H[z] &= \frac{X[z]}{D[z]} \\ &= \frac{1+\sum_{k=1}^{q} b[k]z^{-k}}{1+\sum_{k=1}^{p} a[k]z^{-k}} \\ &= \frac{B[z]}{A[z]} \\ h[k] &= Z^{-1}\{H[z]\} \quad , \end{aligned} \tag{4.2}$$

where $Z^{-1}\{\ \}$ denotes the inverse z-transform operation. It has been shown [14] that $P_{xx}[z]$ is related to $P_{dd}[z]$ by

$$\begin{aligned} P_{xx}[z] &= P_{dd}[z]H[z]H^{*}[1/z^{*}] \\ &= P_{dd}[z]\frac{B[z]B^{*}[1/z^{*}]}{A[z]A[1/z^{*}]} \quad . \end{aligned} \tag{4.3}$$

If $d[n]$ is a zero-mean, white noise sequence of variance ρ, then $P_{dd}[z] = \rho$. The ARMA PSD is obtained by substituting $z=\exp(j2\pi fT)$ into Equation 4.3 and scaling by the sample interval T to give,

$$P_{\text{arma}}[f] = T\rho \frac{|B[f]|^2}{|A[f]|^2} \quad , \quad -\frac{1}{2T} \leq f \leq \frac{1}{2T} \quad , \tag{4.4}$$

where $A[f]$ and $B[f]$ are defined as

$$A[f] = 1+\sum_{k=1}^{p} a[k]\exp(j2\pi fT) \tag{4.5}$$

and

$$B[f] = 1+\sum_{k=1}^{q} b[k]\exp(j2\pi fT) \ . \tag{4.6}$$

The PSD estimate can now be obtained by estimating the filter parameters $a[k]$ for k=1 to p, and $b[k]$ for k=1 to q, and by estimating the variance ρ.

If p=0, then all of the AR parameters are zero except $a[0]$=1, and

$$\begin{aligned} P_{\text{arma}}[f] &= T\rho\,|B[f]|^2 \\ &= P_{\text{ma}}[f] \quad , \quad -\frac{1}{2T} \leq f \leq \frac{1}{2T} \ . \end{aligned} \tag{4.7}$$

Likewise, if q=0, then all of the MA parameters are zero except $b[0]$=1, and

$$
\begin{aligned}
P_{\mathrm{arma}}[f] &= \frac{T\rho}{|A[f]|^2} \\
&= P_{\mathrm{ar}}[f] \quad , \quad -\frac{1}{2T} \le f \le \frac{1}{2T} \quad .
\end{aligned} \tag{4.8}
$$

A finite ARMA or MA model can be expressed by an AR model of infinite order. In fact, any filter of one type that has a finite number of parameters can be expressed as another type of infinite order. This is important since many efficient algorithms exist that calculate the AR model, so their efficiency can be exploited while calculating an MA or ARMA model's parameters.

For example, it is desired to approximate an ARMA model of order p AR parameters and order q MA parameters, ARMA(p,q), with an AR model of infinite order, AR(∞). The $g[k]$ parameters of the AR(∞) model can be calculated by letting

$$
\begin{aligned}
H[z] &= \frac{B[z]}{A[z]} \\
&= \frac{1}{G[z]} \quad ,
\end{aligned} \tag{4.9}
$$

where

$$
G[z] = 1 + \sum_{k=0}^{\infty} g[k] z^{-k} \quad . \tag{4.10}
$$

Taking the inverse z-transform of $G[z]$ yields

$$g[n] = \begin{cases} 1 & \text{, for } n=0 \\ -\sum_{k=1}^{q} b[k]c[n-k]+a[n] & \text{, for } 1 \le n \le p \\ -\sum_{k=1}^{q} b[k]c[n-k] & \text{, for } n > p \end{cases} \tag{4.11}$$

with initial conditions $g[-1]=g[-2]=\cdots=g[-q]=0$. If enough parameters $g[k]$ are calculated, the new AR sequence will be a reasonable approximation to the original ARMA(p,q) sequence.

B. Relationship of Model Parameters to the Autocorrelation Sequence

The AR, MA, and ARMA parameters can be derived from the autocorrelation sequence (ACS). If Equation 4.1 is multiplied by $x^*[n-m]$ and the expectation $\mathcal{E}\{\}$ is taken,

$$\begin{aligned} \mathcal{E}\{x[n]x^*[n-m]\} &= -\sum_{k=1}^{p} a[k]\mathcal{E}\{x[n-k]x^*[n-m]\} \\ &\quad + \sum_{k=0}^{q} b[k]\mathcal{E}\{d[n-k]x^*[n-m]\} \end{aligned} \tag{4.12}$$

then the result is

$$R_{xx}[m] = -\sum_{k=1}^{p} a[k]R_{xx}[m-k] + \sum_{k=0}^{q} b[k]R_{dx}[m-k] \ . \tag{4.13}$$

Since $d[n]$ was assumed to be a white noise process, it has been shown [14] that

$$R_{dx}[i] = \begin{cases} 0 & , \text{for } i > 0 \\ \rho & , \text{for } i = 0 \\ \rho\, h^*[-i] & , \text{for } i < 0 \quad . \end{cases} \tag{4.14}$$

Thus, the ACS is related to the ARMA parameters by

$$R_{xx}[m] = \begin{cases} R_{xx}^*[-m] & , \text{for } m < 0 \\ -\sum_{k=1}^{p} a[k]R_{xx}[m-k] + \rho\sum_{k=m}^{q} b[k]h^*[k-m] & , \text{for } 0 \le m \le q \\ -\sum_{k=1}^{p} a[k]R_{xx}[m-k] & , \text{for } m > q \quad . \end{cases} \tag{4.15}$$

If from Equation 4.15, $R_{xx}[m]$ is evaluated by setting $q \neq 0$, then a set of nonlinear equations relates the MA parameters to the ACS. Finding the MA parameters $b[k]$ is therefore often much easier by finding an AR equivalent model and then transforming those parameters into MA parameters.

If from Equation 4.15, $R_{xx}[m]$ is evaluated by setting $q=0$, then a set of linear equations result which relate the AR parameters to the ACS:

$$\begin{pmatrix} R_{xx}[0] & R_{xx}[-1] & \cdots & R_{xx}[-p] \\ R_{xx}[1] & R_{xx}[0] & \cdots & R_{xx}[-p+1] \\ \vdots & \vdots & \ddots & \vdots \\ R_{xx}[p] & R_{xx}[p-1] & \cdots & R_{xx}[0] \end{pmatrix} \begin{pmatrix} 1 \\ a[1] \\ \vdots \\ a[p] \end{pmatrix} = \begin{pmatrix} \rho \\ 0 \\ \vdots \\ 0 \end{pmatrix} . \tag{4.16}$$

If given the autocorrelation sequence for lags 0 to p, then the AR parameters $a[k]$ may be found as a solution to Equation 4.16. The above matrix expression forms what is known as the AR Yule-Walker normal equations. Note from Equation 4.15 that lags $|m|>p$ can be recursively calculated from

$$R_{xx}[m] = - \sum_{k=1}^{p} a[k] R_{xx}[m-k] , \tag{4.17}$$

so that the autocorrelation sequence for any lag m can be calculated. Thus, the following equation holds true:

$$\begin{aligned} P_{\mathrm{ar}}[f] &= \frac{T\rho}{|A[f]|^2} \\ &= T \sum_{m=-\infty}^{\infty} R_{xx}[m] \exp(-j2\pi f k T) \quad . \end{aligned} \tag{4.18}$$

Both Equation 4.17 and Equation 4.18 show that the AR PSD estimator matches the correlogram estimator for lags $m=0$ to p, but then uses a non-zero extrapolation for lags $|m|>p$.

C. Relationship to Linear Prediction Analysis

If $\hat{x}[n]$ denotes the estimate of the random process $x[n]$, then the forward linear prediction estimate is

$$\hat{x}^f[n] = -\sum_{k=1}^{m} a^f[k]x[n-k] \ , \tag{4.19}$$

where $a^f[k]$ is the forward linear prediction coefficient at time index k. The forward linear prediction error

$$e^f[n] = x[n] - \hat{x}^f[n] \ , \tag{4.20}$$

is just the difference between the forward estimate and the true value of $x[n]$. The variance of the forward error

$$\rho^f = \mathbf{E}\left\{\left| e^f[n]\right|^2\right\} \ , \tag{4.21}$$

represents the prediction error power. Substituting Equation 4.19 and Equation 4.20 into Equation 4.21 will yield the matrix expression

$$\begin{pmatrix} R_{xx}[0] & R_{xx}^*[1] & \cdots R_{xx}^*[m] \\ R_{xx}[1] & R_{xx}[0] & \cdots R_{xx}^*[m-1] \\ \vdots & \vdots & \ddots \ \vdots \\ R_{xx}[m] & R_{xx}[m-1] & \cdots R_{xx}[0] \end{pmatrix} \begin{pmatrix} 1 \\ a^f[1] \\ \vdots \\ a^f[p] \end{pmatrix} = \begin{pmatrix} \rho^f \\ 0 \\ \vdots \\ 0 \end{pmatrix} , \tag{4.22}$$

which is structurally identical to Equation 4.16. So the equations of linear prediction analysis are structurally identical to the Yule-Walker normal equations of an AR process.

D. Algorithms to Compute AR Parameters

The basis for any AR PSD estimation method's high resolution is that there is no truncation of the ACS. Rather, there is an implied extrapolation of the ACS that generates estimated values for the unavailable lags. Since there is no windowing of the ACS, the AR PSD estimators do not possess the high sidelobes that are characteristic of classical PSD estimators.

The Yule-Walker algorithm represents the most direct approach for finding the AR model parameters and white noise variance ρ of Equation 4.15. Biased autocorrelation estimates $\hat{R}_{xx}[m]$ are substituted for the unknown autocorrelation function $R_{xx}[m]$. $\hat{R}_{xx}[m]$ ensures a stable AR filter, since $\hat{R}_{xx}[0] \geq \hat{R}_{xx}[|m|]$. The Levinson algorithm recursively solves matrix Equation 4.16 for $a[1],...,a[p]$ and ρ given $\hat{R}_{xx}[m]$ for $0 \leq m \leq p$. Given a long data record, the Yule-Walker PSD estimate yields reasonable resolution; however, poor-resolution PSD estimates will result for short data records.

Figure 4.1 shows the connector pane, front panel, and block diagram of the Yule-Walker VI created in LabVIEW 2 to estimate AR model parameters for boiler sensor data. Reading from left to right on the block diagram, an extended-precision floating-point array of data is passed from the connector pane (or front panel) into the node

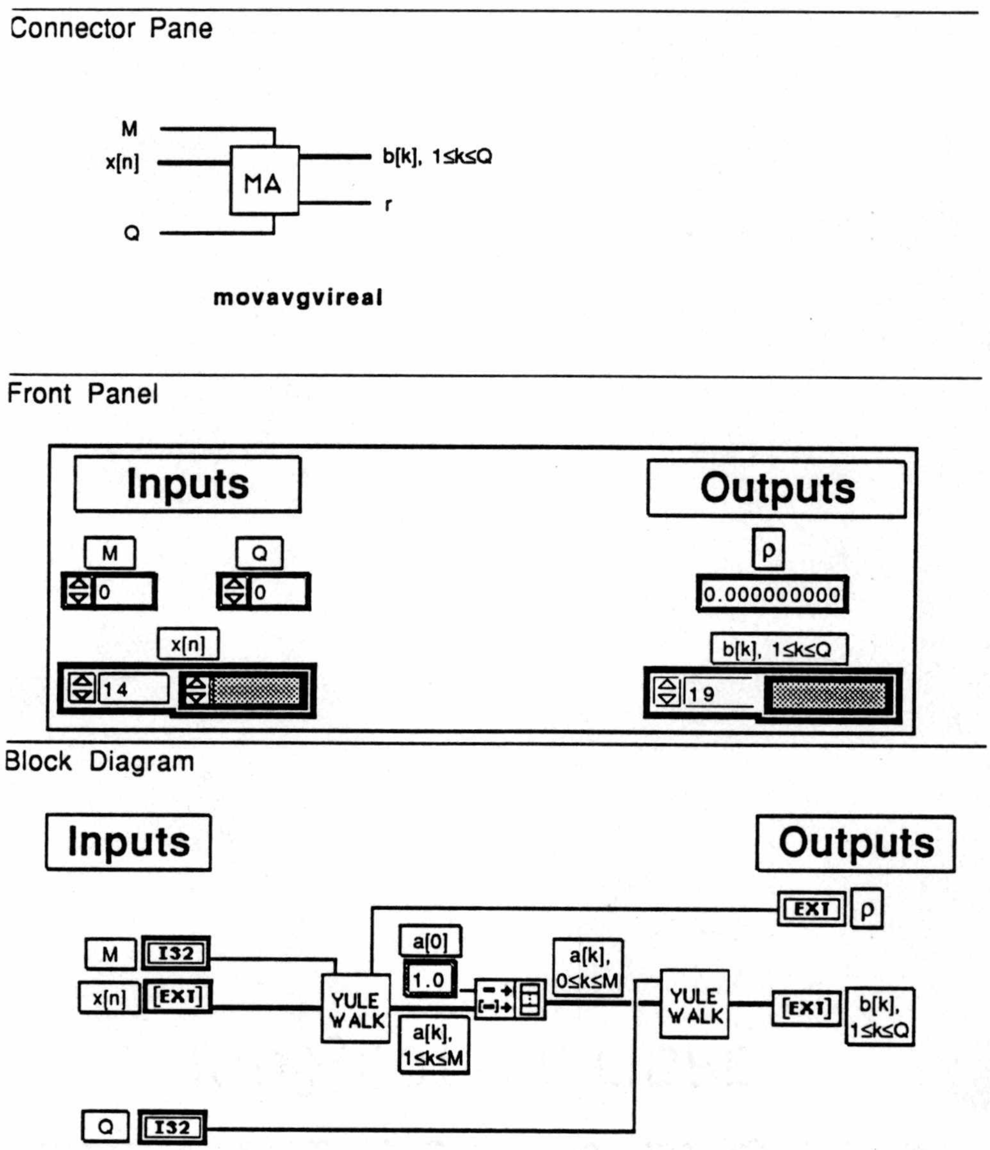


Figure 4.1. Connector Pane, Front Panel, and Block Diagram of Yule-Walker VI.

labeled "$x[n]$". The desired model order is passed in as a 32-bit integer through the node labeled "P". Next, an autocorrelation is performed on the input array $x[n]$ yielding $R_{xx}[m]$, which is then biased by dividing by N (the array size of $x[n]$). Lags m=0 to $m=P$ are obtained after passing the biased autocorrelation array through the icon (or VI) labeled "Subarray". The model order P and the P+1 lags of the biased $R_{xx}[m]$ are then passed to the icon labeled "Lvsn". This VI solves matrix Equation 4.16 by using the Levinson recursive algorithm to estimate the driving noise variance, ρ, and the model parameters $a[k]$ where $1 \leq k \leq P$.

Least squares linear prediction estimation methods improve resolution in the PSD estimate by minimizing the prediction error power ρ^f of Equation 4.21. This ensures more accurate AR coefficient estimates. One algorithm that attempts to do this is the covariance method. This method operates directly on the observed data and performs a least squares minimization with respect to all of the linear prediction coefficients. It uses a separate forward and backward linear prediction algorithm to estimate the AR coefficients.

Figure 4.2 shows the connector pane, front panel, and block diagram of the covariance VI used to estimate AR parameters. This VI is a fast algorithm for the solution of the covariance least squares normal equations. The algorithm computes all lower-order least squares solutions as part of its recursive structure. The case structure in the left part of the diagram takes advantage of this feature. As long as the reset button on the front panel is in the off

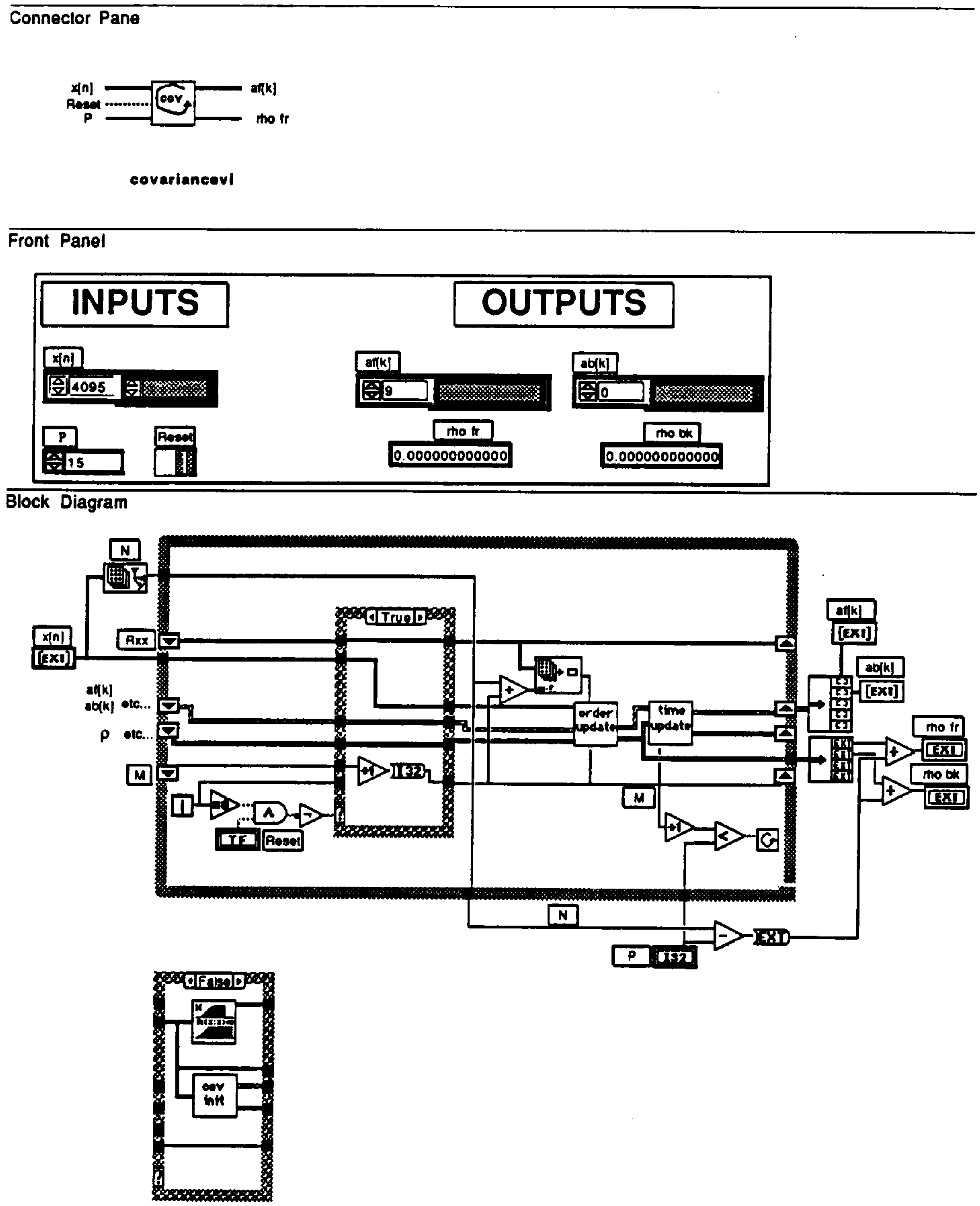


Figure 4.2. Connector Pane, Front Panel, and Block Diagram of Covariance VI.

position, lower-order solutions are saved in shift registers located along the edges of the while loop. This makes it easier to find the best model order number for a given input sequence $x[n]$. Computing a 255th-order solution after finding a 250th-order solution requires only 5 additional recursions, rather than 255 recursions.

Data is passed into the VI through the connector pane (or front panel) as an extended-precision floating-point array into the node labeled "$x[n]$". The model order is passed as a 32-bit integer into the node labeled "P". The reset status is passed into the node labeled "Reset" as a boolean variable. Reading from left to right on the block diagram, the $x[n]$ array is then passed into the "while loop" that executes P times. When reset is "true" on the first loop execution, two basic things occur in the "false" case structure (since the boolean "true" is passed through an inverter): (1), an autocorrelation is performed on $x[n]$ and stored in a shift register for subsequent loop executions; and (2), forward and backward variances and AR model parameters are initialized by the icon labeled "Covar Init". On the first and all subsequent executions, model parameters and white noise variance are recursively updated by order in the icon labeled "order updt", and by time in the box labeled "time updt". After the first execution when reset is "true", all subsequent executions pass data transparently through the case structure labeled "true". The model parameters $a[k]$ (where $1 \le k \le P$), and white noise variance ρ, are then output through their nodes to the front panel for display or the connector pane for use by calling VI's.

E. Order Closing in AR Methods

The improved resolution for finding sinusoids in white noise is a function of the signal-to-noise ratio (SNR). Marple [14] determined that a measure of the sinusoidal resolution of equal-power sinusoids in the case of a known autocorrelation to be roughly

$$F = \frac{1.03}{Tp[\mathrm{SNR}(p+1)]^{0.31}} \,, \tag{4.23}$$

where F is the resolution in Hz, T is the sample interval in seconds, p is the highest lag of the ACS, and SNR is the signal-to-noise ratio of a single sinusoid expressed in linear units.

When the model order p is high relative to N, spurious peaks due to additional zeros appear in the PSD estimate. When $p \ll N$, the resolution suffers, so it may be necessary to experiment with the value of p to find the appropriate resolution/smoothing trade-off for a given process. One guiding factor is the observation of ρ^f; though it always decreases as the model order p increases, there is a point of diminishing returns where its value should not decrease significantly as p increases.

Experimenting with different model orders to determine the best tradeoff between increased resolution and decreased variance for a given set of observed data is termed "order closing". Many criteria have been developed to assist in this process. Akaike [21] suggested two criteria: the final prediction error (FPE) and the

Akaike information criterion (AIC). The FPE for an AR process is defined as

$$FPE[p] = \hat{\rho}_p\left(\frac{N+(p+1)}{N-(p+1)}\right) , \tag{4.24}$$

where N is the number of data samples, p is the order, and $\hat{\rho}_p$ is the estimated white noise variance or linear prediction error variance. Note that the term in parentheses increases as p increases to account for the uncertainty of the estimate $\hat{\rho}_p$. The particular order p is selected for which the FPE is minimum. Assuming the process has Gaussian statistics, the AIC for an AR process is defined as

$$AIC[p] = N \ln(\hat{\rho}_p) + 2p \ . \tag{4.25}$$

The term $2p$ penalizes the function when the increasing of p no longer significantly decreases the prediction error variance.

Figure 4.3 shows the front panel of "ARtuner". This VI is specifically designed for order closing, taking advantage of the lower-order solutions stored in shift registers in the covariance VI it calls. Determining the best model order is greatly aided by observing the front panel graph labeled "Tuner". This graph plots both the FPE and AIC as functions of the model order, and stores each execution's result in the appropriately labeled array next to the graph. The highest model order with a corresponding minimum FPE or AIC value would be an appropriate model order choice.

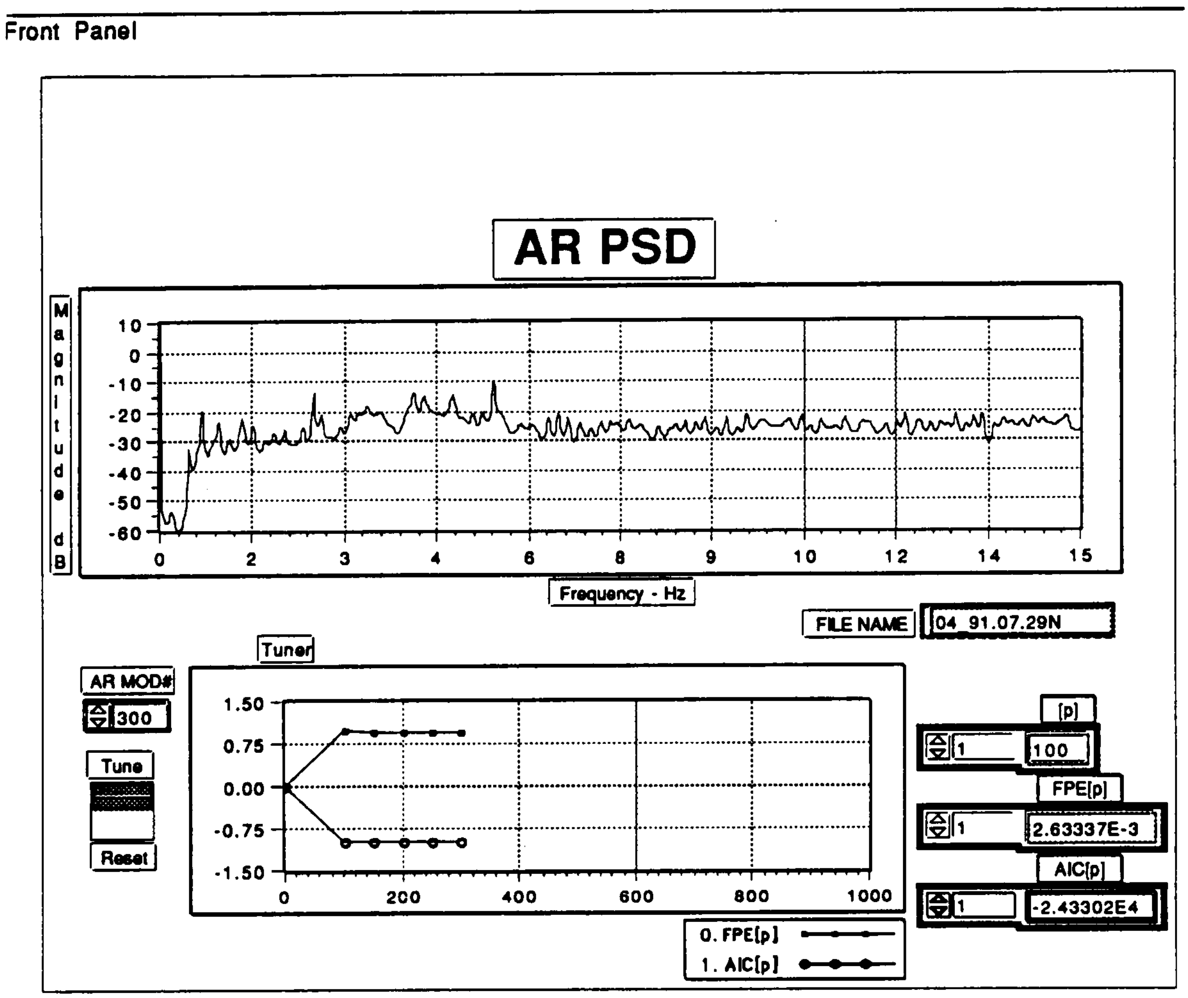


Figure 4.3. Front Panel of Order Closing VI Called "ARtuner".

F. Algorithm to Compute MA Parameters

At first glance, it may be tempting to solve the nonlinear equation produced by Equation 4.15 when $x[n]$ is to be represented by a MA process (q>0). The solutions, however, often involve difficult spectral factorization techniques [16]. As discussed previously, any parametric model type can be estimated by another model type, in this case:

$$B[z] = \frac{1}{A_{\infty}[z]} \ . \tag{4.26}$$

An infinite-order AR process is impractical, but an AR model order much greater than the MA model order desired is usually sufficient. Since

$$b[k] = Z^{-1}\left(\frac{1}{A_{\infty}[z]}\right) , \tag{4.27}$$

it can be shown [14] that the MA parameters $b[k]$ can then be estimated from the AR parameters by substituting the AR parameters into the autocorrelation matrix of Equation 4.16

$$\begin{pmatrix} a[0] & a^*[1] & \cdots a^*[m] \\ a[1] & a[0] & \cdots a^*[m-1] \\ \vdots & \vdots & \ddots \vdots \\ a[m] & a[m-1] & \cdots a[0] \end{pmatrix} \begin{pmatrix} 1 \\ b^f[1] \\ \vdots \\ b^f[p] \end{pmatrix} = \begin{pmatrix} \rho^f \\ 0 \\ \vdots \\ 0 \end{pmatrix} . \tag{4.28}$$

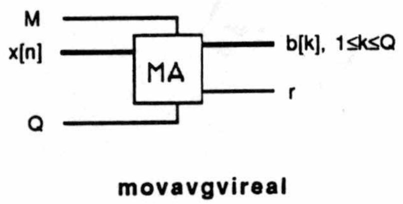
Figure 4.4 shows the connector pane, front panel, and block diagram of the moving average VI called "movingvireal". This VI employs the Yule-Walker AR algorithm to find the MA model parameters. If the desired MA model order passed from the connector pane (or front panel) is Q, then the long AR model used to estimate the MA parameters should have a model order $M \geq 2*Q$. Reading from left to right on the block diagram, the data array $x[n]$, and the AR model order M, are passed into the Yule Walker VI. This VI outputs the driving white noise variance ρ, and the AR model parameters $a[k]$ where $1 \leq k \leq M$. The first parameter, $a[0]=1$, is then appended to the front of this array; it is then passed back into the Yule Walker VI along with the desired MA model order Q. The Yule Walker VI solves matrix Equation 4.28 to produce the MA model parameters $b[k]$.

Processes that have broad peaks or sharp nulls are best estimated by MA processes. MA PSD estimates can display deep nulls in spectra, yet they lack the sharpness of resolution of the AR PSD estimates. The *AIC* criterion can be used for order closing in MA models as well, if the MA model order q is substituted for the AR model order p in Equation 4.25.

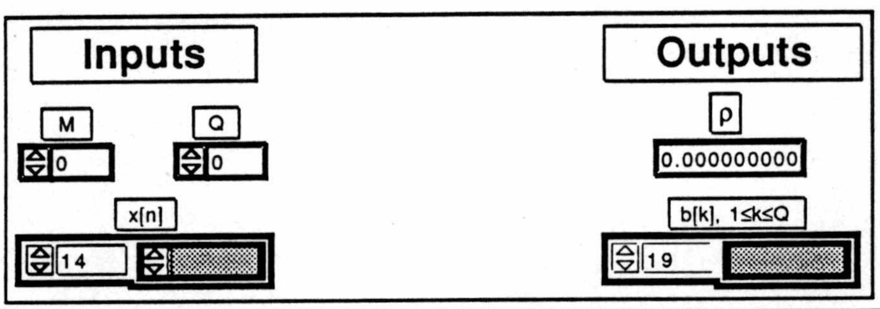
G. <u>Algorithm to Compute ARMA Parameters</u>

It is, as discussed previously, difficult to solve the nonlinear equations required to simultaneously estimate the parameters of an ARMA model. Practical techniques, however, have been developed

Connector Pane



Front Panel



Block Diagram

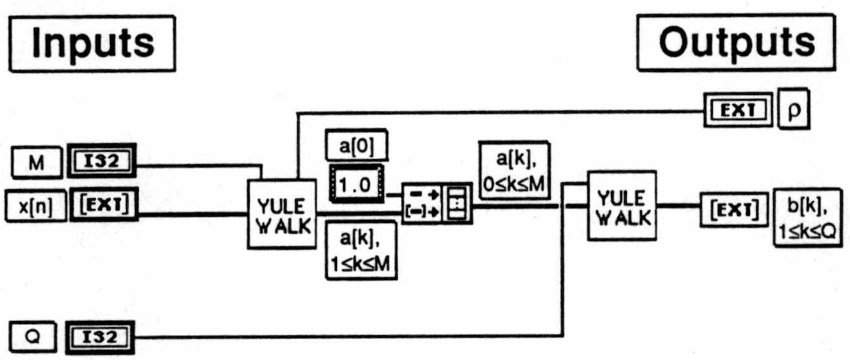


Figure 4.4. Connector Pane, Front Panel, and Block Diagram of Moving Average VI.

to significantly reduce this computational complexity. These techniques, though suboptimum, use linear equations to estimate the MA and AR parameters separately. Usually these techniques estimate the AR parameters first, then use an inverse filter derived from these parameters to apply to the original data [14]. The resulting filtered data is then input to a MA parameter estimator.

Figure 4.5 shows the connector pane, front panel, and block diagram of the ARMA VI used to estimate the ARMA model parameters. The original data sequence $x[n]$, the maximum lag to use for autocorrelation estimates M, the AR model order P, and the MA model order Q are passed into the block diagram from the connector pane (or front panel). Reading from left to right on the block diagram, $x[n]$ is passed into the VI "Auto Sub" which calculates an unbiased autocorrelation and returns $\hat{R}_{xx}[k]$, where $(Q-P+1) \leq k \leq M$. $\hat{R}_{xx}[k]$ is then input to the covariance VI where the AR parameters $a[k]$ are estimated. Next, the original sequence is filtered by convolving it with $a[k]$. Finally, the MA parameters and the driving noise variance ρ are estimated from the filtered sequence in the MA VI.

ARMA PSD estimates generally have model orders p and q much smaller than AR PSD estimates. The PSD spectrum usually displays the deep nulls associated with the MA PSD spectrum and the sharp peaks associated with the AR PSD spectrum. Order closing can then be estimated by modifying Equation 4.25 as follows:

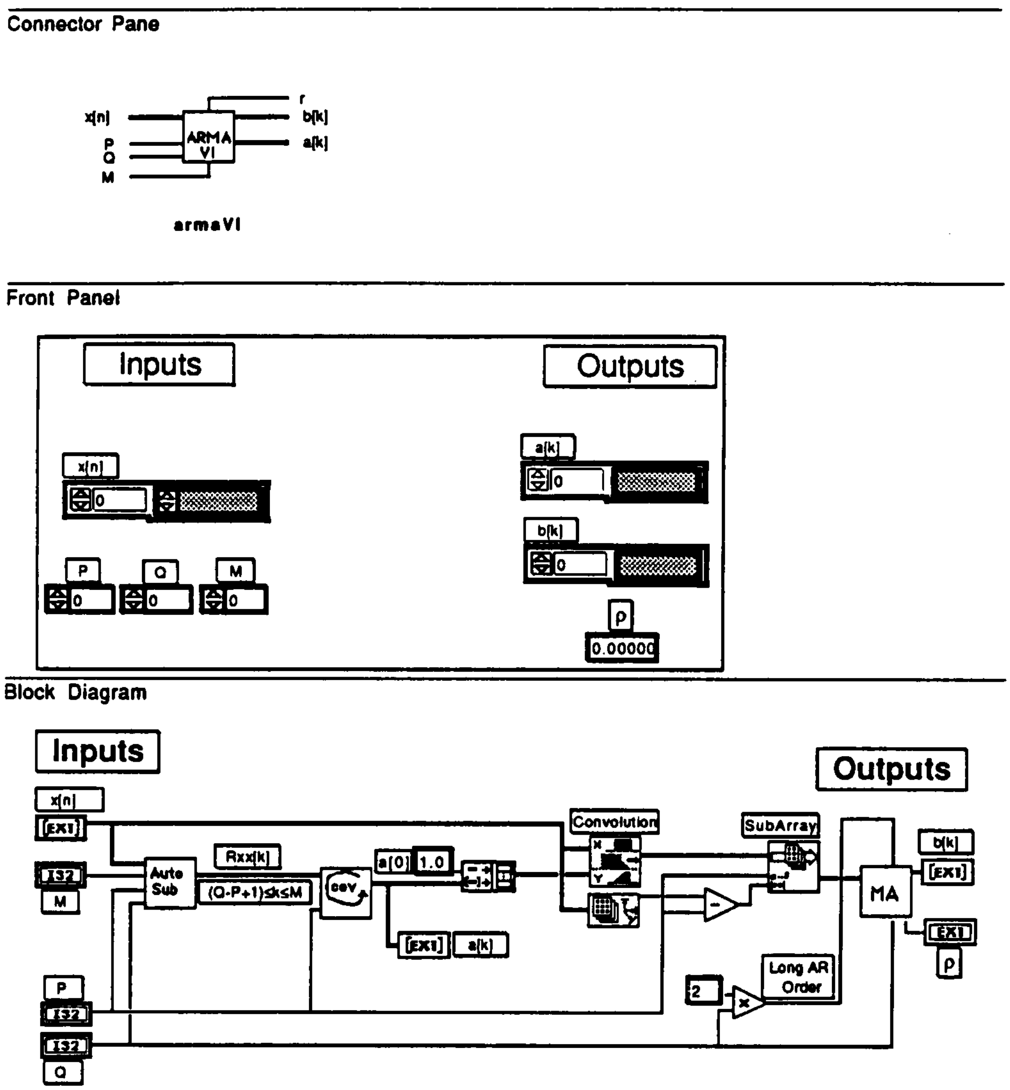


Figure 4.5. Connector Pane, Front Panel, and Block Diagram of ARMA VI.

$$AIC[p,q] = N \ln\left(\hat{\rho}_{pq}\right) + 2(p+q) \; . \tag{4.29}$$

This is not a simple procedure since orders p and q must be adjusted to find the minimum of $AIC[p,q]$. The optimum order for an unknown process may be different than the computed order using an order closing formula; therefore, Marple suggests that subjective judgement is still required to determine the appropriate model order from actual signals of an unknown process [14].

In the next section, the parametric methods discussed in this section and the classical methods of the previous section will be used to calculate and graph PSD estimates of boiler sensor data. The resulting PSD estimates will be analyzed to determine which method might be best in detecting tube leaks.

V. ANALYSIS OF BOILER ACOUSTIC DATA

The primary goal of this project was to examine boiler acoustic data using both classical and parametric methods of PSD estimation and determine which method or combination of methods produced the most reliable leak detection system. The two classical PSD methods employed were the correlogram and periodogram methods. The three parametric PSD methods used were the MA, AR, and ARMA methods. Acoustic data for all 24 sensors of Unit 5 of the TVA Kingston Fossil Fuel Power Plant were recorded and observed during the course of a year. This provided a sea of information for processing and analysis. It thus became clear that concentrating on one or two sensors located near leak-prone areas of the furnace might be the most reasonable approach during early development. Sensor 4 of the superheat furnace was chosen as a case study for this section because multiple leaks were discovered near it during the most intensive data collection stage of the project.

A. Preliminaries

Acoustic data used in this analysis was recorded from the Fall of 1990 through the Fall of 1991. The earliest recordings were taken approximately once or twice a month primarily in response to leak occurrences. Data was recorded from each sensor channel during a known leak occurrence before the operator shut the unit down for repairs. Then each channel was recorded once the unit was repaired and placed back on-line.

These early recordings were instructive. Each sensor's PSD was examined for the presence of the relatively high frequency tones that Stulen [5] suggests are a leak's spectral signature. Then each sensor's leak PSD was compared against its corresponding non-leak PSD. The non-leak data obtained immediately after repair was initially assumed to be the best benchmark for comparison, since it represented data that was least likely to have an undetected leak present.

Figure 5.1 shows a leak PSD and non-leak PSD of sensor channel 17 in the reheat furnace of Unit 5. The legend at the bottom of the graph lists the data file name and its corresponding type of line-plot. For example, the first file is represented by a dashed line and in this particular case has a name of "17_90.11.15N". The first two characters, "17", of the file name represent the sensor number from which the data in the file was recorded. After the underscore, "90.11.15" indicates that it was recorded on November 15, 1990. The last character, "N", indicates that it was recorded when no official leak condition had been declared. If this character is instead an "L", the file contains the raw acoustic data recorded during a known leak condition. In other words, the furnace operator had officially acknowledged that a leak was present somewhere in the furnace.

According to the legend, the graph of Figure 5.1 is the overlay of two PSD estimate plots of sensor 17. The solid line plot represents

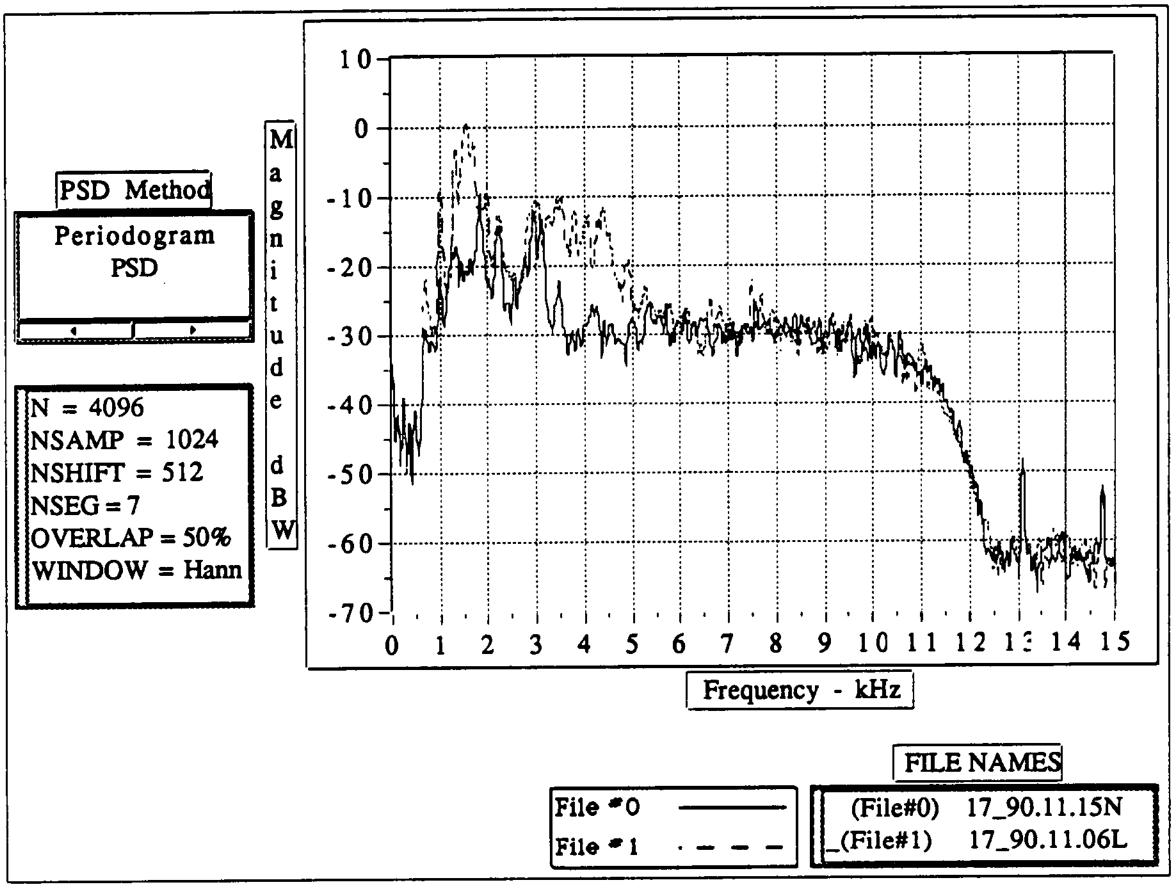


Figure 5.1. Overlay of Leak and Non-Leak PSD Plots of Sensor 17.

non-leak data obtained on November 15, 1990. The dashed line plot represents leak data obtained on November 6, 1990. The periodogram PSD estimation method was used in this case and is shown in the text box to the left of the graph.

First impressions of the two plots indicate that a leak's spectral signature on sensor 17 is generally elevated 15 to 20 decibel-Watts (dBW) between 1000 and 2000 Hz and between 3000 and 5000 Hz. This conclusion, however, does not take into account the variation that might normally occur in the PSD estimate from one day to the next. In other words, it may be that this channel normally undergoes 15 to 20 dBW swings in the frequency bands between 1000 and 5000 Hz, regardless of the leak condition.

Any meaningful analysis therefore, must account for normal fluctuations in the PSD estimate. Since each sensor channel's PSD has its own characteristic fluctuations, developing an average PSD estimate for a particular sensor would serve as a baseline PSD plot to overlay on subsequent PSD plots of sensor data from that channel. This baseline plot would indicate with a higher degree of confidence whether suspected leak PSD estimates actually deviated from the norm in a frequency band. In addition, it would aid in the analysis to have upper and lower threshold PSD plots of a particular sensor to overlay on subsequent PSD plots. This would give perspective to occasional fluctuations that may be averaged out in developing the baseline PSD.

Developing standard ways to view and present PSD analysis became increasingly necessary as research and study of the data progressed. Optimum parameters for each PSD method were eventually settled upon and are noted where applicable in the following sections. The standard VI panels and graphs that follow in this section evolved as the seemingly innumerable ways to view the data were investigated. The key to navigating the maze was the flexibility afforded by the LabVIEW software. Changes in graphical PSD presentation, both programmatic and visual, were relatively easy to achieve in LabVIEW.

B. Baseline and Threshold PSD Development

The baseline average, upper threshold, and lower threshold PSD estimate for each method and each sensor became the standard ingredient for individual file comparison and analysis for this project. Figure 5.2 shows the front panel and Figure 5.3 shows the block diagram of the "BaseThresh" VI created to compute upper and lower thresholds and average baselines. Each of the three arrays computed can be saved to its own file for use in subsequent PSD plot comparisons. For each sensor being analyzed, thresholds and baselines should be computed for each PSD method, since every PSD method will have its own spectral peculiarities given the same acoustic data. The bottom left corner of Figure 5.2 shows the front panel control for choosing the desired PSD method. The graph "PSD Average" overlays the plot of the baseline (or average) with the plot

Front Panel

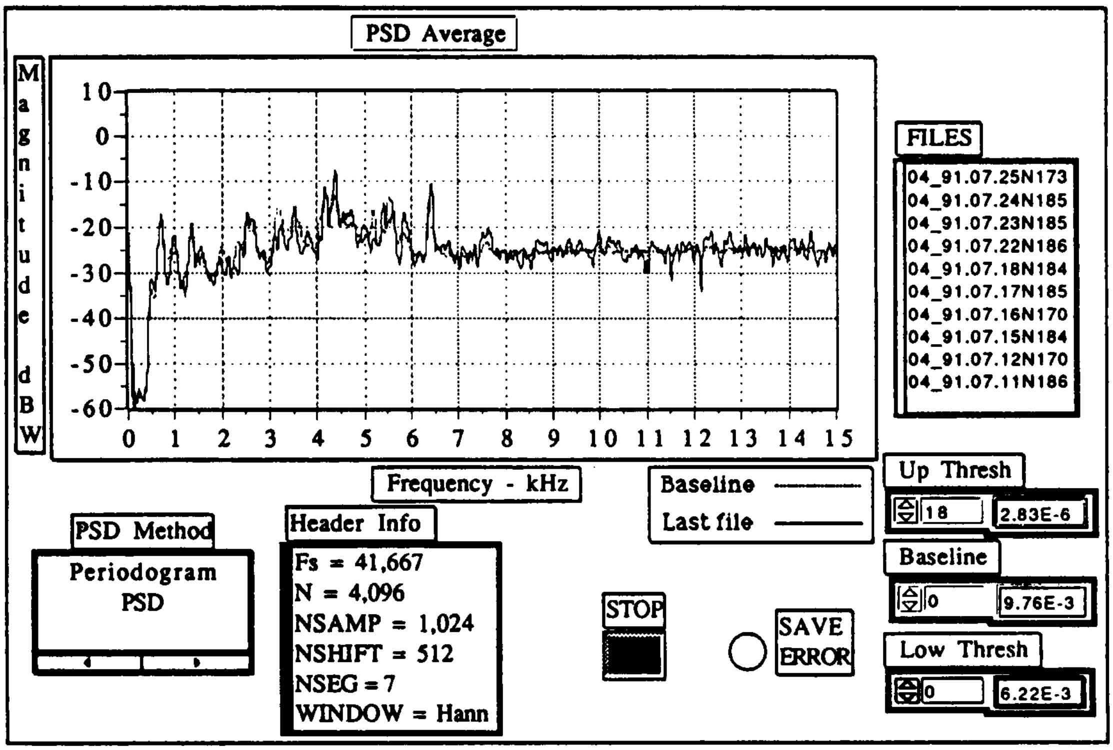


Figure 5.2. Front Panel of "BaseThresh" VI.

Block Diagram

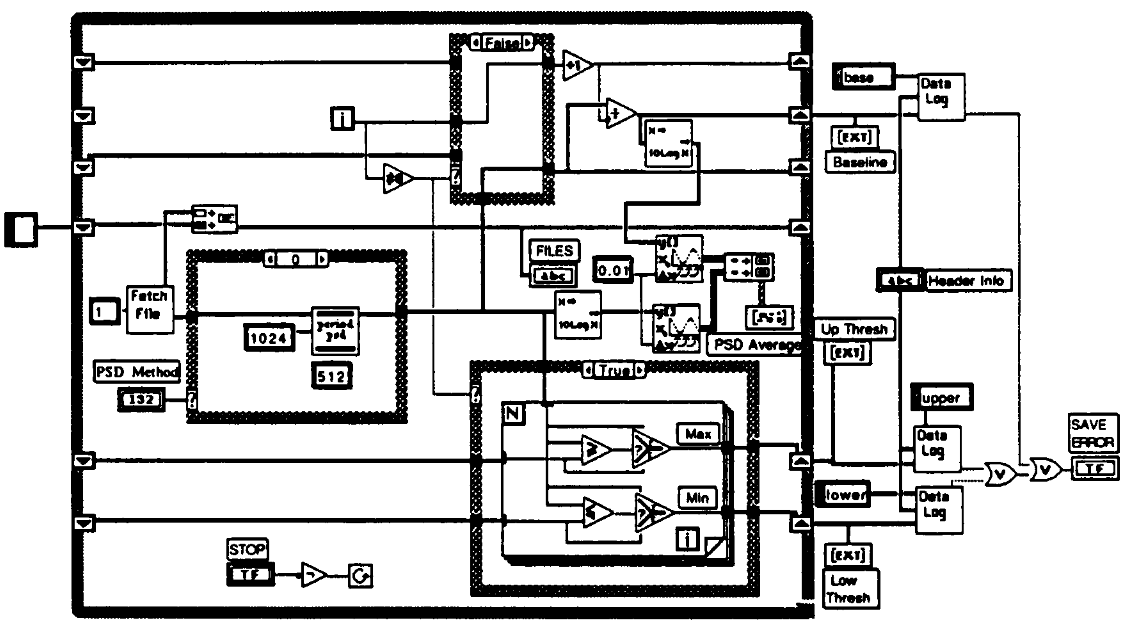


Figure 5.3. Block Diagram of "BaseThresh" VI.

of the last file used in computing the average. The text box to the right of the graph lists the raw acoustic data files used in computing the baseline.

The main structure in the block diagram of Figure 5.3 is the LabVIEW while loop. Shift registers along the sides of the while loop store the baseline and threshold arrays and the list of file names used. The VI labeled "Fetch File" in the left part of the loop allows the user to interactively retrieve acoustic data files and pass them into the program. This VI returns the file name and the data. The file name is appended to a string array and stored in the shift register immediately above the icon. The data is passed to the right into the case structure that contains the selected PSD method. The computed PSD estimate is then passed out the right of the PSD case structure and into two other case structures. The aforementioned case structure contains five different cases, one for each PSD method; whereas, the other two case structures have only a logical "true" and "false" case. If the loop execution count is zero (indicating the first loop execution), the "false" case is selected and the PSD data is simply loaded into the shift registers. On every subsequent loop execution, the "true" case is selected and the PSD is averaged in the top case structure and thresholds are determined in the lower case structure. Once all of the desired files have been processed, the while loop is terminated and the baseline, upper threshold, and lower threshold arrays are saved to disk in separate files by the VI "Data Log" on the right part of the figure.

Characterizing a sensor channel with thresholds and a baseline only has worth if the files used in computing them are carefully chosen. Using data files recorded on days immediately preceding a leak is dangerous. The upper threshold computed might represent the beginnings of a leak occurrence rather than normal fluctuations of the characteristic PSD. In fact, using data files recorded when the furnace is experiencing any brief, abnormal conditions will introduce unwanted bias in the baseline and either the upper or lower threshold. If a file produced a large change in one of the thresholds, files recorded on subsequent days of uninterrupted furnace operation were examined closely to see if they adjusted back toward the baseline. If they did, it was assumed that the PSD magnitude swing was a normal fluctuation. Thus the file in question should be listed in the calculation of the baseline and thresholds.

In order to compute the thresholds for the acoustic data it is necessary to use files recorded on subsequent days of uninterrupted furnace activity. Since during the Fall of 1990 and the Spring of 1991 acoustic data was recorded only sporadically, these recordings are used only for reference. During the Summer of 1991 though, data was recorded virtually every day on every sensor. The furnace was brought down four times for repair, but the month of July presented the most consecutive, uninterrupted days of furnace operation. Files recorded in this month hold the most promise for providing adequate baselines and thresholds for this analysis.

Figure 5.4 displays the baseline PSD of Sensor 4 calculated using the periodogram method. The text box in the bottom left of the figure contains the header information stored when this file was saved to disk. This stamps the file with the values of pertinent parameters used in the periodogram method. After much analysis during the project, it was determined that the optimum parameters for the periodogram method for the sensor data are as listed in the text box. Since the data was recorded in blocks of 4096 points, segment lengths of 1024 points with a shift of 512 points between segments (50 percent overlap) resulted in seven segments. Seven segments employing the Hann data-window seemed to give the best tradeoff between resolution and variance.

The baseline PSD is an average of ten different acoustic data files, so the spectrum will appear smoother than a single data file's PSD. The several peaks in Figure 5.4 between 750 and 6,000 Hz range from 5 to 15 dBW in magnitude and can be considered a normal characteristic of Sensor 4. The peak centered at 6,400 Hz is approximately 15 dBW in magnitude and sporadically appears in the spectrum of this and other sensor channels. The baseline plot is relatively stable from about 6,500 to 15,000 Hz.

Figure 5.5 shows the lower threshold PSD of Sensor 4 calculated using the periodogram method. The 15 dBW peak at 6,400 Hz is absent illustrating its sporadic nature. Virtually every peak between 750 and 6,000 Hz in the baseline spectrum is preserved in the lower threshold. The fluctuations in magnitude do not distort the plot

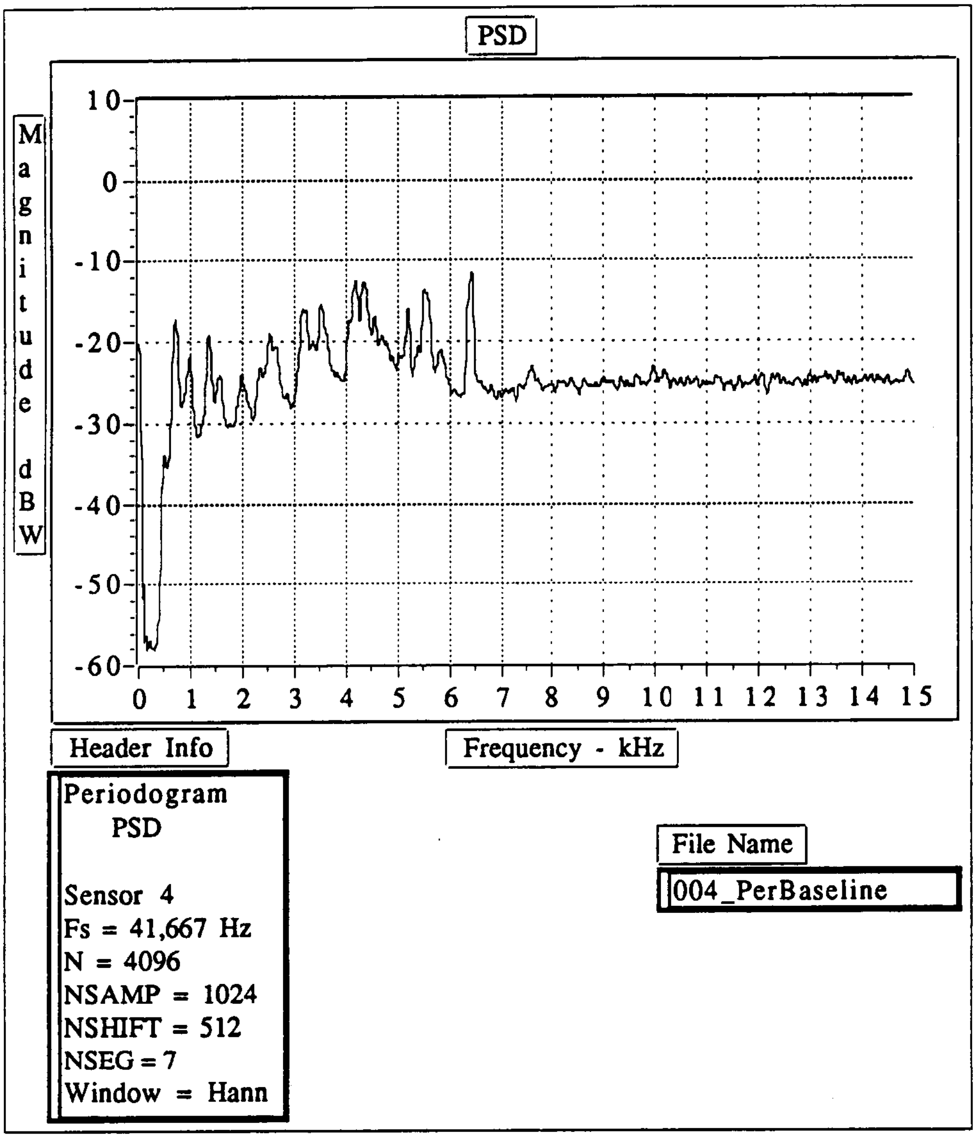


Figure 5.4. Baseline PSD of Sensor 4 Produced by Periodogram Method.

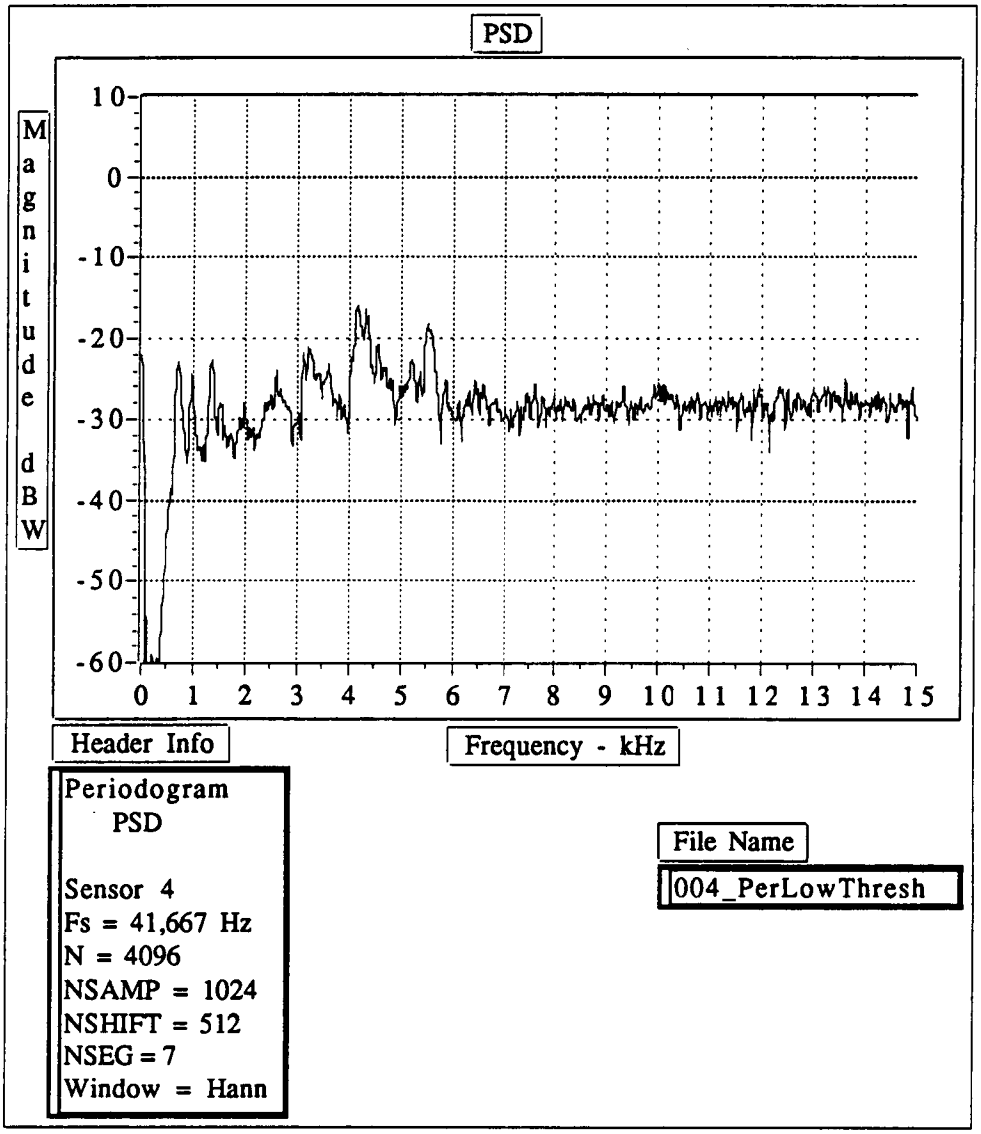


Figure 5.5. Lower Threshold PSD of Sensor 4 Produced by Periodogram Method.

significantly. This means that its general shape is relatively stable.

Figure 5.6 shows the upper threshold PSD of Sensor 4 calculated using the periodogram method. The general shape of the spectrum is comparable to the average baseline PSD. No odd shape appears in the plot, but the peaks from 750 to 6,500 Hz are accentuated. The 5 dBW peak that appears at 7,600 Hz appears as a 3 dBW peak in the baseline spectrum and is almost indistinguishable in the lower threshold PSD. The upper threshold plot does not fluctuate from the baseline plot from 7,000 to 15,000 Hz as much as the lower threshold does. Peaks more than a couple of dBW are rare in this higher frequency range; however, 5 to 10 dBW fluctuations from the average are not uncommon below 7,000 Hz.

Figure 5.7 overlays the periodogram PSD plots of the baseline, upper threshold, and lower threshold of Sensor 4. The legend at the bottom of the graph indicates that the plot of the baseline is a dashed-line; however, it may be difficult to distinguish from the threshold plots. This graph, though, illustrates the range within which a PSD plot of Sensor 4 acoustic data would be expected to fall. Fluctuations that exceed these thresholds would not be considered normal and might indicate the presence of a leak.

Figure 5.8 overlays the correlogram PSD plots of the baseline, upper threshold, and lower threshold of Sensor 4. The Hann data-window and 10% lag index were eventually regarded as optimum for the acoustic data. The shape of each plot generally agrees with its periodogram counterpart. The correlogram method however,

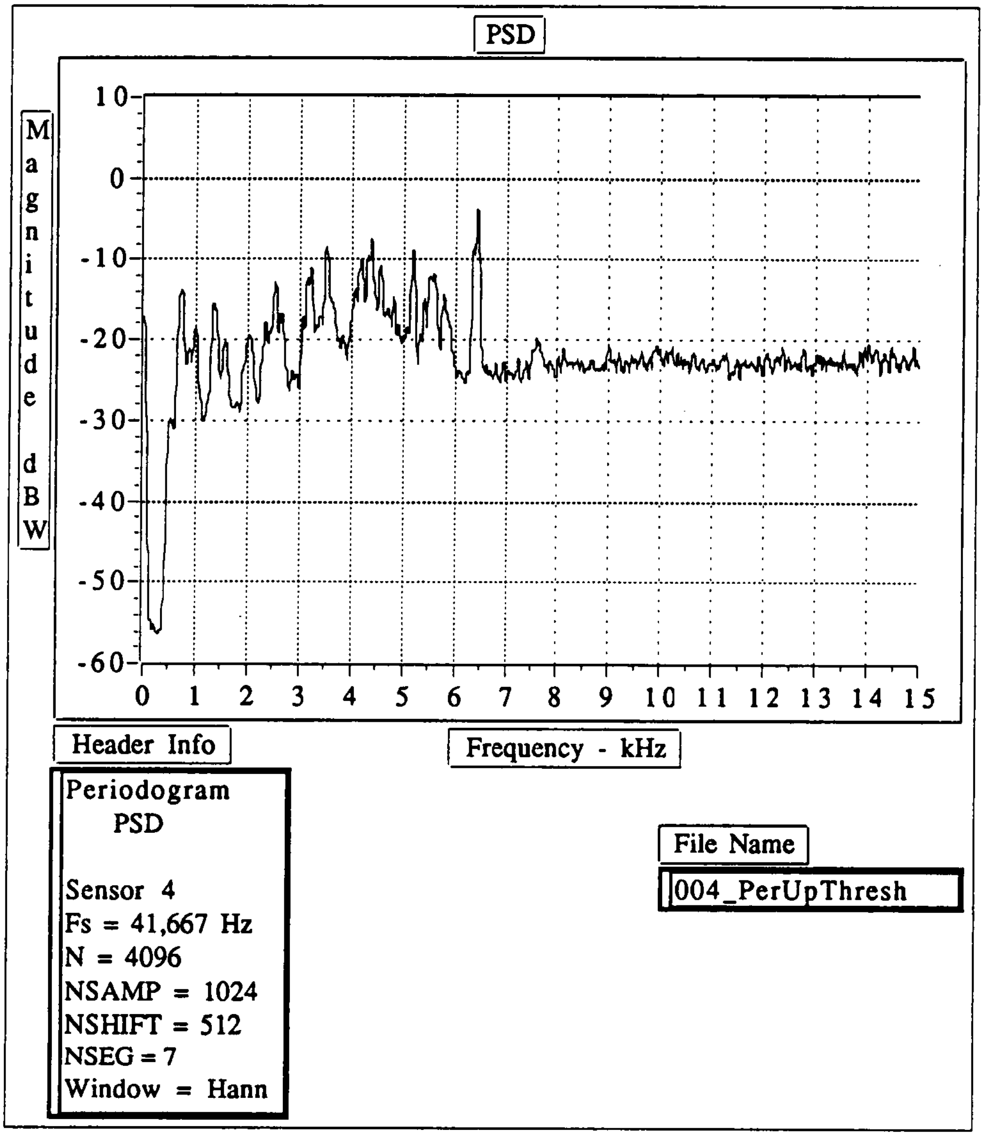


Figure 5.6. Upper Threshold PSD of Sensor 4 Produced by Periodogram Method.

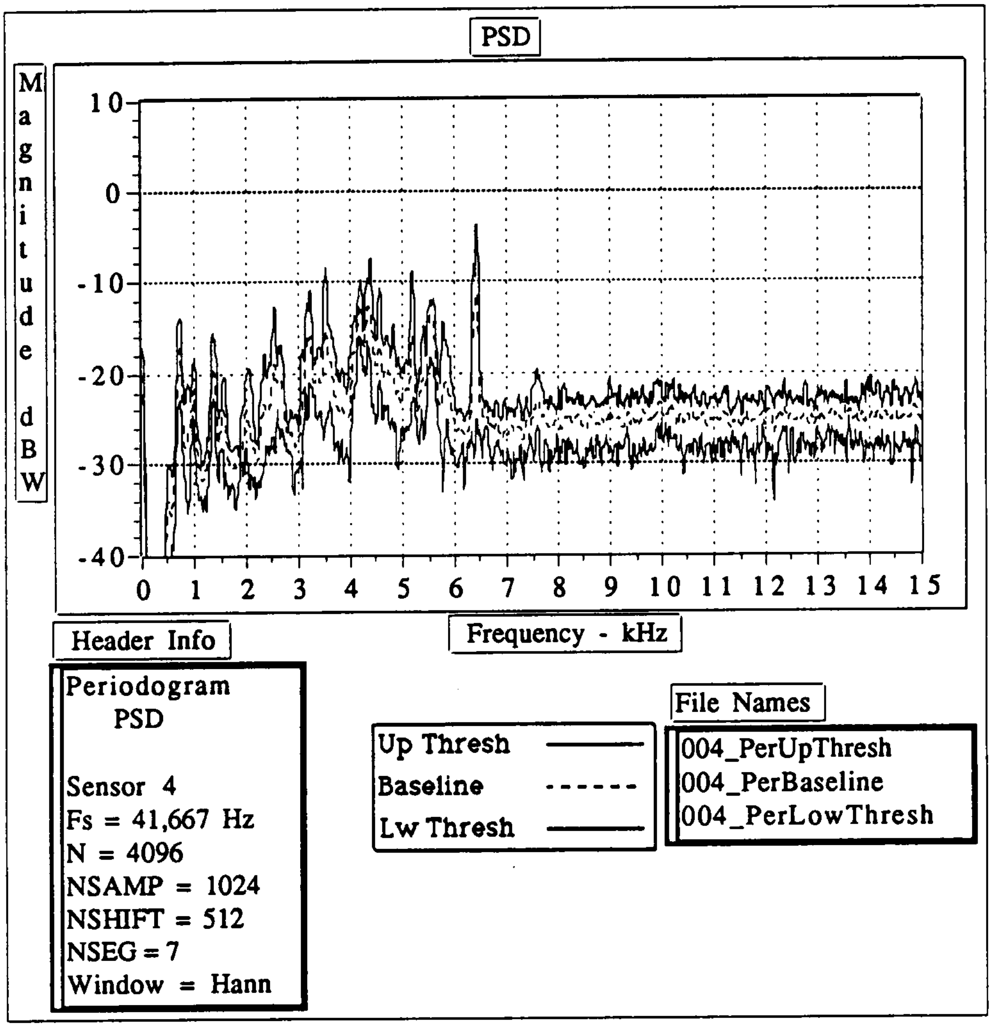


Figure 5.7. Overlay of Periodogram PSD Plots of Baseline, Lower Threshold, and Upper Threshold of Sensor 4.

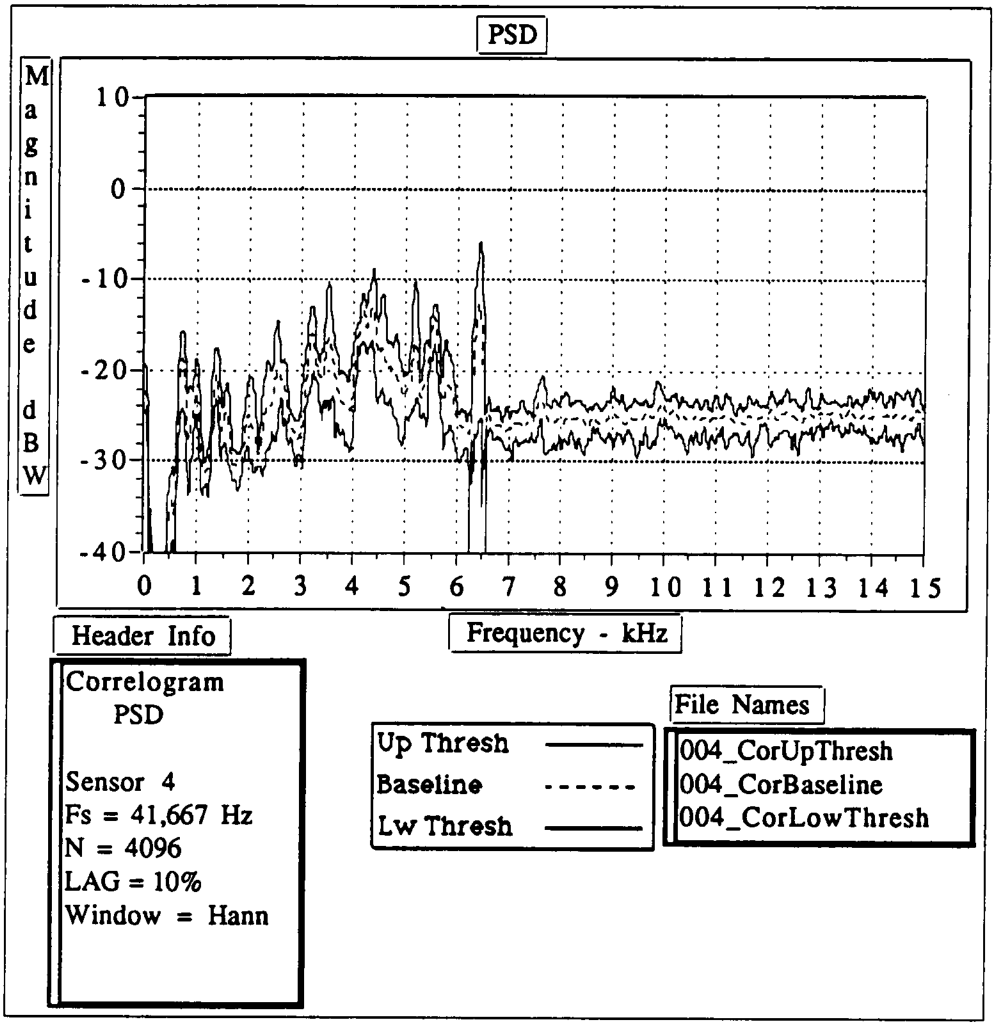


Figure 5.8. Overlay of Correlogram PSD Plots of Baseline, Lower Threshold, and Upper Threshold of Sensor 4.

displays deep nulls in the PSD spectrum that the periodogram method omits. This characteristic is best illustrated by the lower threshold plot at 6,400 Hz. The deep nulls reveal the presence of what may be the sporadic peak that occurs at this frequency. Peaks, however, are not quite as sharp with this method and tend to smear together.

Figure 5.9 overlays the parametric MA PSD plots of the baseline, upper threshold, and lower threshold of Sensor 4. The tuning techniques described in the previous section to find the appropriate MA model order were determined to be too low for the acoustic data. A model order of 300 eventually became standard after many trials. The shape of each plot is similar to the correlogram plots. Since the MA PSD is a moving-average, the relative smoothness and peak attenuation is not surprising. The MA PSD is especially suited for detecting overall level shifts of the spectra, and tends to be better at displaying nulls around peaks than the AR PSD method.

Figure 5.10 overlays the parametric AR PSD plots of the baseline, upper threshold, and lower threshold of Sensor 4. Tuning techniques suggested that a model order of 450 would yield the highest resolution without causing too much variance. The AR PSD is especially suited for resolving adjacent tones in the spectra, and tends to accentuate peaks. For instance, two peaks are seen in the upper threshold at 6,400 Hz with this method; whereas, both of the classical PSD methods and the parametric MA PSD method show this

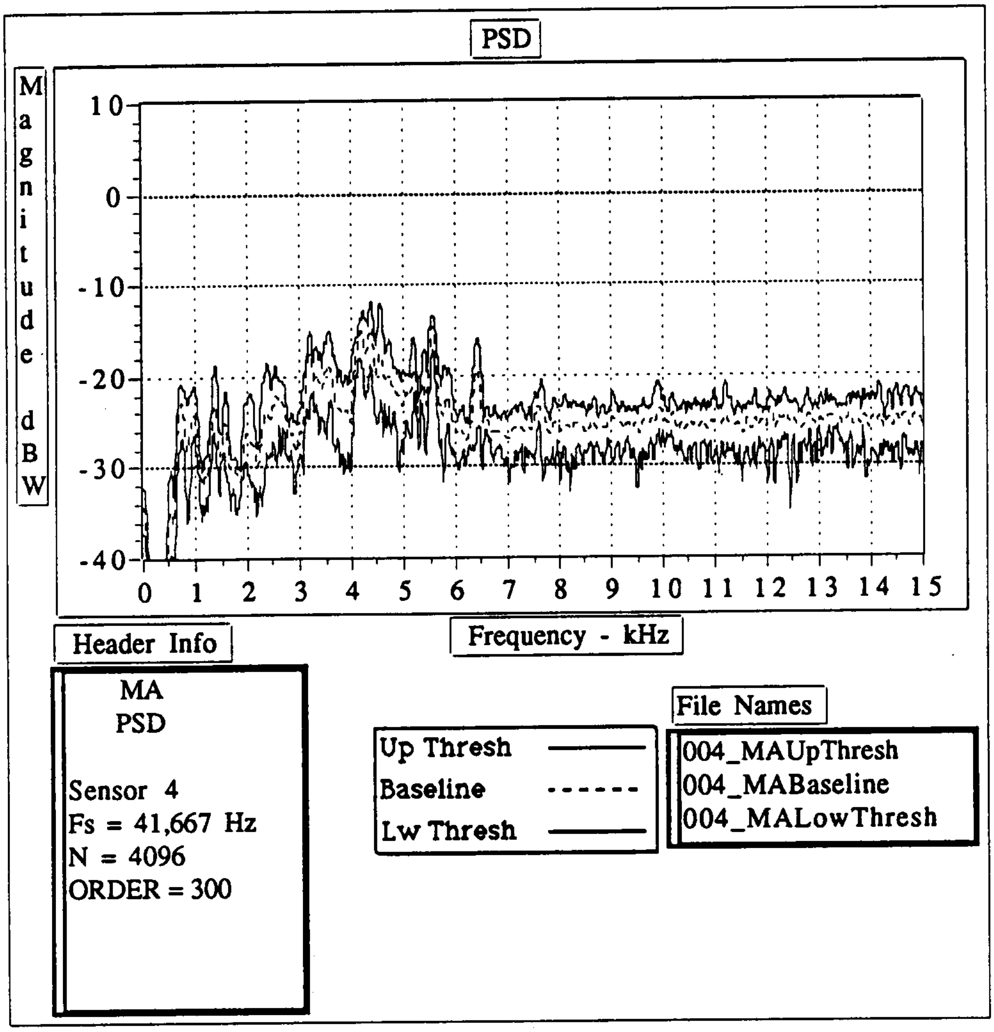


Figure 5.9. Overlay of Parametric MA PSD Plots of Baseline, Lower Threshold, and Upper Threshold of Sensor 4.

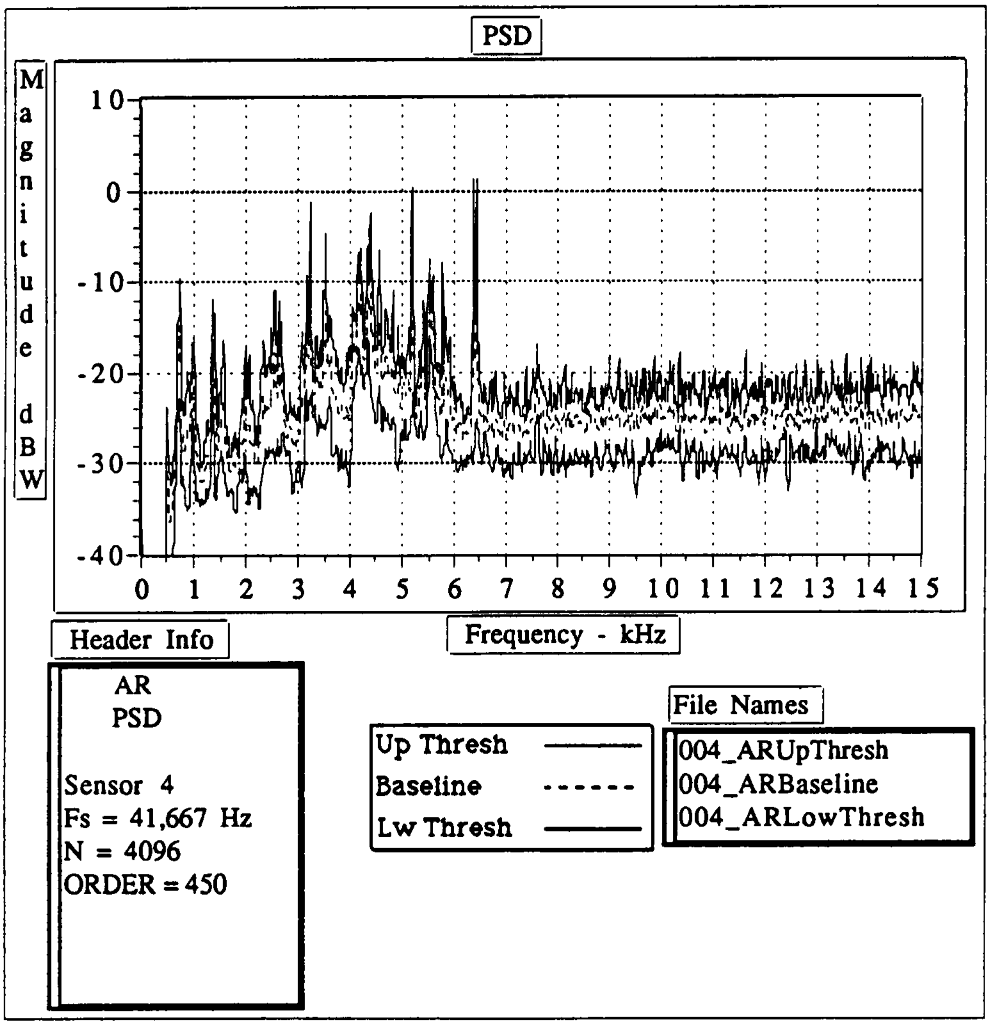


Figure 5.10. Overlay of Parametric AR PSD Plots of Baseline, Lower Threshold, and Upper Threshold of Sensor 4.

as one tone. Of course, since the upper threshold is a conglomerate of values derived from 10 different files, it may be that this represents a slight shift in tone location from one file to the next. The extra performance of resolution and peak amplification come at a cost of increased variance. This is especially seen in the range from 7,000 to 15,000 Hz.

Figure 5.11 shows the baseline ARMA PSD plot of Sensor 4. Trial and error produced an optimum combination of an MA model order of 50, and an AR model order of 300. The ARMA PSD combines the sharp resolution of the AR method with the smoother, deep-nulled MA method. This increased resolution is due to the absence of windows that cause smearing in the classical methods. The baseline plot generally agrees with the classical PSD plots, but each peak is sharper and each null is deeper. Yet the ARMA PSD plot is remarkably stable throughout the spectrum.

Figure 5.12 overlays the parametric ARMA PSD plots of the baseline, upper threshold, and lower threshold of Sensor 4. There is more fluctuation in magnitude from 750 to 6,000 Hz than in the classical methods; however, about the same fluctuation occurs from 7,000 to 15,000 Hz as in the periodogram method. Peaks in the upper threshold in this range are more sharply defined but still vary no more than 5 dBW from the baseline.

C. <u>File Comparison</u>

Each method's baseline and threshold plots have been

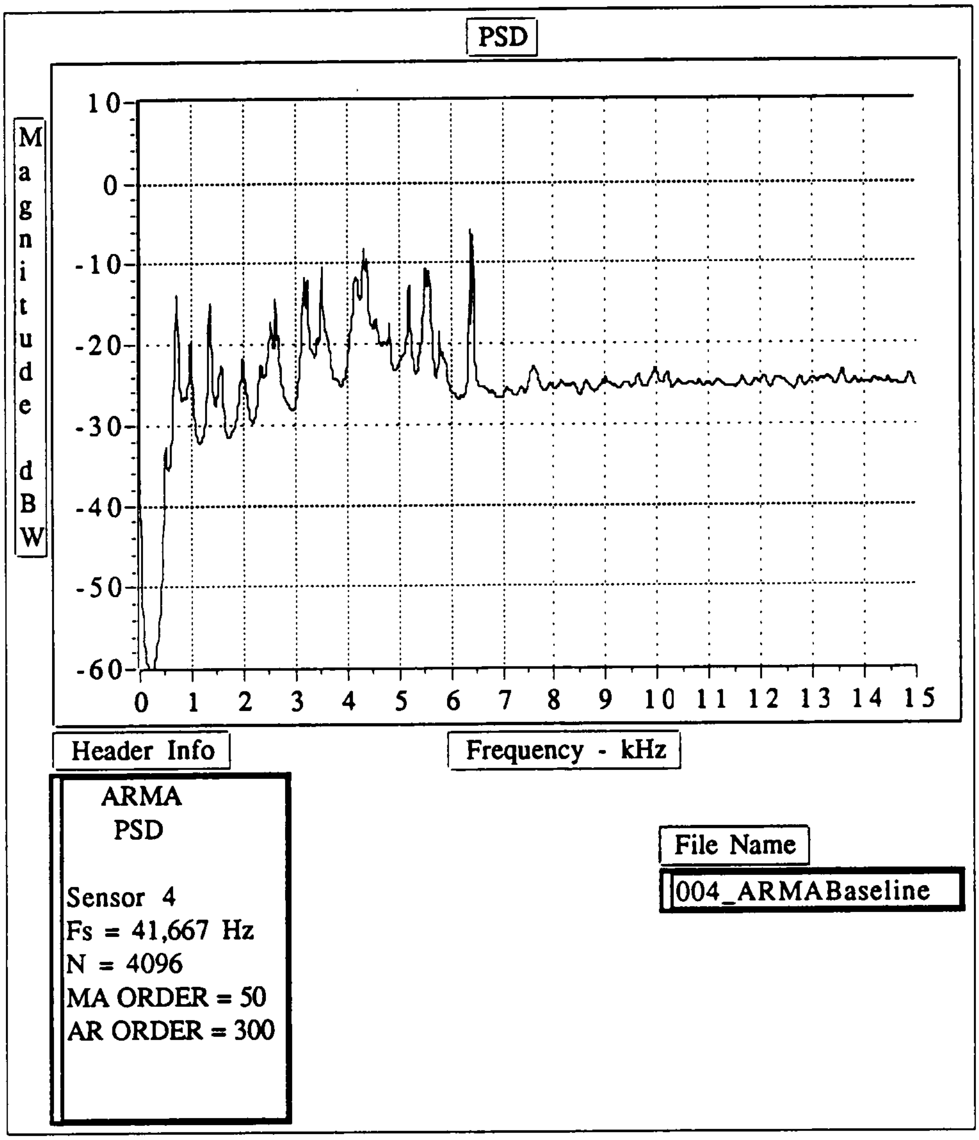


Figure 5.11. Baseline PSD of Sensor 4 Produced by Parametric ARMA PSD Method.

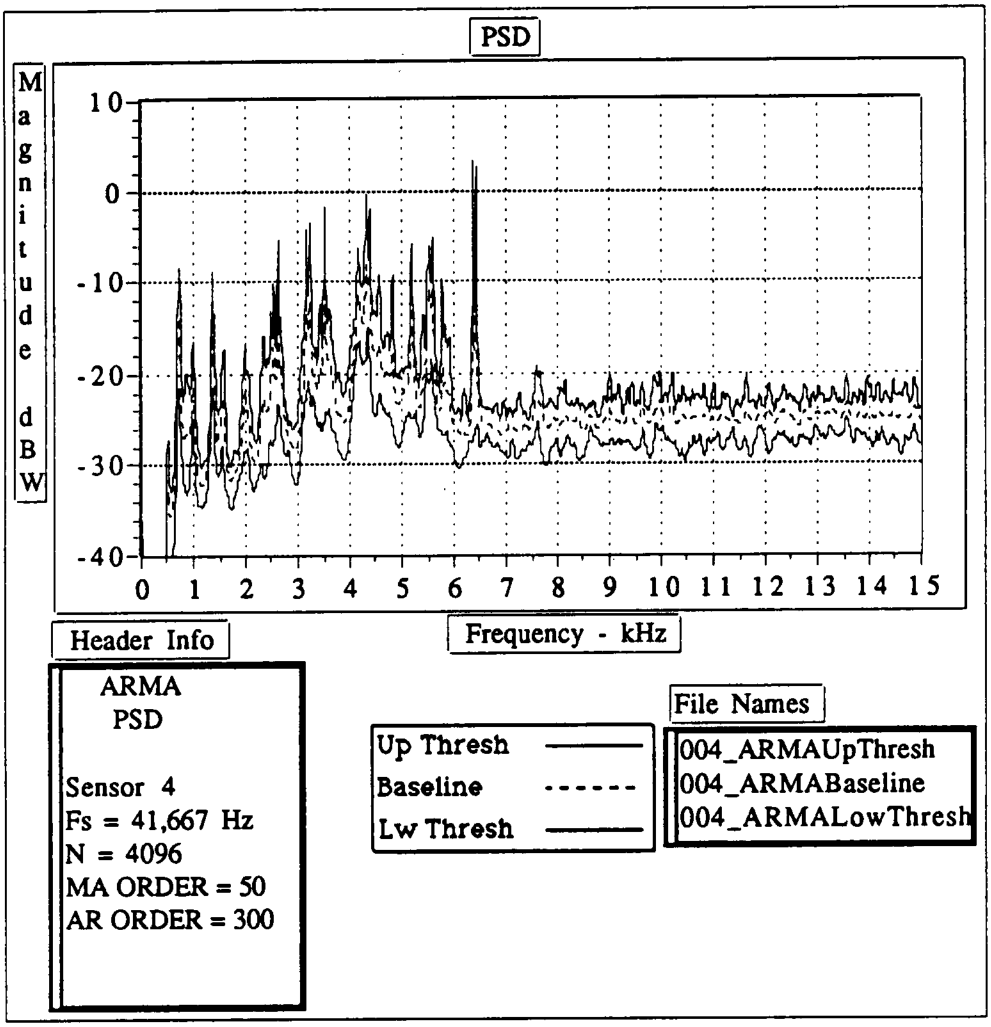


Figure 5.12. Overlay of Parametric ARMA PSD Plots of Baseline, Lower Threshold, and Upper Threshold of Sensor 4.

displayed and discussed. Comparing a single acoustic data file with these thresholds could lead to instructive analysis; however, simply overlaying all four plots on one graph would make the separate plots impossible to distinguish. It seems necessary, though, to be able to graphically compare the baseline and each threshold against a single file on one graph.

Figure 5.13 shows the front panel of "PSD Compare", a VI designed to compare single acoustic data files against appropriate baselines and thresholds. Subtracting the upper threshold, lower threshold, and baseline average from the file to be analyzed produces three, non-overlapping plots. These three are overlayed on the bottom graph labeled "Threshold Difference PSD" as discrete value plots. The top plot illustrates where the file exceeds the upper threshold; if the difference is negative for a particular index, the value assigned is zero. The middle plot displays both positive and negative differences between the file and the baseline average that do not exceed the thresholds. The bottom plot shows only where the file dips below the lower threshold. Both the upper and lower plots are shifted to adjust for the maximum and minimum value of the middle plot.

The top graph in Figure 5.13 is labeled "File PSD" and plots the file alone to show its actual magnitude and shape. It is important to note that the plots on the lower graph distort the true shape of the file's PSD. In other words, the upper threshold is not really 10 dBW above the baseline average from zero to 15,000 Hz. Rather, it has a

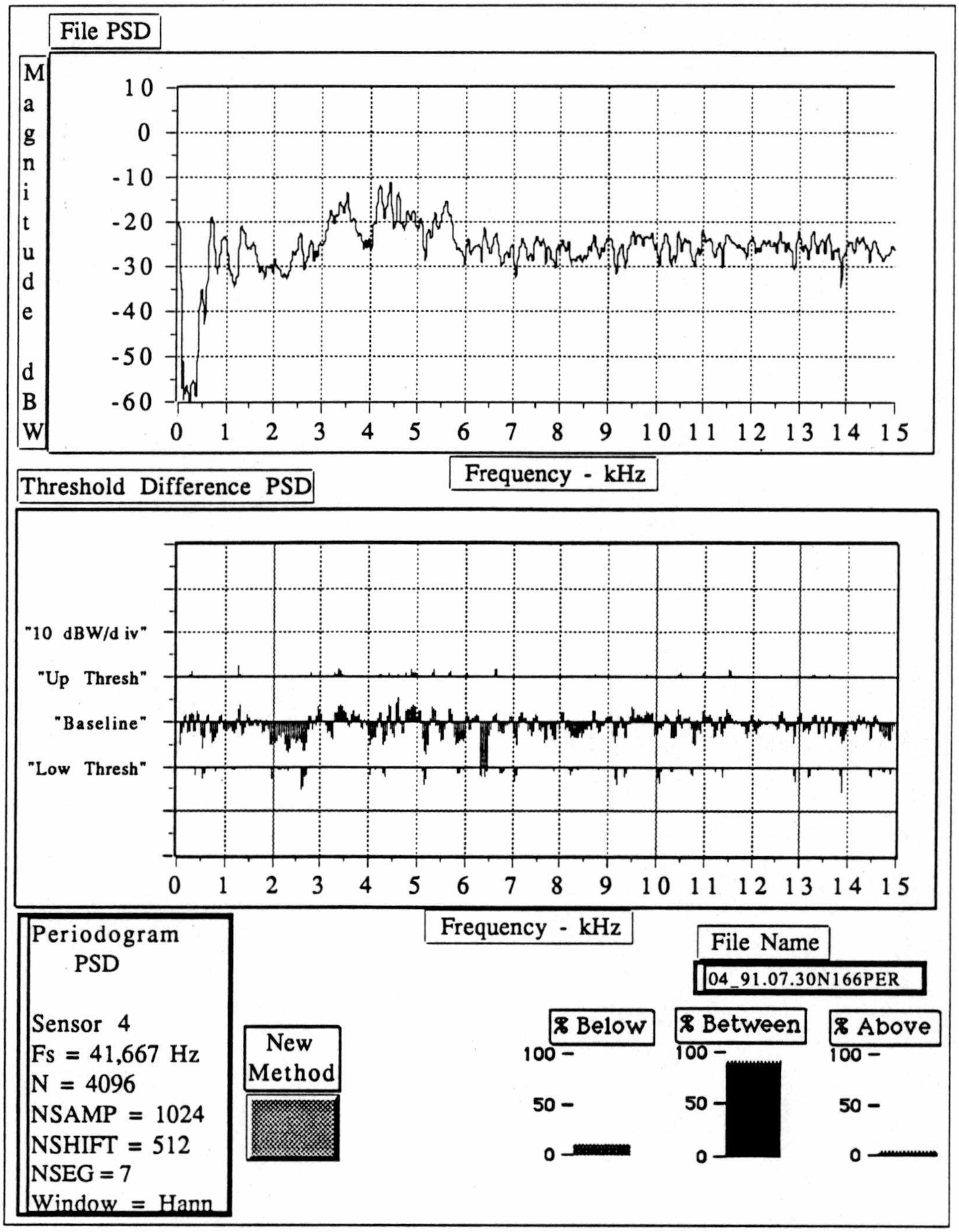


Figure 5.13. Comparison of Periodogram PSD of Sensor 4 on July 30, 1991, with Baseline and Threshold Plots.

unique value every 10 Hz. The bottom graph, however, is particularly useful in displaying at which frequencies the threshold or baseline is exceeded, and what the magnitude difference is in dBW. The three tank displays at the bottom of the figure indicate what percentage of the power of the signal falls below the lower threshold, between the two thresholds, and above the upper threshold. The button labeled "New Method" allows the user to change PSD methods by loading the appropriate thresholds into the graphs.

The acoustic data file being compared in Figure 5.13 is one recorded on July 30, 1991 on Sensor 4. The periodogram method was used to compute its PSD, and thus the periodogram thresholds are loaded in the graphs. The tank displays at the bottom of the figure indicate that almost all of the file's PSD falls between the thresholds; therefore, this file could be considered normal for Sensor 4. The bottom graph indicates that only trace amounts of power are found above and below the thresholds, and more than half of the file's PSD falls below the baseline average. The top graph of the signal's PSD looks characteristic of a normal signal on this sensor channel.

Figure 5.14 shows the periodogram PSD comparison of a file recorded the next day, July 31, 1991, on Sensor 4. The tank displays indicate that most of the file's PSD falls between the thresholds; however, a significant percentage of power is registered as being above the upper threshold. The bottom graph shows several 5 to 10

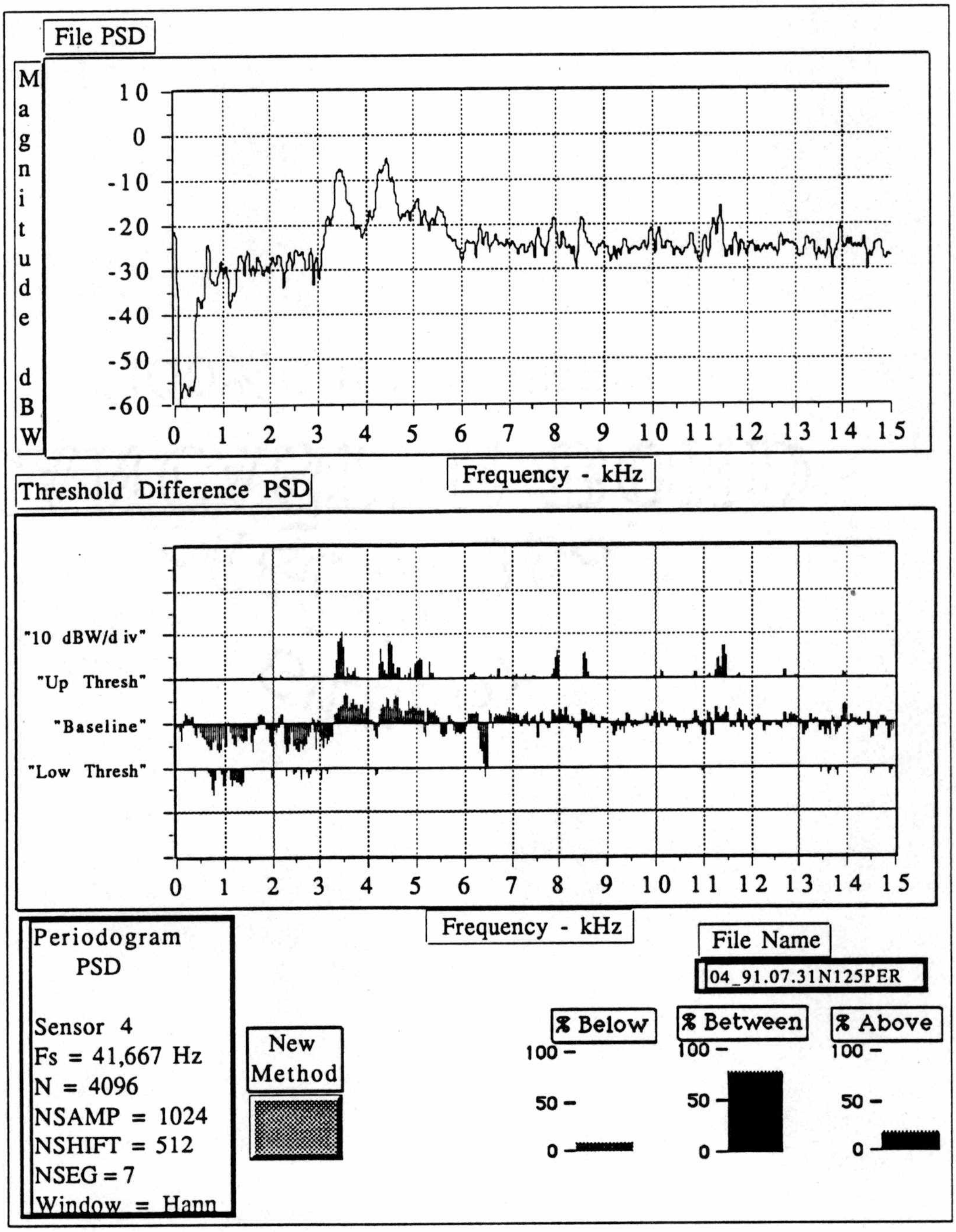


Figure 5.14. Comparison of Periodogram PSD of Sensor 4 on July 31, 1991, with Baseline and Threshold Plots.

dBW peaks above the upper threshold from 3,200 to 5,200 Hz, which indicates that this might not be a normal signal for this sensor. The most dramatic indicator that this might be a leak signature are the three peaks centered at 7,900, 8,600, and 11,400 Hz. Each are at least 7 dBW above the upper threshold, and the peak at 11,400 Hz has a significant area of surrounding power exceeding the threshold. These three have added significance because they are located in the higher frequency range (7,000 to 15,000 Hz) that rarely has any magnitude fluctuations more than 2 or 3 dBW above the baseline. The top graph of the signal's PSD no longer looks characteristic of a normal signal on this sensor channel.

Figure 5.15 shows the periodogram PSD comparison of a file recorded the following day, August 1, 1991, on Sensor 4. The tank displays indicate a dramatic increase in percentage of power exceeding the upper threshold, and a corresponding drop in percentage falling between the thresholds. The bottom graph indicates that virtually all the signal's power is above the baseline average from 3,000 to 15,000 Hz. The top plot shows that almost every significant peak from the day before has doubled to 10 to 20 dBW above the upper threshold, and the area of power surrounding each peak has grown as well. Several new peaks have appeared, the most significant being a 10 dBW peak at 14,100 Hz. Each peak appears much as the leak PSD signature Stulen [5] described. The two peaks centered at 6,600 and 8,000 Hz represent larger leaks than the two peaks at 11,400 and 14,100 Hz. Power falls off the

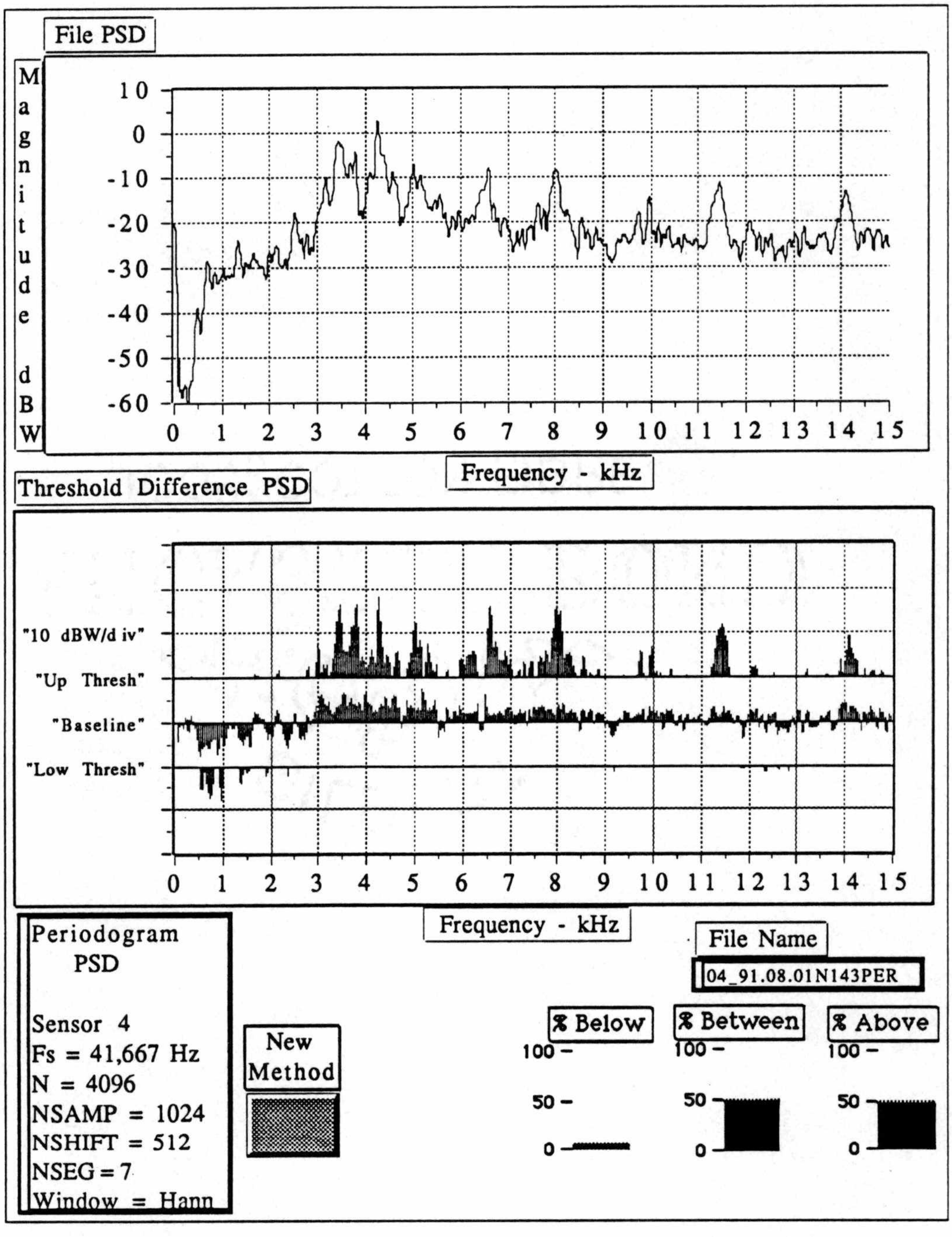


Figure 5.15. Comparison of Periodogram PSD of Sensor 4 on August 1, 1991, with Baseline and Threshold Plots.

center frequency gradually in the peaks at 6,600 and 8,000 Hz indicating a larger size leak. The two peaks at 11,400 and 14,100 Hz should represent leaks small in diameter because of their relatively high location along the frequency axis and because power falls off from the peak less gradually.

Figure 5.16 shows the periodogram PSD comparison of a file recorded several days later, August 5, 1991, on Sensor 4. Many of the peaks now exceed the upper threshold by 25 dBW. The presence of new peaks and the increase in magnitude of many of the peaks chart both the growth in size of the leak and possible impingements occurring over the four days since August 1. The top graph shows that the whole level of the spectrum has shifted up dramatically.

Figure 5.17 shows the periodogram PSD comparison of a file recorded eight days later, August 13, 1991, on Sensor 4. The size of the leak has continued to grow as indicated by the increase in magnitude of several of the peaks to almost 30 dBW above the upper threshold. It would seem, however, that the rate of growth has slowed somewhat in that the increase is not as great as between August 1 and August 5, and because very few new peaks have appeared.

The operator officially declared a leak condition in the superheat furnace on August 13, 1991. The unit was brought down the same day for repair as a non-scheduled outage. As Figure 5.18 shows, approximately nine feet from Sensor 4 a pinhole leak was found and repaired in the right side waterwall. The reported cause

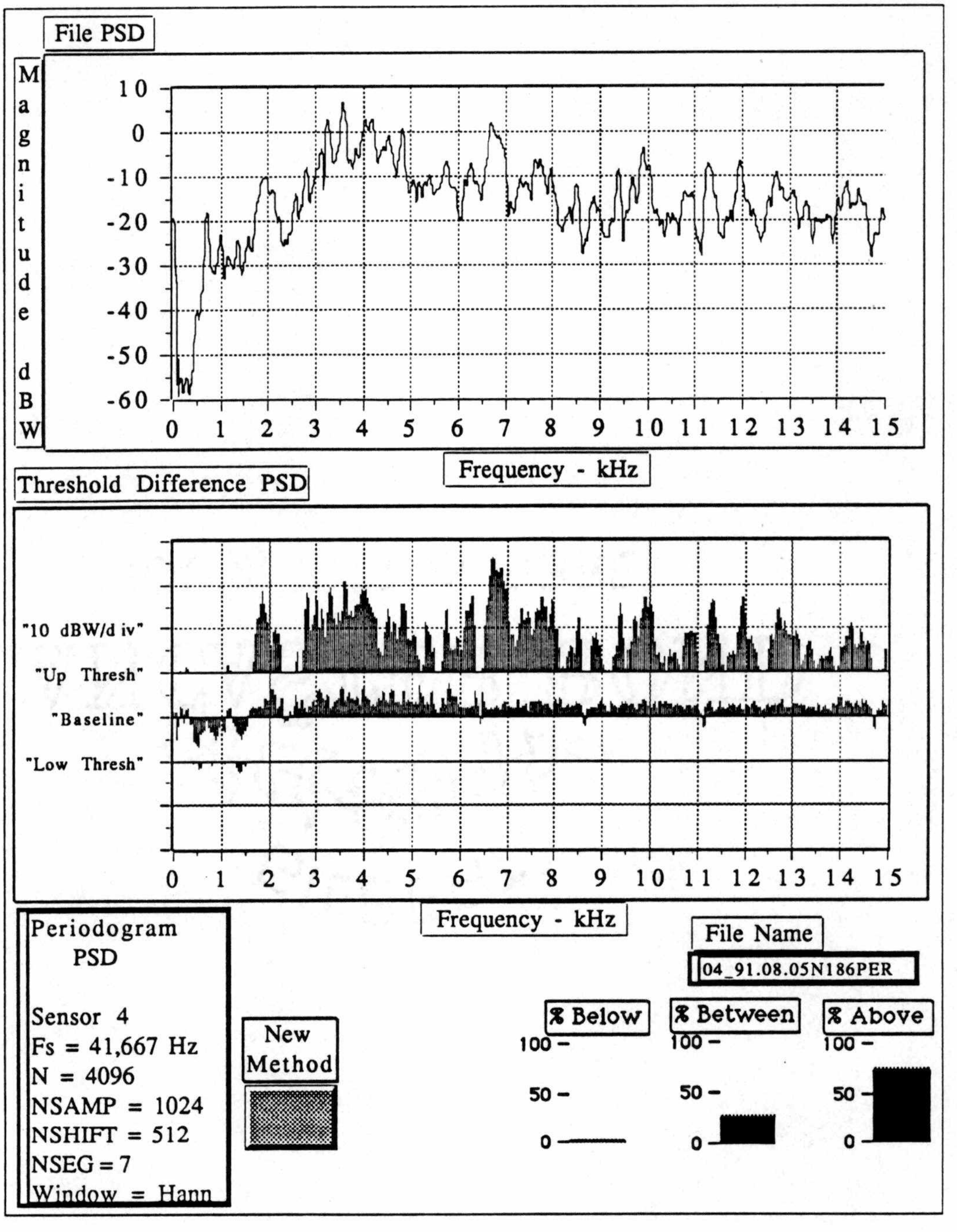


Figure 5.16. Comparison of Periodogram PSD of Sensor 4 on August 5, 1991, with Baseline and Threshold Plots.

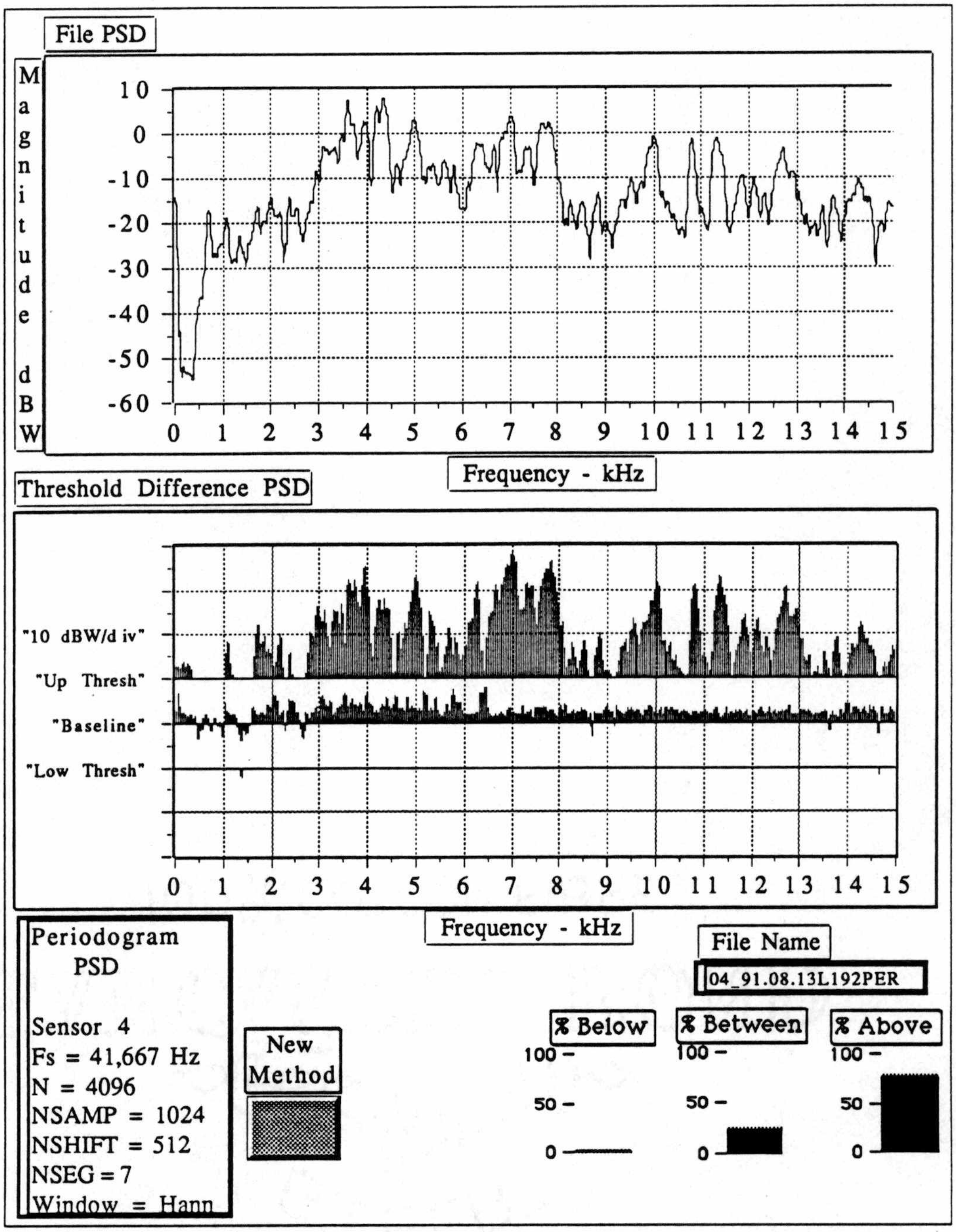


Figure 5.17. Comparison of Periodogram PSD of Sensor 4 on August 13, 1991, with Baseline and Threshold Plots.

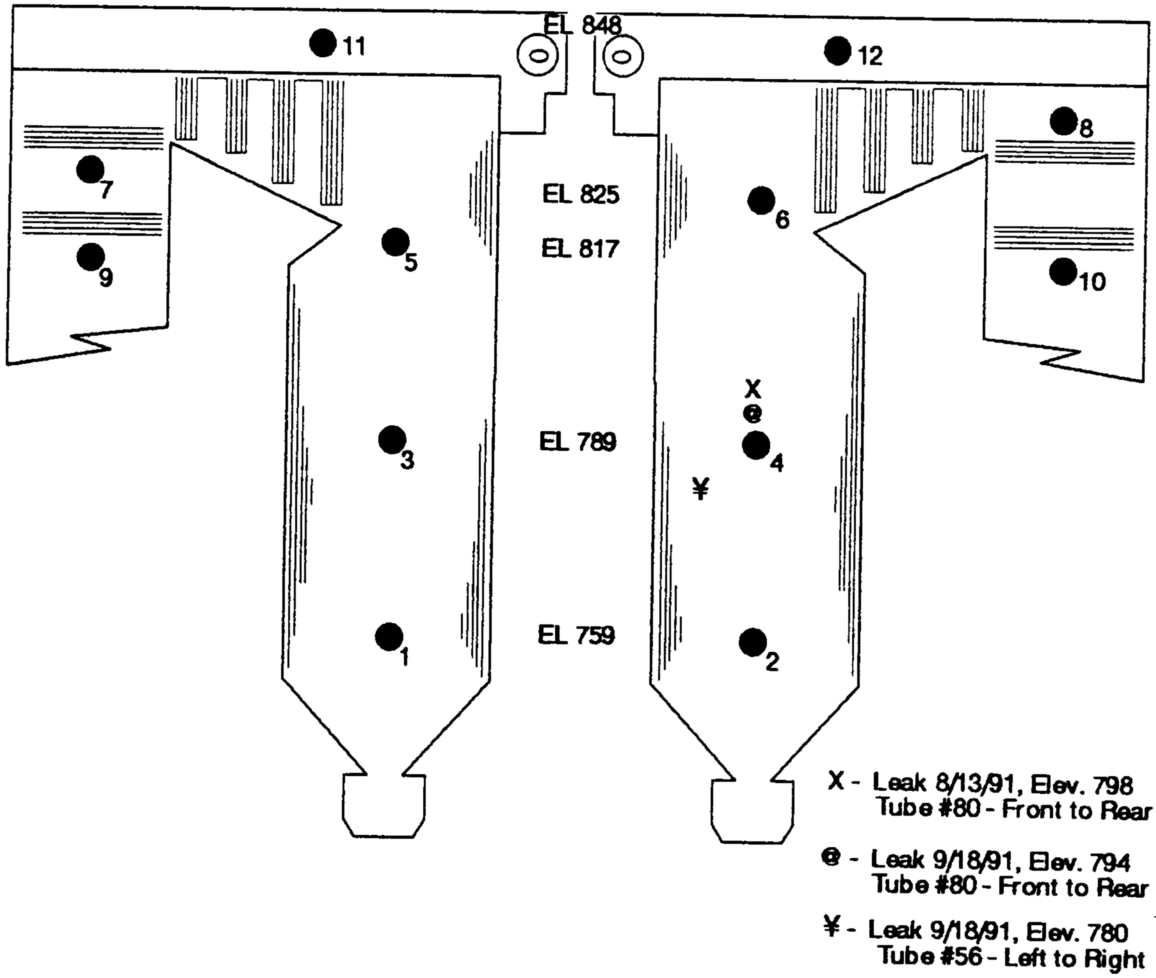


Figure 5.18. Leak Location Near Sensor 4 in Unit 5 Superheat Furnace.

Adapted from:

(a) R. D. Moss and R. L. Keyser, "An Introduction to the Tennessee Valley Authority Online Diagnostic Monitoring System Project," TVA publication TVA/PBO/R&D-89/8, Knoxville, TN, May 1989.

(b) "Boiler Tube Failure Report", TVA's Kingston Fossil Plant, Unit 5 Superheat Furnace, TVA Document TVA 7640-GEN (F&H PR 1/83), August 21, 1991.

(c) "Boiler Tube Failure Report", TVA's Kingston Fossil Plant, Unit 5 Superheat Furnace, TVA Document TVA 7640-GEN (F&H PR 1/83), October 3, 1991.

was a weld failure in an old field weld. This was the only leak found within 32 feet of Sensor 4 [22].

The unit was brought back on-line on August 16, 1991. No acoustic data was recorded for any of the sensors for the next two weeks. Recording began again on September 5, 1991. Figure 5.19 shows the periodogram PSD comparison of a file recorded September 5, 1991, on Sensor 4. The tank displays indicate a significant percentage of power exceeding the upper threshold, and most of the signal is above the baseline average. The multiple peaks and general raised PSD level between 2,700 and 8,100 Hz is very similar to the PSD's of the same sensor between August 1, 1991 and August 13, 1991. The shape of the two peaks at 9,600 and 10,000 Hz bear remarkable resemblance to the peaks at the same frequencies on August 13, 1991 (see Figure 5.17).

Figure 5.20 shows the periodogram PSD comparison of a file recorded September 16, 1991, on Sensor 4. The PSD level does not appear to have increased significantly since September 5, 1991. The peak at 1,600 Hz and the two or three peaks between 10,000 and 13,000 Hz have grown in magnitude, while the peaks between 9,000 and 10,000 Hz and the peak at 14,100 Hz have decreased somewhat.

The operator again declared a leak condition in the superheat furnace on September 18, 1991. The unit was brought down the same day for repair as a non-scheduled outage. As Figure 5.18 shows, approximately five feet from Sensor 4 a pinhole leak was found and repaired in the right side waterwall, very close to the

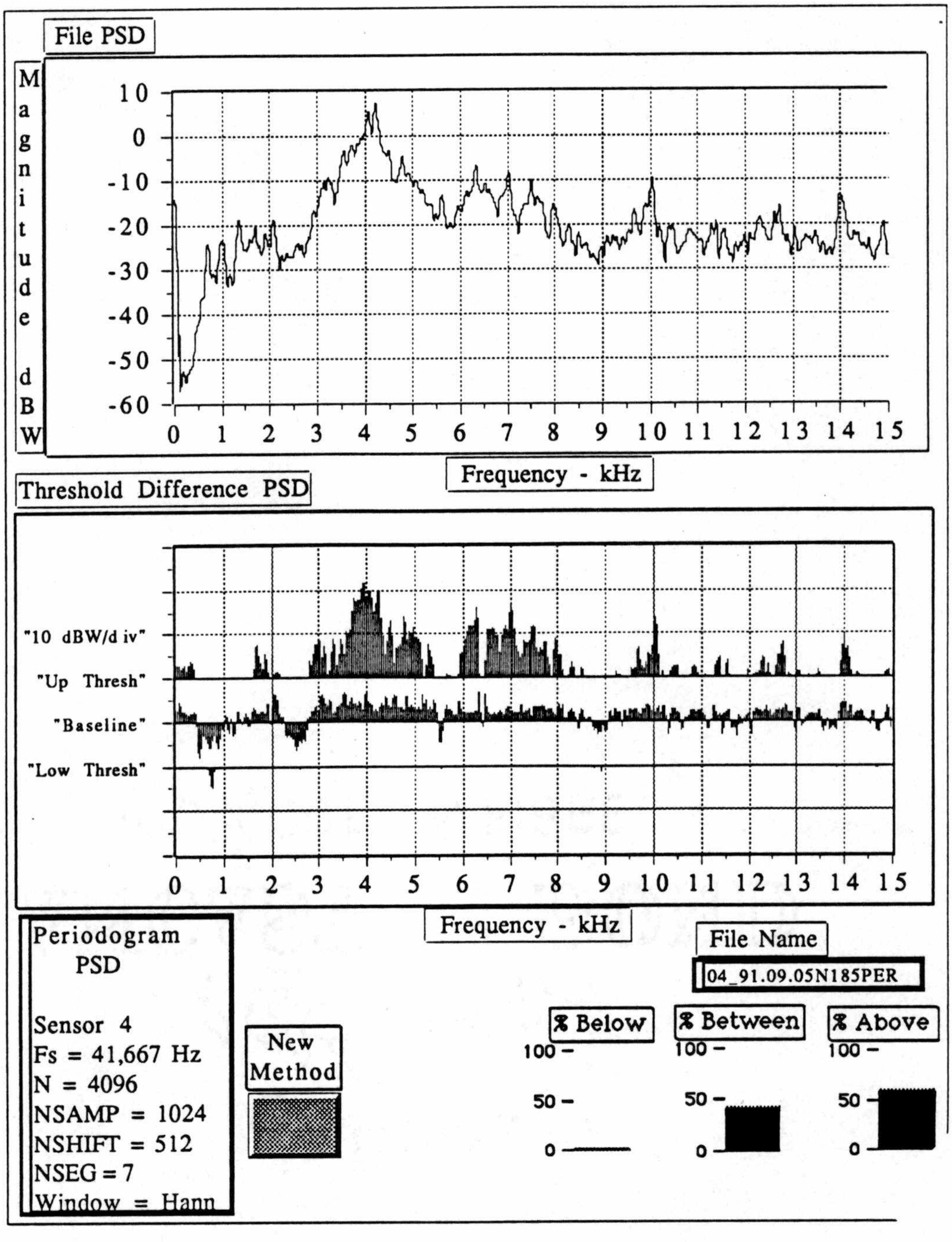


Figure 5.19. Comparison of Periodogram PSD of Sensor 4 on September 5, 1991, with Baseline and Threshold Plots.

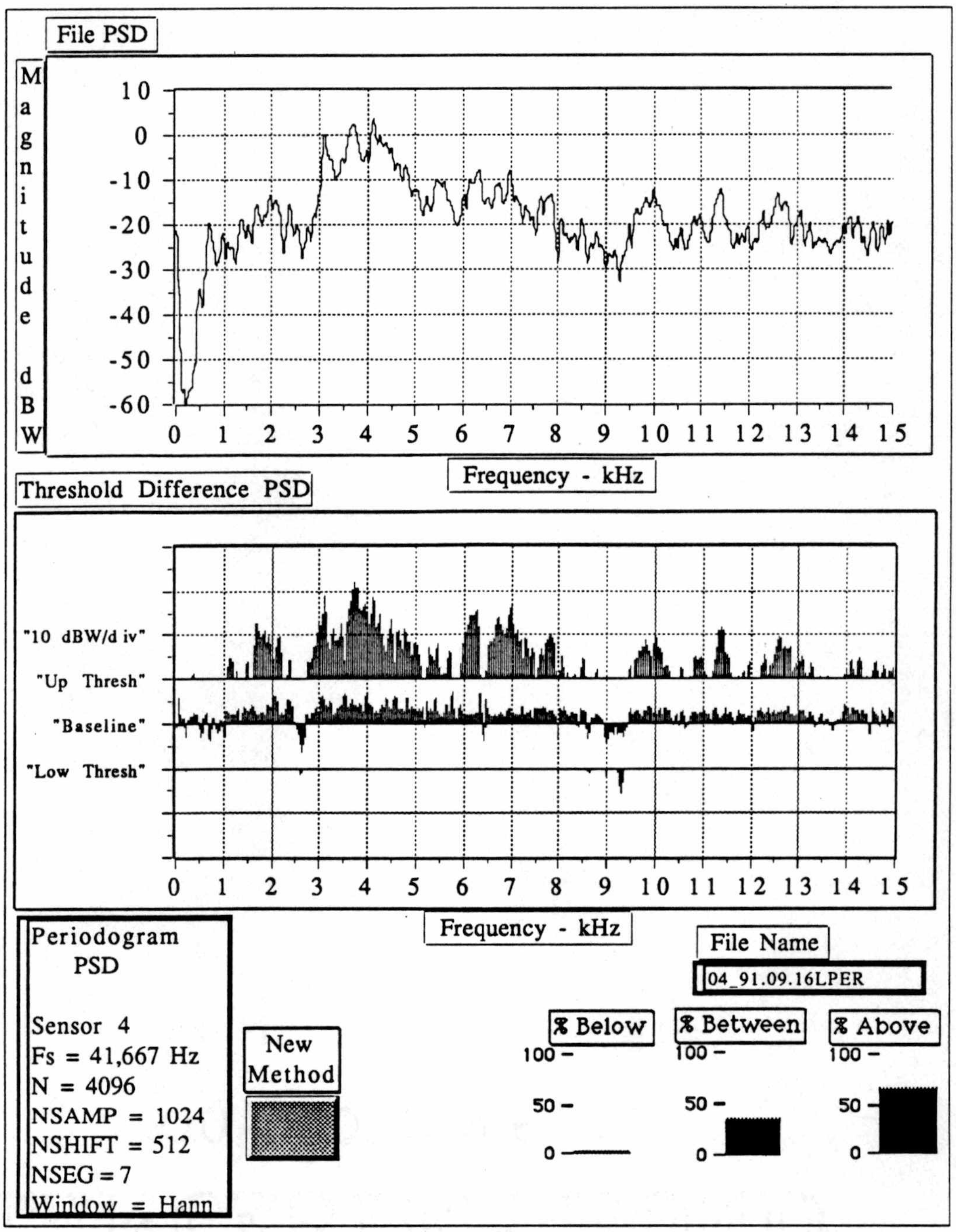


Figure 5.20. Comparison of Periodogram PSD of Sensor 4 on September 16, 1991, with Baseline and Threshold Plots.

previous leak repaired on August 13, 1991. The reported cause was a weld failure in a window weld. Another leak was found and repaired nine feet below and about 20 feet across the furnace on the front waterwall [23].

Figure 5.21 shows the periodogram PSD comparison of a file recorded after furnace repair on September 27, 1991, on Sensor 4. The PSD level has fallen to where most of the power falls below the baseline, yet within the upper and lower threshold limits. Trace amounts are below the lower threshold. The upper graph shows that the general shape of the PSD plot has altered slightly from the norm. The normal peaks between 4,000 and 6,000 Hz have dropped five to 10 dBW, and the peak normally appearing centered at 3,200 Hz is absent. It reappears however, in recordings taken a few days later.

D. Method Comparison

Considering the preceding figures, the leak condition recorded on Sensor 4 probably began on July 31, 1991, and was unmistakable by August 1, 1991. Since leak detection is primarily concerned with leak inception, the two files recorded on July 31, 1991, and August 1, 1991 can serve as the primary files by which to compare the different PSD methods developed for this project.

Figure 5.22 and Figure 5.23 show the correlogram PSD comparison on Sensor 4 of files recorded July 31, 1991 and August 1, 1991, respectively. The bottom graph is generally the same as the periodogram PSD comparison for the same day in each case (see

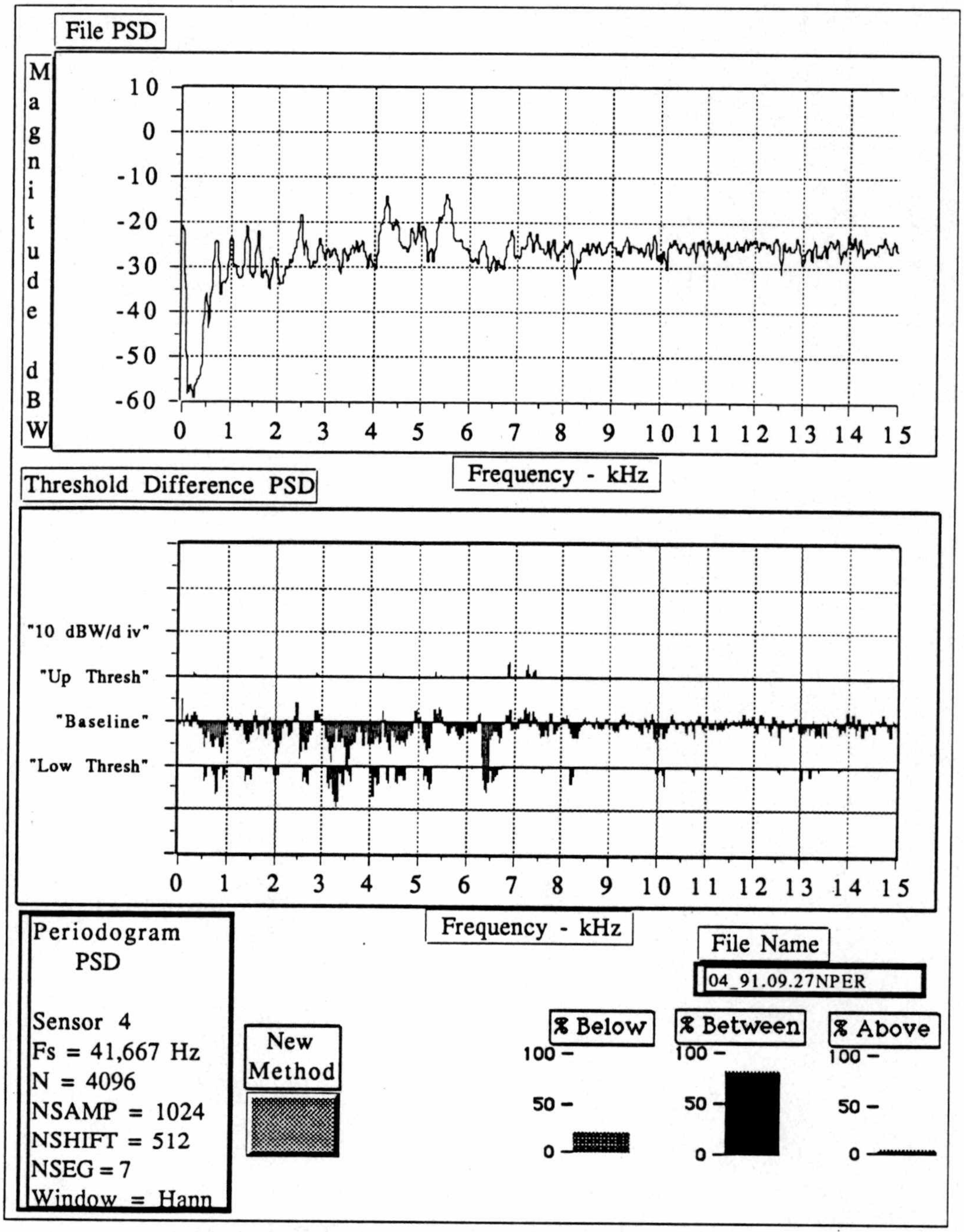


Figure 5.21. Comparison of Periodogram PSD of Sensor 4 on September 27, 1991, with Baseline and Threshold Plots.

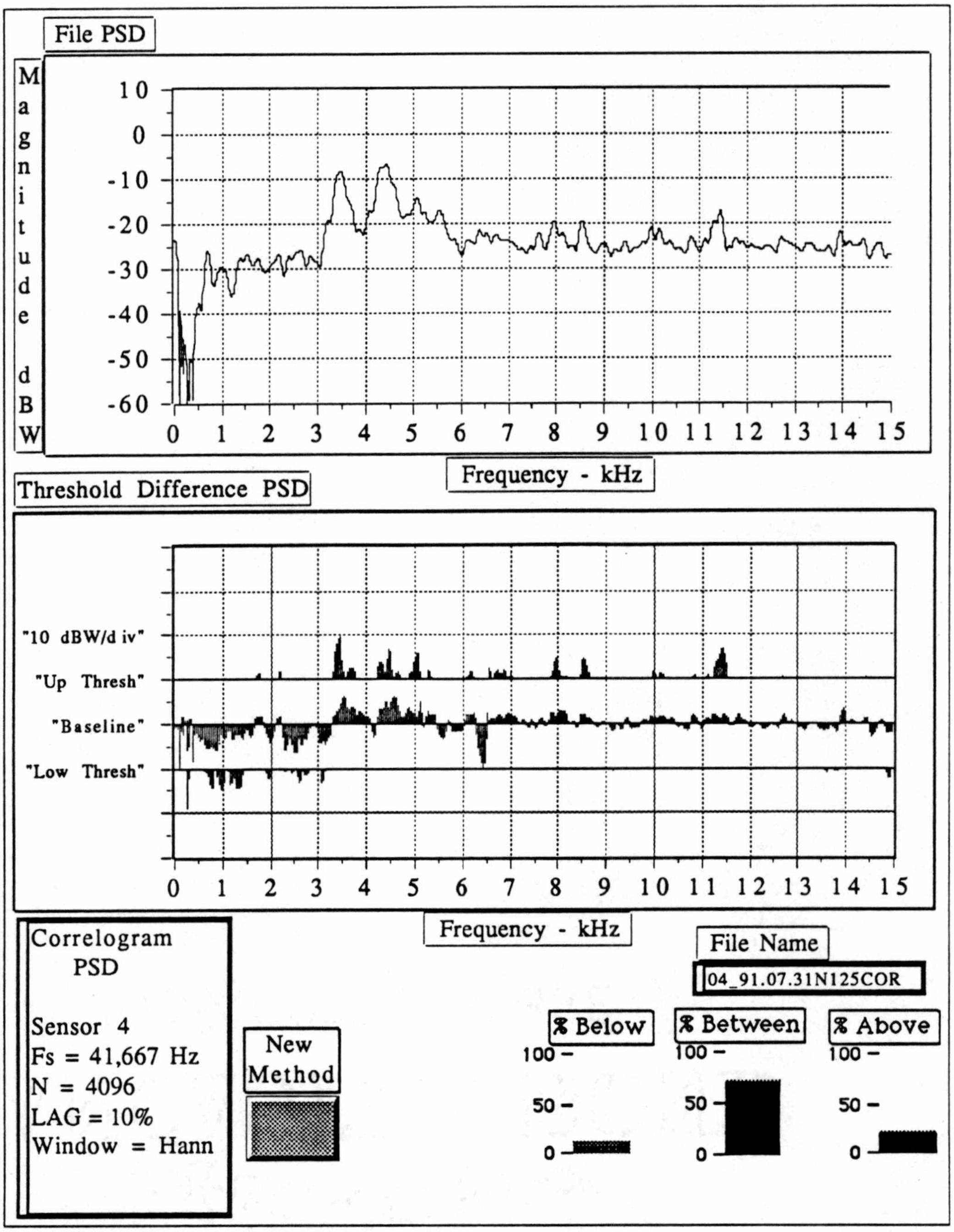


Figure 5.22. Comparison of Correlogram PSD of Sensor 4 on July 31, 1991, with Baseline and Threshold Plots.

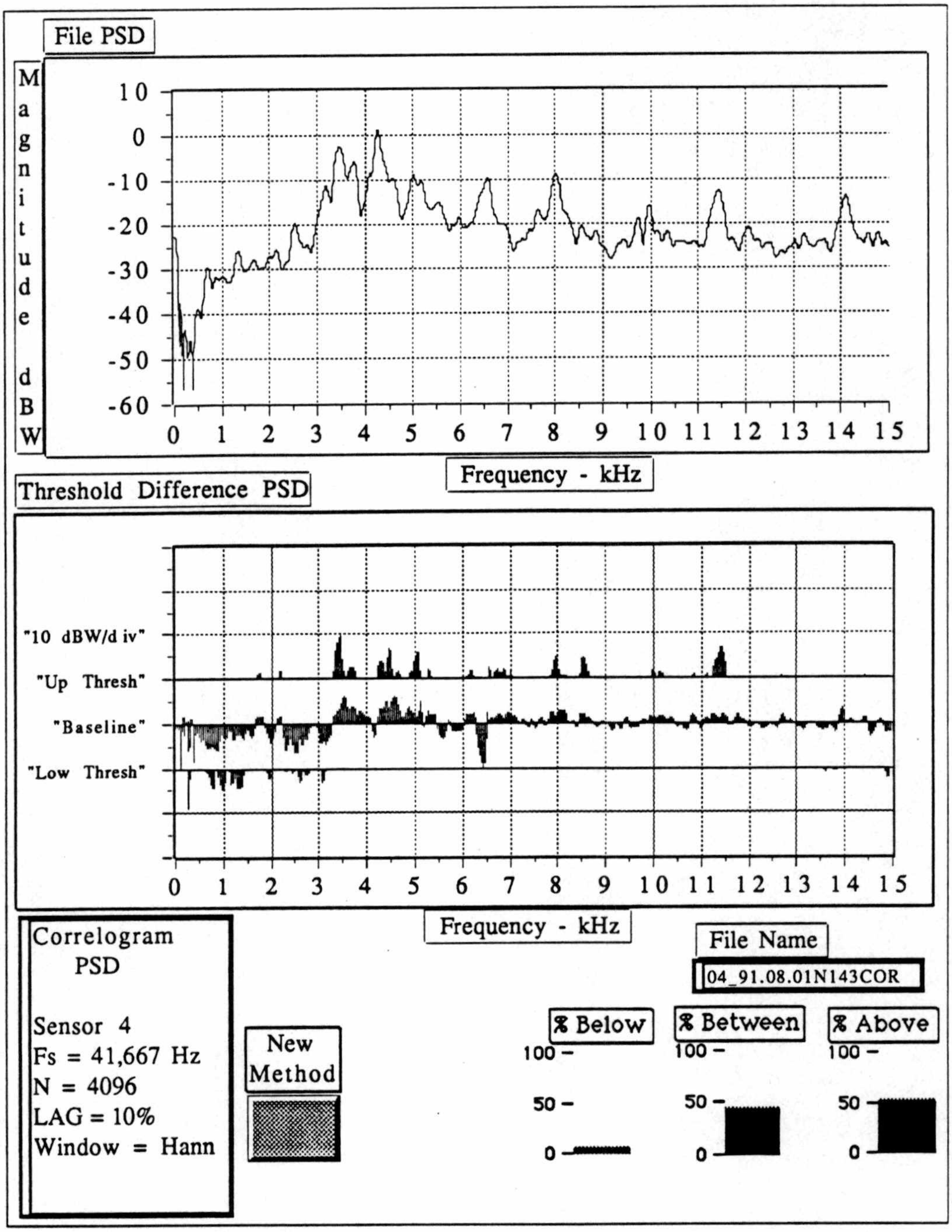


Figure 5.23. Comparison of Correlogram PSD of Sensor 4 on August 1, 1991, with Baseline and Threshold Plots.

Figure 5.14 and Figure 5.15). Each peak, however, is slightly lower in magnitude. The upper graph also shows the relative smoothness and smearing of peaks that is typical of the correlogram PSD method. Because of this, the correlogram method would probably signal a leak alarm slightly later than the periodogram method. Also, the correlogram method would be prone to smearing adjacent peaks together, thus possibly hiding a second leak signature.

Figure 5.24 and Figure 5.25 show the MA PSD comparison on Sensor 4 of files recorded July 31, 1991 and August 1, 1991, respectively. The bottom graph is much the same as the correlogram PSD comparison for the same day in each case (see Figure 5.22 and Figure 5.23). The general attenuation and smearing of peaks are even more significant with the parametric MA method than with the classical correlogram method. This suggests that the boiler's acoustic activity during a leak condition is not adequately represented by a moving-average time-series model. Because of this, the MA method would probably signal a leak alarm later than the periodogram method. The MA method would also be prone to smearing adjacent peaks together, thus possibly hiding a second leak signature.

Figure 5.26 and Figure 5.27 show the AR PSD comparison on Sensor 4 of files recorded July 31, 1991 and August 1, 1991, respectively. The bottom graph is similar to the periodogram PSD comparison for the same day in each case (see Figure 5.14 and Figure 5.15). Peaks are more finely resolved and amplified however, with the parametric AR method than with the classical periodogram

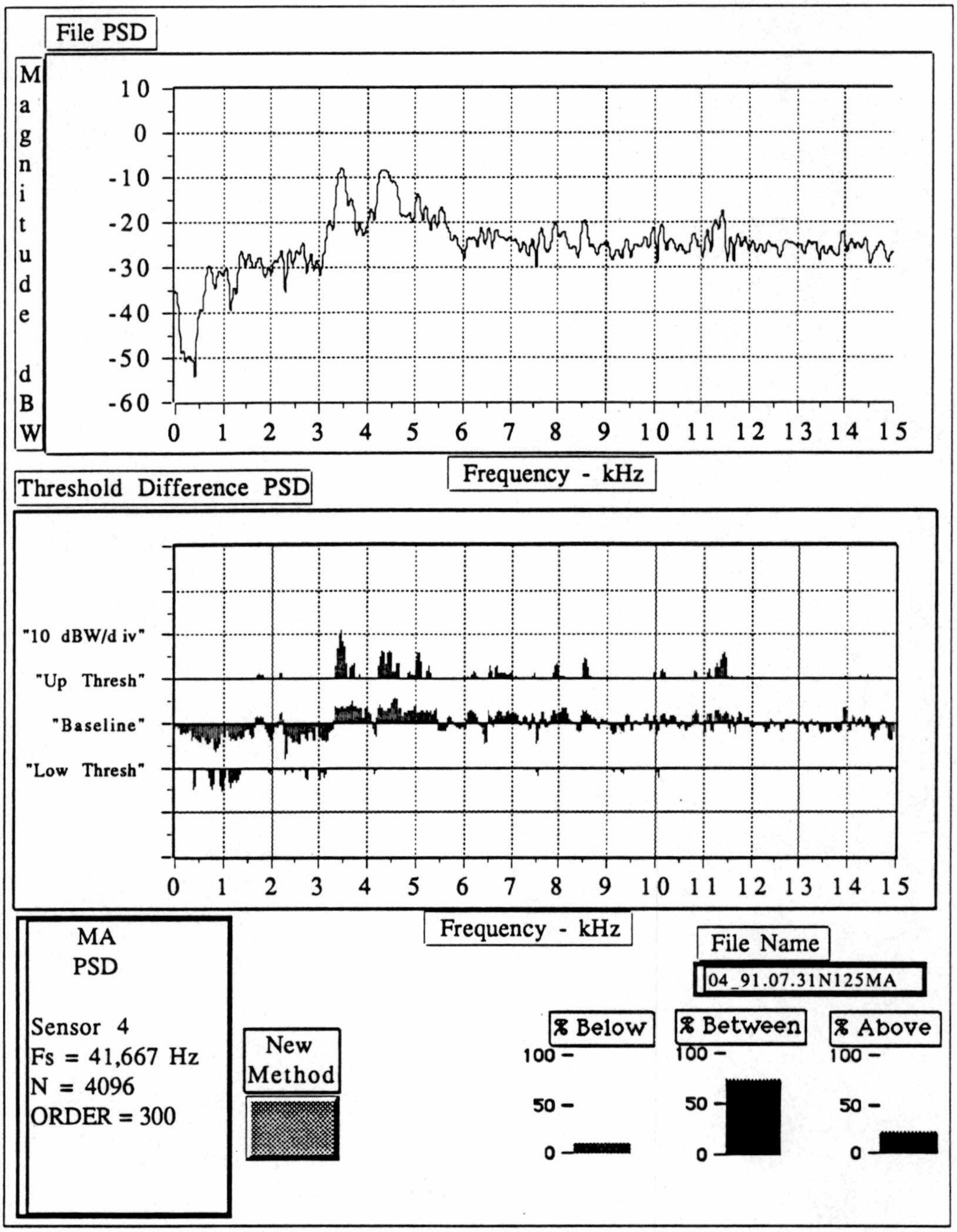


Figure 5.24. Comparison of MA PSD of Sensor 4 on July 31, 1991, with Baseline and Threshold Plots.

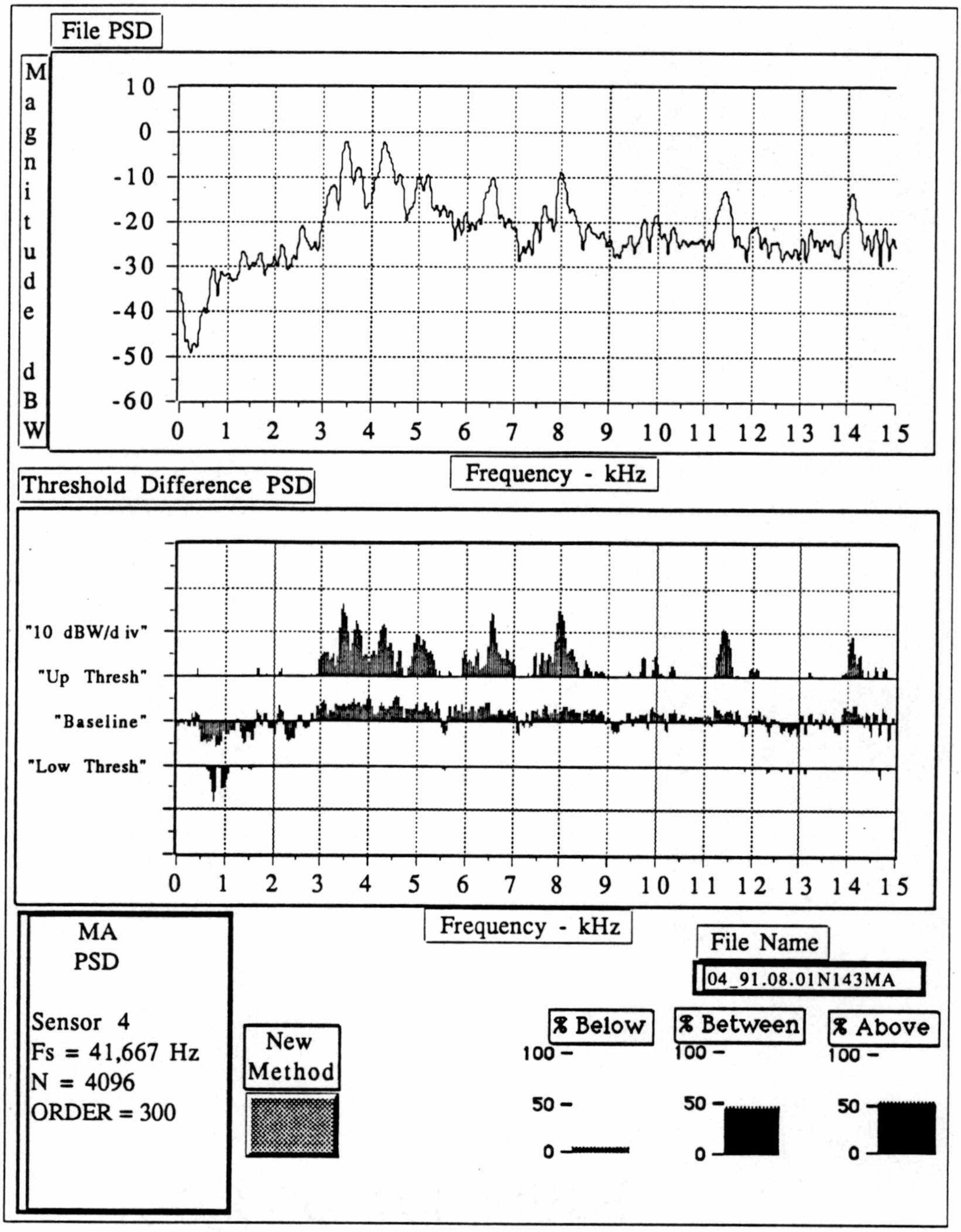


Figure 5.25. Comparison of MA PSD of Sensor 4 on August 1, 1991, with Baseline and Threshold Plots.

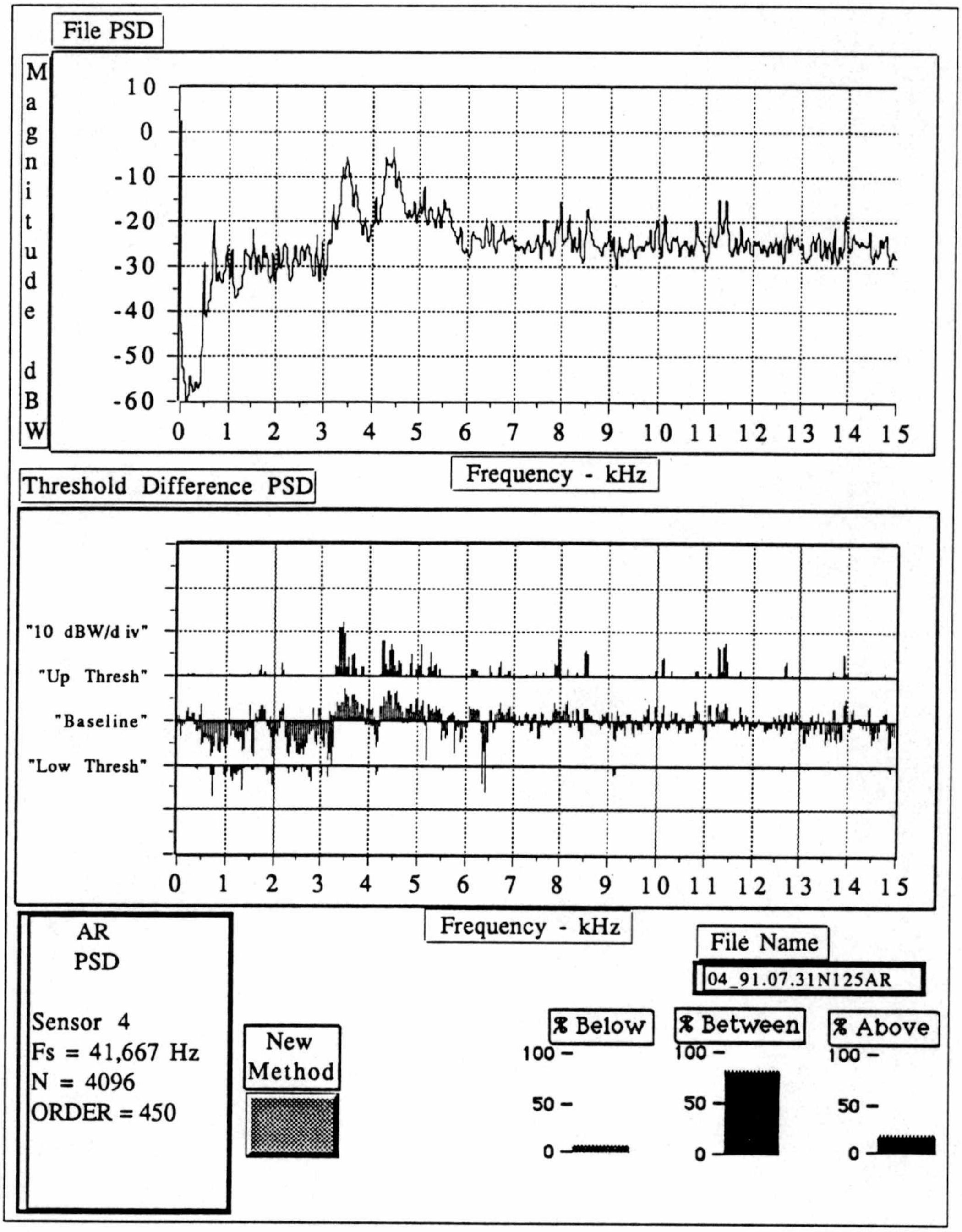


Figure 5.26. Comparison of AR PSD of Sensor 4 on July 31, 1991, with Baseline and Threshold Plots.

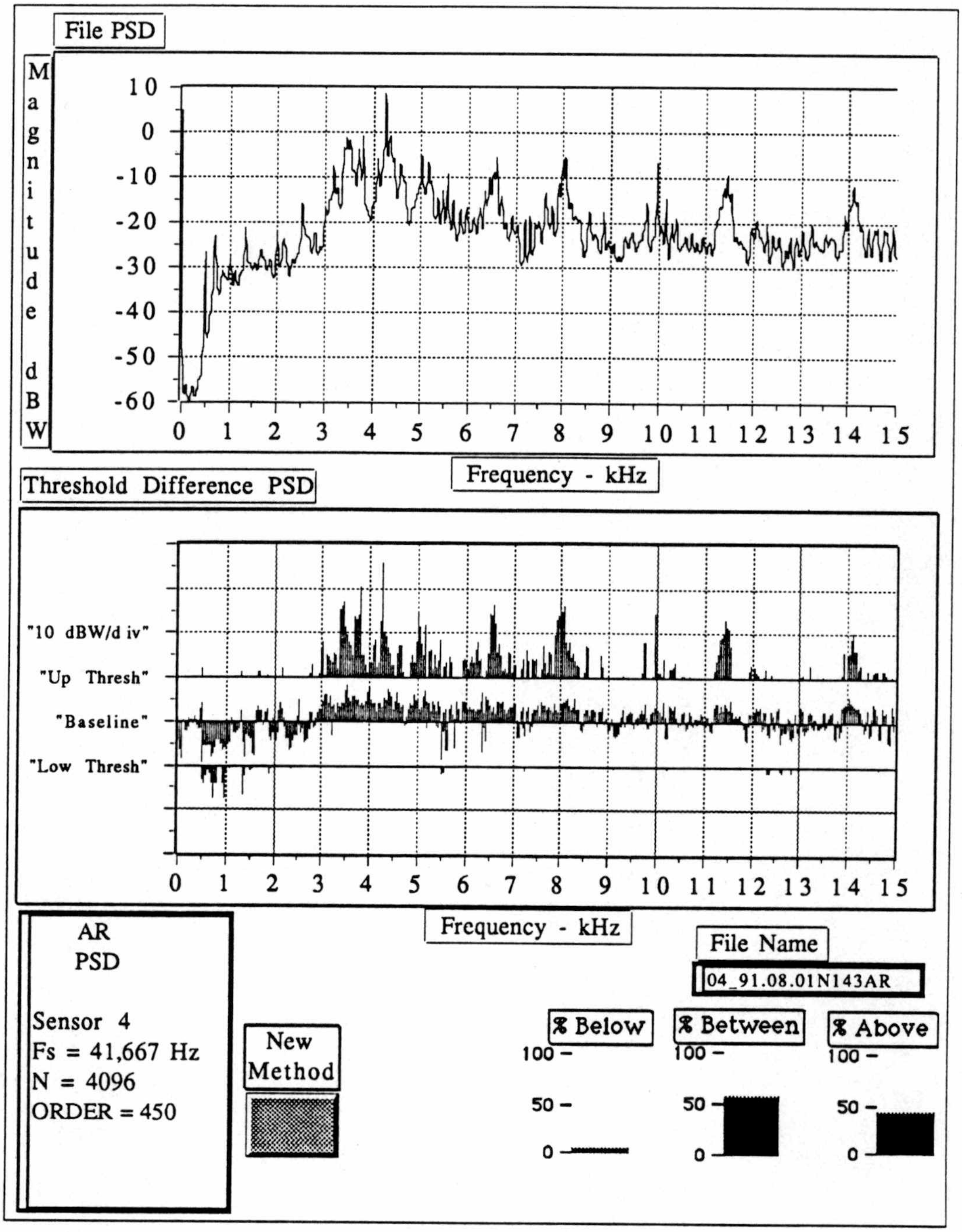


Figure 5.27. Comparison of AR PSD of Sensor 4 on August 1, 1991, with Baseline and Threshold Plots.

method. This suggests that the boiler's acoustic activity during a leak condition is more adequately represented by a auto-regressive time-series model. Because of this, the AR method would probably signal a leak alarm sooner than the periodogram method. The AR method however, would be more prone to signal a false leak alarm due to the greater variance in its PSD estimate.

Figure 5.28 and Figure 5.29 show the parametric ARMA PSD comparison on Sensor 4 of files recorded July 31, 1991 and August 1, 1991, respectively. The bottom graph is similar to the AR PSD comparison for the same day in each case (see Figure 5.26 and Figure 5.27). Peaks are even more finely resolved and amplified however, with the ARMA method than with the AR method. Also, the variation in the PSD estimate with the ARMA method is significantly less than with the AR method, as evidenced by comparing the two upper graph plots. This suggests that the boiler's acoustic activity during a leak condition is well represented by a auto-regressive moving-average time-series model. Because of this, the ARMA method would probably signal a leak alarm sooner than any of the other methods, and would be less prone to signal a false leak alarm due to the relatively small variance in its PSD estimate.

A VI, called "Detect", was developed that utilizes the data produced by the "PSD Compare" VI to detect a leak and signal an alarm to plant personnel. Figure 5.30 shows the connector pane, front panel, and block diagram of the VI "Detect". The connector pane illustrates that the input to the VI is the "Up Thresh Dif" array

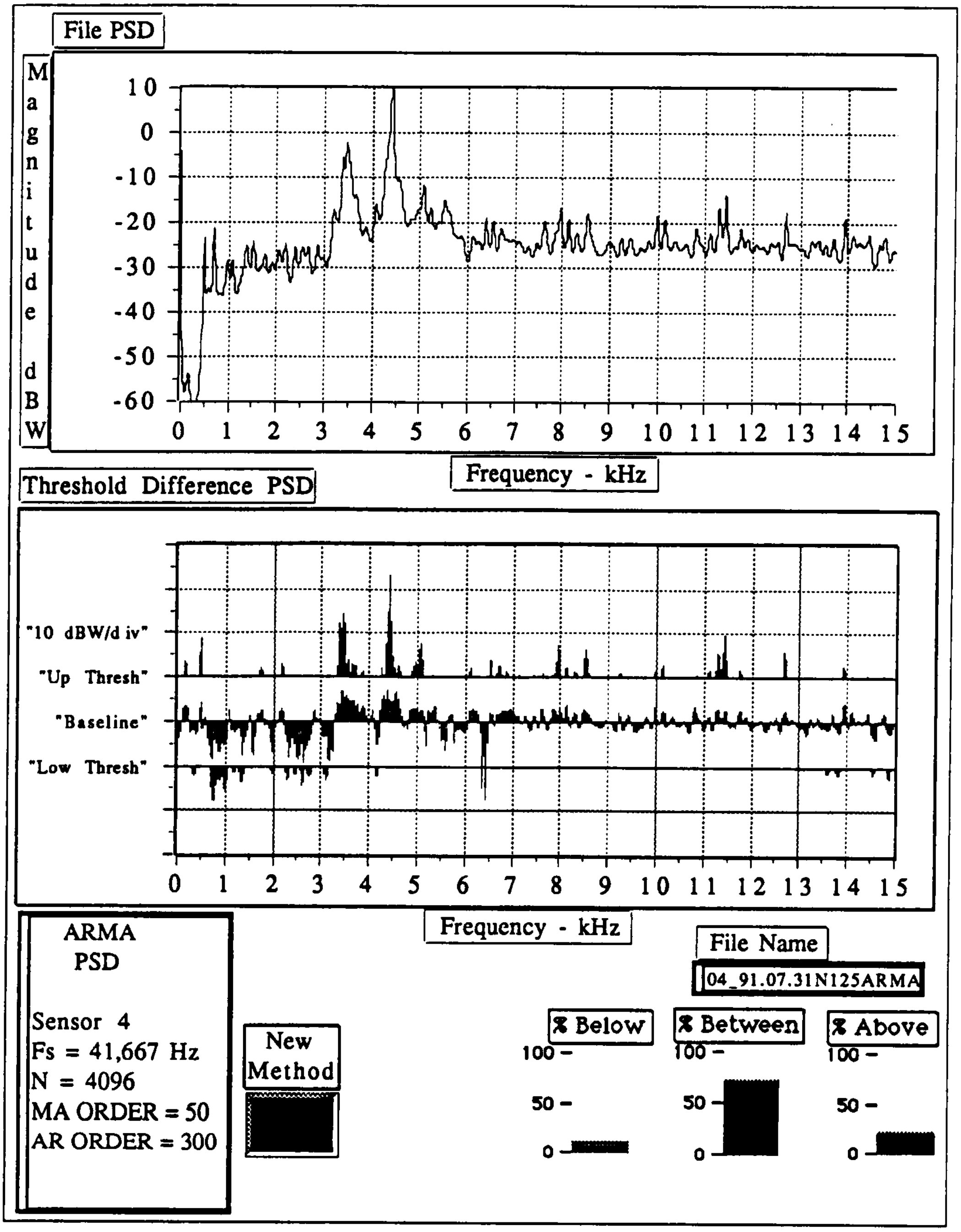


Figure 5.28. Comparison of ARMA PSD of Sensor 4 on July 31, 1991, with Baseline and Threshold Plots.

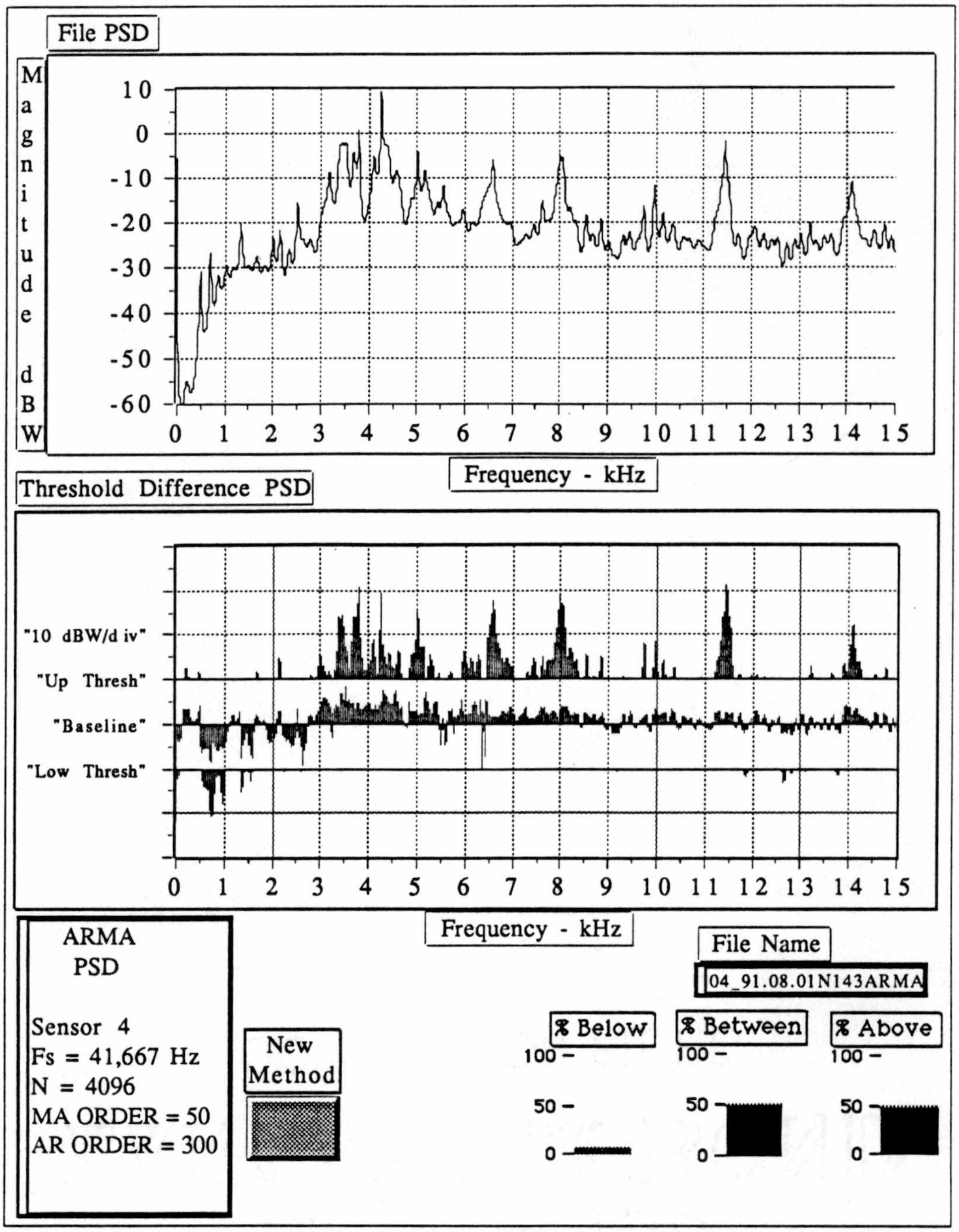


Figure 5.29. Comparison of ARMA PSD of Sensor 4 on August 1, 1991, with Baseline and Threshold Plots.

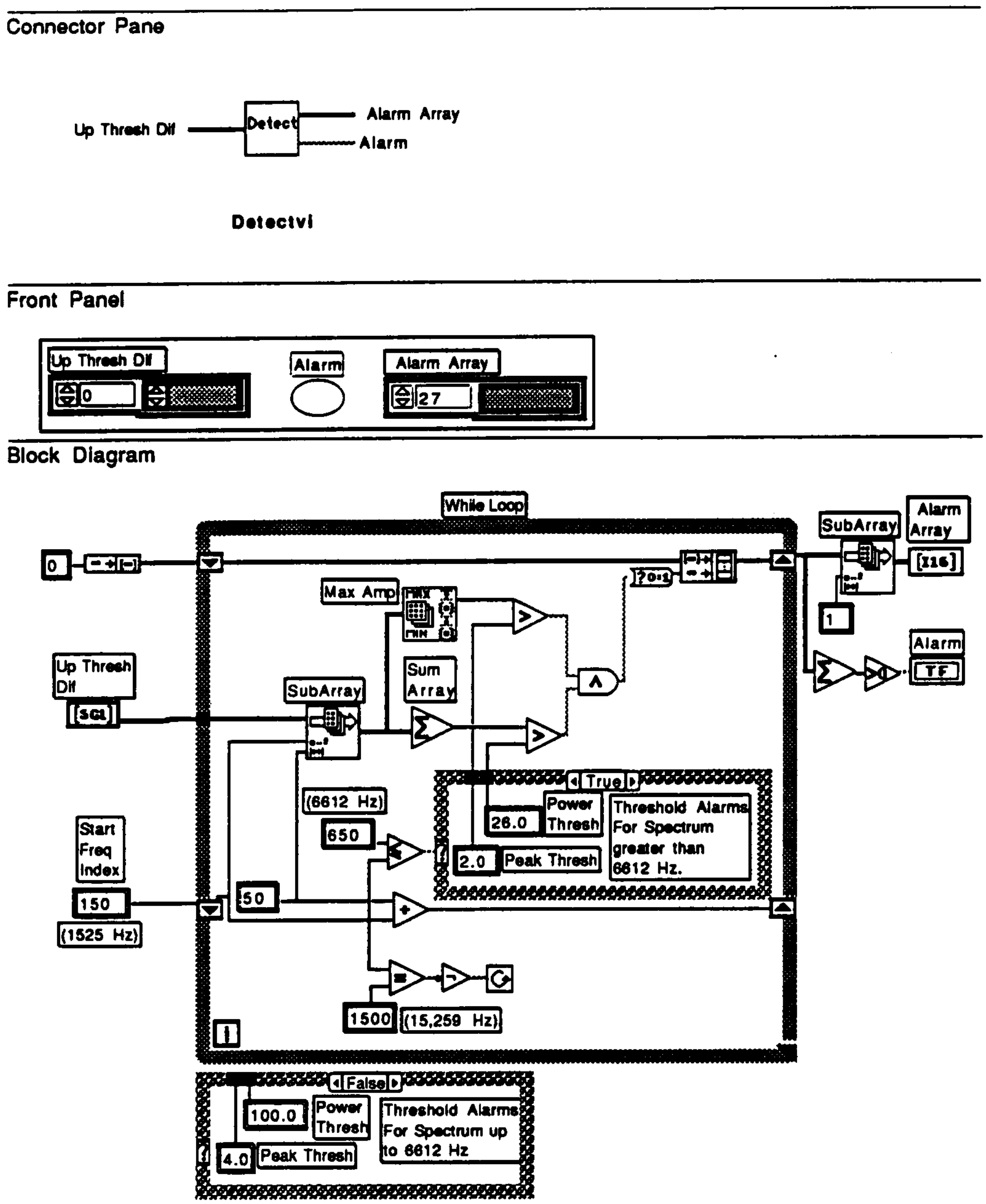


Figure 5.30. Connector Pane, Front Panel, and Block Diagram of "Detect" VI.

produced by the "PSD Compare" VI. This array is produced by subtracting the upper threshold PSD array of a particular sensor from the current PSD of the sensor. If the subtraction produces a negative value at a particular index, meaning the PSD did not exceed the upper threshold at that index, that element of the array is assigned a value of zero.

The block diagram at the bottom of Figure 5.30 shows the alarm algorithm of "Detect". The "Up Thresh Dif" array is passed into the while loop where it can be sequentially sectioned into 500 Hz segments from approximately 1500 Hz to 15,000 Hz. From each 500 Hz segment, a peak amplitude value is obtained from the subVI labeled "Max Amp", and the power surrounding each peak is calculated by the subVI "Sum Array". The case structure holds alarm thresholds for both the peak and the power surrounding the peak. If a particular 500 Hz section of the "Up Thresh Dif" array exceeds both the peak threshold and the power surrounding the peak threshold, it produces an alarm signal. The two different sets of alarm thresholds make the detection algorithm more sensitive to peaks occurring at frequencies above 6612 Hz, and less sensitive to the more volatile region below. These thresholds also make the algorithm insensitive to transient peaks in the spectrum that do not have much acoustic power surrounding them.

The "PSD Compare" and "Detect" VIs, working with any of the PSD estimation VIs studied, could form the heart of a new on-line leak detection system. Once placed on-line, the "Detect" VI's block diagram could be modified to adjust for sensitivity. For instance, instead of analyzing 500 Hz sections of the spectrum, 250 Hz sections could be considered to increase sensitivity. The alarm thresholds in each case could also be adjusted to increase or decrease the size of peak that causes an alarm.

E. Summary

In this section, a case study of two leak occurrences in proximity to Sensor 4 were used to demonstrate and analyze the leak detection capabilities of both classical and parametric PSD methods. The leak detection method developed for the analysis was a PSD comparison of individual acoustic data files with an average baseline, upper threshold, and lower threshold unique for each PSD method.

The two leak occurrences were officially declared on August 13, 1991 and September 18, 1991. Both were leaks that opened around old welds in the waterwall pipes. The numerous peaks appearing on the spectrum were probably due to several pinholes in the opened weld. Whatever the cause, the leak detection method developed and presented in this section showed clear evidence of a leak condition near Sensor 4 as early as July 31, 1991. This represents a 13-day improvement over the method employed by the furnace operator, and a one-day improvement over the existing NDT leak detection system.

Each PSD method used showed clear evidence of a leak condition on July 31, 1991; however, after comparing the graphs produced by each method, the ARMA PSD estimate method showed the most potential to be the earliest, and most reliable, for leak detection. Its unique combination of peak resolution and amplification, and relatively smooth and stable spectrum show the most promise in detecting all types of boiler tube leaks.

A leak detection VI for a sensor channel was presented based on the PSD methods analyzed and compared in this section. Conclusions of this research and suggestions for the future direction of the BAM project will be given in the next section.

VI. CONCLUSIONS

The concern of this research was to develop a leak detection program, or VI, utilizing spectral analysis, and to compare the performance of both classical and parametric methods of PSD estimation in this program. A spectral comparison VI, "PSD Compare", was developed that compares a sensor channel's PSD estimate against an upper threshold, lower threshold, and baseline average believed to be characteristic of the sensor channel during normal operation. As has been shown in the case study of the previous section, the VI developed indicates that a leak can be detected at least a day earlier than the existing on-line leak detection system at the Kingston plant. This is significant, since one day could prove to be the difference between a scheduled shutdown during low demand, or a forced outage during peak demand. A one-day improvement might also limit the leak's impingement on other tubes.

It has been concluded that all the PSD estimation methods studied, both classical and parametric, perform well enough in the leak detection VI to produce a one-day improvement over the existing system. The parametric ARMA method however, shows the most promise for detecting leaks the earliest in an on-line scenario. This on-line testing is crucial, of course, since the leak detection VI has only been tested retrospectively on one sensor with data that was recorded once a day.

It is also important to note that any future leak detection system should consider frequencies up to at least 15,000 Hz. As noted in the case study of the previous section, when the leak first appeared in the spectrum on July 31, 1991, it had a significant peak and surrounding power at 11,500 Hz. The next day, August 1, 1991, showed another significant peak at 14,000 Hz. The present on-line system only considers frequencies up to 10,000 Hz.

Any leak detection system must have a time delay before signalling an alarm after a threshold has been exceeded. This is necessary to avoid sending false alarms to plant personnel due to transient peaks in a sensor channel's spectrum. Therefore, additional savings in a new leak detection system could be achieved by using only one or two antialiasing filters and monitoring the 24 sensor channels sequentially. The present on-line system has a separate and expensive antialiasing and processing hardware module for each sensor channel. This simultaneous monitoring of every channel is unnecessary due to the aforementioned time delay inherent in any leak detection system.

Future study, utilizing a new software-intensive leak detection system like the one proposed, could investigate several possibilities to improve leak detection sensitivity. One possibility is that the overall spectral power level follows the pressure level in the boiler tubes. Another is that spectral fluctuations may be due to changes in the combustion rate. The OLDMS could supply these values to the leak detection system, and when an acoustic data file is saved, these values could be saved as part of the header information.

The research which has been presented here has been a limited project in terms of characterizing the spectral signature of boiler tube leaks. One sensor was used as a case study to develop a new leak detection system, but much remains to be done in terms of on-line testing and analysis to modify it into a robust, working system. It does look promising however, that an improved leak detection system is possible using the techniques presented in this paper.

LIST OF REFERENCES

LIST OF REFERENCES

1. C. S. Jones, D. B. Koch and K. W. Morris, "Classical and Parametric Spectral Analysis Using a Visual Programming Language for Tube Leak Detection," in *Proc. Eighth Power Plant Dynamics, Control, and Testing Symposium*, Knoxville, TN, May 1992.

2. "Technical Manual for the OLM-100 Acoustic Boiler Tube Leak Detection System", NDT International, Inc., West Chester, PA, January 1986.

3. R. D. Moss and R. L. Keyser, "An Introduction to the Tennessee Valley Authority Online Diagnostic Monitoring System Project," TVA publication TVA/PBO/R&D-89/8, Knoxville, TN, May 1989.

4. A. A. Putnam and L. Faulkner, "An Overview of Combustion Noise," Journal of Energy, November-December 1983, pps. 458-469.

5. F. B. Stulen, "A Critical Review of Acoustic and Acoustic Emission Leak Detection in Boilers," Proceedings of the Second EPRI Incipient-Failure Detection Conference, EPRI CS-4748, Palo Alto, CA, January 1986.

6. M. H. Lind, "Boiler Tube Leak Detection System," Proceedings of the Third EPRI Incipient-Failure Detection Conference, EPRI CS-5395, March 1987.

7. D. J. Snow and P. A. Ball, "Acoustic Leak Detection in Fossil Fueled Boilers," Proceedings of the Second EPRI Incipient-Failure Detection Conference, EPRI CS-4748, Palo Alto, CA, January 1986.

8. "Acoustic Leak Detection for Boiler Tubes," EPRI Technology Review, Monitoring and Diagnostic Center.

9. R. D. Litt, J. R. Scheibel, and D. J. Snow, "Acoustic Boiler Leak Detection Systems," Proceedings of the Third EPRI Incipient-Failure Detection Conference, EPRI CS-5395, March 1987.

10. F. B. Stulen, B. Salisbury, and J. R. Scheibel, "Leak Detection in Boilers," Conference on Boiler Tube Failures in Fossil Plants, EPRI Research Project 1890, November 1987.

11. LabVIEW 2 Training Manual, National Instruments Corporation, Austin, TX.

12. L. H. McCarty, "Software Simulates Instrumentation Systems," Design News, May 21, 1990, pps. 72-73.

13. J. Barber, "Data-Flow Programming in a Graphical Language," Power, November 1990, p. 141.

14. S. L. Marple, Jr., Digital Spectral Analysis With Applications, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1987.

15. A. V. Oppenheim and R. W. Schafer, Digital Signal Processing, Second Edition, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1989.

16. S. M. Kay, Modern Spectral Estimation, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1987.

17. G. M. Jenkins and D. G. Watts, Spectral Analysis and Its Applications, Holden-Day, Inc., Englewood Cliffs, NJ, 1987.

18. G. R. Cooper and C. D. McGillem, Probabilistic Methods of Signal and System Analysis, Second Edition, Holt, Rinehart and Winston, New York, NY, 1986.

19. P. D. Welch, "The Use of Fast Fourier Transform for the Estimation of Power Spectra", IEEE Transactions on Audio and Electroacoustics, Vol. AU-15, No. 2, June 1967, pps. 70-73.

20. R. B. Blackman and J. W. Tukey, The Measurement of Power Spectra from the Point of View of Communication Engineering", Dover Publications, Inc., NY, 1958.

21. H. Akaike, "Power Spectrum Estimation through Autoregression Model Fitting," Ann. Inst. Stat. Math., vol. 21, 1969, pps. 407-419.

22. "Boiler Tube Failure Report", TVA's Kingston Fossil Plant, Unit 5 Superheat Furnace, TVA Document TVA 7640-GEN (F&H PR 1/83), August 21, 1991.

23. "Boiler Tube Failure Report", TVA's Kingston Fossil Plant, Unit 5 Superheat Furnace, TVA Document TVA 7640-GEN (F&H PR 1/83), October 3, 1991.

VITA

Curtis S. Jones was born in Huntsville, Alabama on April 23, 1963. He graduated from Huntsville High School in 1981, and received a Bachelor of Science degree in Electrical Engineering from the University of Alabama in Huntsville in 1985. He worked as an engineer in Huntsville for New Technology Incorporated from 1986 to 1988, and for Teledyne Brown Engineering in 1989. He also worked as a campus minister for the Central Church in 1988. He began work toward a Master's degree in January 1990 at the University of Tennessee in Knoxville (UTK).

The author worked eighteen months as a graduate research assistant in the Electrical Engineering Department at UTK, and for one year as an electrical engineer at Mullican Lumber and Manufacturing Company of Maryville, Tennessee while continuing his graduate studies. Mr. Jones will continue work as an engineer for Mullican Lumber and Manufacturing Company after graduation.