Acoustic Anomaly Detection Using Robust Statistical Energy Processing

Farakh Nayaab John Salik

A Thesis

in

The Department

of

Electrical And Computer Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science (Electrical Engineering) at
Concordia University
Montreal, Quebec, Canada

August 2007

Your file Votre référence
ISBN: 978-0-494-40896-4
Our file Notre référence
ISBN: 978-0-494-40896-4

ABSTRACT

Acoustic Anomaly Detection Using Robust Statistical Energy Processing

Farakh Nayaab John Salik

An anomaly is the specific event that causes the violation of a process observer's expectations about the process under observation. In this work, the problem of spatially locating an acoustic anomaly is addressed. Once reduced to a problem in robust statistics, an automated observer is designed to detect when high energy sources are introduced into an acoustic scene. Accounting for potential energy from signal amplitude, and kinetic energy from signal frequency in wavelet-filtered sub-bands, an outlier a robust statistical characterization scheme was developed using the Teager energy operator. With a statistical expectation of energy content in sub-bands, a methodology is designed to detect signal energies that violate the statistical expectation. These *minor anomalies* provide some sense that a fundamental change in energy has occurred in the sub-band. By examining how the signal is changing across all sub-bands, a detector is designed that is able to determine when a fundamental change occurs in the sub-band signal trends. Minor anomalies occurring during such changes are labeled as *major anomalies.* Using established localization methods, position estimates are obtained for the major anomalies in each sub-band. Accounting for the possibility of a source with spatiotemporal properties, the median of sub-band position estimates provides the final spatial information about the source.

ACKNOWLEDGEMENTS

For his guidance and support in completing this academic work, I wish to thank my supervisor Dr. Mojtaba Kahrizi. Also, for inspiring me to publish and for giving me academic advice, I wish to thank my friend Bill Pontikakis.

DEDICATION

My life as an engineer has by now spanned seven years. In that time, my academic life has crossed over into my professional life sometimes enriching my experiences, and sometimes depleting them. The completion of this work marks a positive milestone in my career and highlights another personal achievement attributed to hard work. I wish to take this opportunity to thank those that have positively contributed to my growth as both a person and as an engineer. Through your support this thesis was possible, which gives you partial ownership in its completion.

Mukhtar & Masih Salik Mother and Father. Simple words can never fully express my gratitude. Thank you for maintaining the spirit and body of an absent son. As you know, this work was completed through great turbulence, but your voice and gentle presence always told my heart when I was *home*. Thank you for the monumental support that amounted from the little things that meant everything to me. With no uncertainty, this work would not have been possible without you. *Part of this thesis is yours.*

Bonnie Roussy For the years of my advanced studies, you have never failed to make me laugh. Your gentle demeanor softened my heart and inspired me to be a better man. My professional conduct has only heightened since I met you since you taught me the value of my humanity when it was dampened by wavering moral in the face of short deadlines, and difficult meetings. You are the sparkle in my eye, and the spirit in my laughter. Thank you for consistently being being there for me as my companion, my best friend, and my great love. *Part of this thesis is yours.*

Bill Pontikakis More than a peer or friend, I now consider you a brother. Our common plight bound us in a way that will not be forgotten. Your advice during my studies has always been good and over the years, your honesty has not only been helpful, but humorous as well. A voice in the night, no other person could have punctuated the solitude or monotony any better. You are funny and spirited. Brilliant and determined. Thank you for your companionship and advice. *Part of this thesis is yours.*

Mojtaba Kahrizi You stepped in where nobody else would. You have been patient and encouraging. Your decency as a man is clearly visible. You have served your role as a great supervisor and you are an excellent role model. Thank you for staying with me on the final months of this difficult journey. You have my implicit trust and gratitude. *Part of this thesis is yours.*

David Brownlee & Sonja Huntgeburth The balance of heart and mind is a serious matter and you have both taught me this. I have watched you conduct yourselves with great humanity and professionalism in the face of adversity and disorder. By example, you have both shown me that the best research is done in relative calm, and that one should strive to achieve this through balance of one's personal life with one's professional life. Thank you for being my council and for teaching me to recognize when I simply put too much work on my plate. *Part of this thesis is yours.*

Rensselaer Davis & Rula Kiblawi More than just work colleagues, you have become friends. My council when things went bad and my companions when things went good, I cannot ignore your influence in the early stages of this work. In the time we have worked together, you have conducted yourselves with friendly professionalism and sound engineering judgment. Thank you for leading by example, stressing that engineering quality is more important than technical accolades. *Part of this thesis is yours.*

Finally, I would also like to thank you, my siblings, friends, and past colleagues, for your patience and understanding. In your own unique ways, you have all inspired me to complete this journey with both pride and dignity. Thank you for honoring me with your implicit support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Anomaly Detection & Localization

An *anomaly* is the specific event that causes the violation of a process observer's expectation about that process. When a context has changed significantly, within a qualified scene, an intelligent system denotes this as an anomaly. While context and scene features can vary across observers, the detection mechanism for significant context change is essentially a salient feature of intelligent observers. Causal biological systems are essentially reactive systems that retain some remarkable predictive qualities due to their ability to qualify their environmental context in a terse, computationally efficient manner that allows for reliable predictive assertions to be made based on information constrained to time-frequency windows. The plasticity of this type of short-duration, predictive-reactive system is more apparent in the long-term observation of biological systems where inherent operational control laws are stable and remain stable where there are radical changes in scene context.

Sensor array geometries that localize point sources in the far-field require a signal phase estimate from time-limited samples at each of its point sensors. Since the spectral composition of an anomaly is generally unknown, frequency isolation can be especially important for phase estimation where the narrow band power of the anomaly is insignificant when compared to that of the wide band signal. Mitigating the fact that short-duration events cannot be limited in both time and frequency, compact, shift-able and scalable wavelet bases allows for accurate signal representation while offering analysis capability of scalable time-frequency windows.

This proposed method for anomaly detection and localization first attempts to statistically characterize Teager energy in filtered sub-bands. By distinguishing between extreme and outlier sample values that have appeared in the sub-bands of array sensor data. The outlier data in the time-frequency window can then be used to estimate array phase data required for computing wavefront direction of arrival in the far-field.

1.2 Thesis Overview

The robust anomaly detection and localization system proposed consists of two major subsystems that are interlinked: the robust anomaly detection subsystem, and the sub-band anomaly spatial localizer. The robust anomaly detection subsystem is primarily designed for detecting total energy deviation in signals that are wide-sense stationary, or short-time (trend) stationary. Sub-band localization in the far field allows for anomaly positioning, even in the presence of a more powerful wide-band sources.

1.2.1 Methodology

In this work, the problem of anomaly detection and localization has been reduced to a problem in robust statistics. An automated observer is designed to detect when the total energy in wavelet filtered sub-bands radically changes in an acoustic scene. With an assumption that the source consists of several contributing narrow-band sources, demodulation of each can give a measure of instantaneous total energy in that band if the further assumption is made that the constituent sources are modeled as a second order systems.

Typically Laplacian audio is rendered Gaussian with a transformation of random variable from which total energy is measured. Another transformation of the windowed-average of the total energy allows for its robust statistical characterization using the robust MCD estimator. A detection scheme is designed to detect when the modified sub-band energies violate the statistical expectation. The energy deviation metric is the Mahalanobis distance for which a confidence threshold can be computed. The violating energies imply that a fundamental change has occurred in their corresponding sub-band. Major deviations in energy trends across all sub-bands imply that the acoustic source has changed significantly in its energy content. The modified energy content of the source highlights the importance of the sub-band expectation violations and warrants localization. Using established localization methods, position estimates are obtained for these violations in each sub-band. Accounting for the possibility of a source with spatiotemporal properties, the median of sub-band position estimates provides the final spatial information about the source.

1.2.2 Contents

This work outlines the development of an acoustic anomaly detection and localization system as well as some of the theoretical concerns affecting its performance. Section 1.3 of this chapter is a general survey of literature that describe different methods and practices that influence either the detection of anomalies or the localization of sources, which both directly and indirectly influence this work. A critical review of previous work is essential to understanding the nature of the problem at hand and to justify a sound design path.

Described in Chapter 2 are the theoretical foundations that support this work.

Section 2.1 covers the general methodology concerning localization of an energy source in using sensor arrays in the far field by estimating phase differences between sensors configured in an array. Section 2.2 describes the Teager Energy operator's properties used for measuring total source energy and for signal demodulation. Sample outlier detection using robust Mahalanobis distances obtained from the robust scatter estimator from the Fast-MCD algorithm is discussed in in Section 2.3. With a theoretical foundation laid, Chapter 3 clearly states the problem of *anomaly detection and spatial localization* in Section 3.1.1 with a hypothesis for its solution in Section 3.1.2. The remaining sections describe theoretical contributions that support the thesis hypothesis concluding with a high-level discussion of the proposed system architecture. Chapter 5 finally concludes this work with an overview of the design and with a projection of future research that stems from this work.

1.3 Literature Review

1.3.1 Overview

The detection and localization of unspecified anomalies in array sensor stream data can be used in a wide variety of areas including weapon systems, mission-critical system fault monitoring, medical diagnosis, and intelligent robotic data acquisition. Typical methods that have been implemented, or appear in literature make use of environmental assumptions that may, or may not hold true in all real world conditions. Research and development in this area is typically guided by some notion of signal stationarity where the source's control law can be reasonably assumed, decomposed or estimated. Assuming statistical stationarity justifies the use of well established methods such as maximum-likelihood parameter estimation, neural networks, radial basis functions, and principal component analysis to characterize signals. While these methods work well to model normal data flow, in stationary (stochastic) processes, piecewise stationary and chaotic signals maybe difficult to model since iterative techniques will have difficulty to converge to a solution, and the statistical models assumed for parameter estimation may no longer be valid. Assumptions about noise also play a major role in how signal pre-processing is done. Implicitly, most of these methods assume an unrealistically high signal-to-noise ratio. More often than not, with sensor arrays in practical scenarios, this assumption may be somewhat stretched. The type of noise is also of concern, especially where more than one sensor is of concern. While some systems have good properties with white noise, colored noise remains a problem but can be addressed using specialized techniques. In order to design any system, some sort of assumption will have to be made at some point. With respect to anomaly detection, it better that few assumptions be made about a possible anomaly. This way, we attempt to design a system that will not be overly tuned as a detector for signals that have been constrained to a set defined by the assumptions. While the problem of anomaly detection in data streams can be difficult, it is not impossible to design a system that can be used for practical purposes.

Localization with sensor arrays requires assumptions to achieve reasonable performance with reasonable computational complexity. While it may not be the case, signal stationarity is typically assumed for the localization process. Through careful control of sampling, we can make this a more reasonable assumption for smaller signal samples (we assume stationarity for shorter signals). Furthermore, we make assumptions on the signal wave's geometry. Planar waves make for simplistic computation and are a good choice where the curvature of the wavefront is nearly flat. These assumptions combined allow for detection of phase differences in planar waves using such things as linear or planar rectangular sensor arrays. The cross-correlation method is widely used for phase difference estimation produced by an incident source across linear, uniformly spaced sensors. This method relies on the relationship between the Fourier transform of the source and its autocorrelation function. The

Wiener-Khinchin theorem establishes this relationship on the condition of wide-sense stationarity. Localization this way will depend on the coherency of the signals at each of the sensors given reasonable assumptions about the randomness of the source as well as the shape of the energy waves it produces.

In an incremental fashion, the literature review will build the topic of anomaly detection and localization from the work of previous authors. After assumptions about simple localization are discussed from the point of view of previous authors, some of their results are highlighted and compared between works. Previous works in anomaly detection will be compared with an emphasis on the constraints on the incoming signal. Finally, feature extraction in low SNR signals is also discussed, but only in the context of signal characterization.

1.3.2 Localization

The localization of a source in the vicinity of a sensor array can be done in various ways. With a varying numbers of sensors, the extraction of spatial information is dependent on the sensor geometry. Processing takes advantage of the fact that an energy wavefront emanating from a source will not pass through each of the sensors at exactly the same time because of the medium in which it travels. The simplest geometry is the two sensor array (uniform linear array - ULA) which in acoustics, is typically a *stereo-microphone*. To estimate the direction of incidence of the acoustic wavefront generated by a target in a stereo field, the following methods can be used [1]:

- sound intensity,
- time delay estimation using cross-power spectral phase
- time delay estimation using cross-correlation function analysis.

Using sound intensity for localization allows for compact microphone arrangement however discrimination and separation of sound reflections is impossible therefore making localization sensitive to reflection. Also, the microphone sensor arrangement requires that there be precise phase and amplitude matching between microphones. The time delay of arrival (TDOA) can be estimated from the cross-spectrum of the spatially separated microphones. The *phase* of the cross-spectrum contains the information regarding the delay and hence the direction of arrival of the planar wavefront. Spatially separated sensors will experience a time delay in their data corresponding to a phase shift in the frequency domain. Consider the signal $x(t)$ (whose Fourier transform is $X(\omega)$) as it arrives unimpaired at the secondary sensor:

$$x(t-\tau_d) \Leftrightarrow X(\omega)e^{j\omega\tau_d} \tag{1.1}$$

While this statement holds true for any signal, this will only hold true for spatially separated sensors on the assumption that the noise at each of the microphones is incoherent, which can occur only after an infinitely long averaging time. The author of [1] showed that there are short comings to this method. .

The same information offered in the cross-spectrum can also be obtained from its time domain counterpart, the cross-correlation function. The cross-correlation function for two spatially separated sensors a and b is obtained from the following expected value (1.2):

$$R_{ab}(\tau) = E[a(t)b(t+\tau)] \tag{1.2}$$

We can estimate the cross-correlation function for a time window of width T, centered at time t:

$$\widehat{R}_{ab}(t,\tau) = \frac{1}{T}\int_{t-\frac{T}{2}}^{t+\frac{T}{2}} a(u)b(u+\tau)du \tag{1.3}$$

This estimate can also be obtained from the well known relation:

$$R_{ab}(\tau) = \int S_{ab}(\omega)e^{j\omega\tau}d\omega \tag{1.4}$$

Where S_{ab} is the cross spectral power. This estimate of $R_{ab}(\tau) = R_{ss}(\tau - \tau_d)$, where the latter ($R_{ss}$)is the autocorrelation function of the source signal whose peak appears at the time shift τ_d in which we are interested. Regardless of the method used to find an estimate for $R_{ab}(\tau)$ the peak (maximum) will reveal τ_d allowing us to estimate the angle of incidence θ_i:

$$\cos\theta_i = \frac{c\tau_d}{d_{mic}} \tag{1.5}$$

Where c is the speed of sound and d_{mic} is the physical distance between the microphones. This method is the most widely used technique since it is robust to multi-path signals and multiple noise sources. Peak detection is sensitive to noise and averaging time, and that slight changes in the acoustic environment may quickly shift the peak. With the purpose of facilitating peak detection in a sometimes deceptive cross-correlation function, spectral *pre-whitening* can be done. While this alleviates the problem it can be addressed further with interpolation near the maximum. In theory, where the signal is white noise, we expect a delta function to appear as the maximum of the cross-correlation function. Alternatively, the author of [1] suggests use of the *Hilbert transform* of $R_{ab}(\tau)$ to detect the peak by *zero-crossing*, producing the most accurate results.

The experimental work done by [2] confirms these findings. The authors of this work used the entire sound waveform from a robotic platform with binaural (stereo) sound recordings to increase sensitivity of several time-domain localization methods. The TDOA method was tested. They elaborated on the assumptions they made in their work:

- sound waves propagate along a single path from the source to the microphone pair,

- the response is approximately the same for both microphones,
- aligned with each other, the microphones are relatively near to each other when compared to the distance from the array to the source.
- there are no obstructions between the microphones.

Amongst the methods reviewed by these authors, PHAT (Phase Transform), or cross-spectrum phase, was used as an alternative to simple peak-finding. This method assumes *non-stationarity* of the source utilizing a weighting function based on the short-time Fourier transform. This weighting function is used to enhance peak detection in the simple cross-correlation function. Intuitively, we know this to be true since the weighting function performs spectral whitening of the source as well as that of its phase shifted counterpart. The cross correlation of two noise sources will result in a delta function in the cross correlation function at the phase shift making peak-finding more reliable. A maximum likelihood method was also tested to determine phase shift using the Fisher discriminant. While this method is easy to implement in practice, performance degenerates if training data is non-stationary (ie. acoustic targets are moving). Finally, a *perceptron* was used to determine phase as a multi-class separation problem. Initially only linear classification rules were used, followed by the use of radial basis functions (kernels).

It was found by [2] that equalized cross-correlation functions were inferior to discriminative methods. There was no statistically significant difference between the use of Fisher's multi-class discriminant and the perceptron. Ultimately, PHAT proved to be most robust with a reasonable computational overhead. While maxima-finding is done in the time domain, spectral whitening is done in the frequency domain and consequently bearing the larger computational load.

Grassi and Shamma designed a learning, biologically inspired algorithm for localization [3]. In their work, they noted that the barn owl (a nocturnal predator) has exceptional localization abilities. This animal's highly developed localization pathway was used as a model for their work. These birds, as do humans and many other animals make use of ITD (*interaural time difference*) and ILD (*interaural level difference*) in order to localize sound. Their model used a *location estimate per channel* where a bank of simulated cochlear filters logarithmically spaced between 2 kHz and 11 kHz was used with a 70 ms processing window. Their model computes this estimate, the ABL (*average binaural level*) for each channel of spectrally decomposed sound. This spectral decomposition solves a problem of localization which is where SNR in certain frequency bands is very poor. The estimates are combined using a weighted average to produce an estimate of the direction of arrival for the sound. Interestingly, experimental data suggests that barn owls have the ability to locate on *both* the horizontal and vertical axis using ITD and ILD only.

1.3.3 Anomaly Detection

The problem of detecting anomalies in data streams has been examined in the past with some success despite the difficulties stemming from the non-specificity of the problem's parameters. A general survey of literature will highlight that typical anomaly detection strategies will make some or all of the following assumptions [4]:

- The background is static and/or uniform
- The data's control law does not change.
- The event of interest or its spectrum is known.

Raeth and Bertke [4] offer an approach for detecting *unspecified anomalies* in *unspecified data streams* that is spectrum independent. In their work, they attempted to find interesting and unexpected events in in continuous data streams using an automated process. They also attempted to detect *potential events* without having to specify beforehand the data source or its characteristics. They had developed an adaptive detection scheme that predicted the next sample in a data stream. Their prediction model was composed of a network of independent Gaussian radial basis functions such as $g_i(x, \xi_i)$ shown in (1.6):

$$g_i(x, \xi_i) = e^{\pi\sigma_i^2 \|x-\xi_i\|^2} \tag{1.6}$$

Where ξ_i is the location of the node, and σ_i^2 is its variance. In their scheme, they make use of the basis function in a function aproximator:

$$f(x) = \sum_{i=1}^{n} c_i g_i(x, \xi_i) \tag{1.7}$$

In operation, the amplitudes of the basis functions and hence the signal model approximation is continuously adjusted through c_i. As the model is being built, it gradually becomes able to predict the next sample in the continuous data stream's sequence. Models with a detected event stop evolving until the event is no longer present to prevent the event from becoming part of the model's background predictions. The latter will retain the model's sensitivity to future such events. This behavior is controlled by a set of heuristic rules that essentially measure the amount of signal departure from the adjusted function aproximator (1.7) according to some set threshold, and over a set number of samples.

Through their experimentation, it was clear that the methodology had good merit when tested on both images and sound data. In the case of sound experimentation, they choose to detect a voice event that was immersed in the noise generated from a box fan. This event was detected with some reliability. While detection of the event was done with no preconceived knowledge of the event's spectrum, the background's characteristic was not well discussed, nor what the adjustment rule was for the weight

c_i in (1.7). If an iterative method were used on a *signal departure* error metric then the basis function weights would have to converge in order for the background to be reasonably qualified. It follows that weight convergence would require at least some notion of background stationarity in the statistical sense. In their analysis of one-step-ahead prediction functions, Modha and Masry support this idea by showing that neural networks and Legendre polynomials are consistent estimators, even when there is a constraint on the number of samples used [5].

Theoretically, a signal cannot be limited in both time and frequency simultaneously. Short-time Fourier transforms offer an analysis method for fixed time-frequency windows [6]. While this is a good method where the window is well-known for a problem (*ie.* sample length and bandwidth known), anomalies can occur across varying time intervals, with varying bandwidths. In order to deal with this, wavelets can be used. Wavelets are functions that form an orthonormal basis similar to the sine and cosine functions in the well known Fourier basis with the important exception that they are *well localized in both the time and frequency domain.* Furthermore, they offer themselves as a time-frequency analysis tool (although constrained by the time-frequency uncertainty principle). Using wavelet analysis, the authors of [7] demonstrated that common quality disturbances in electronic power supplies are caused by short-circuits, harmonic distortions, notchings, voltage sags and swells as well as transients during power switching could not only be detected, but identified and localized in time over varying bandwidths using wavelet decomposition of the power signal. Their method made use of wavelet analysis primarily as a pre-processing method prior to feature extraction. Their method involved the computation of energy in the wavelet coefficients at varying levels of frequency decomposition of the pure sinusoid used in their simulation experiments. Disturbances were then simulated and wavelet energy characterization was done for each decomposition level. Finally, the pure signal's decomposition energy is compared to that of seven different "curve families" of power quality disturbances. They used a simple normalized distance measure(1.8):

$$dp(j)(\%) = \left[\frac{en\_dist(j) - en\_ref(j)}{en\_ref(7)}\right] \times 100 \tag{1.8}$$

Where they provide the following description:

$dp(j)(\%)$ "deviation between the energy distributions of the signal in study and its corresponding fundamental sinusoidal wave signal, at each wavelet transform level.

$en\_dist(j)$ energy distribution concentrated in each wavelet transform level of the signal in study.

$en\_ref(j)$ energy distribution concentrated in each wavelet transform level of the correspondent fundamental component of the signal in study.

$en\_ref(7)$ energy concentrated in at level 7 (which concentrates the highest energy) of the corresponding fundamental component of the signal in study."

The feature vector they used consisted of the above distance measure for ten decomposition levels. They found in simulation that this was a very good feature vector, being able to detect and classify known disturbance types without any iterative training (deviations are characterized once). Essentially, the system can be considered as an anomaly detector if we do not attempt to classify disturbances, but simply recognize that the decomposition energy of the the "normal" signal has changed.

While the previously mentioned work work examined a very constrained signal with a small set of classification targets, it was done with the assumption of a very high SNR. Seekings and Potter examined the classification problem of marine acoustic signals where there is generally a low signal-to-noise ratio (SNR). The authors specifically examined whale song which is often considered to consist of sequences of repeated stereotyped units [8]. The purpose of their work was to recognize and classify each unit that constituted the whale song. The song units could consist of long tonals, short pulses or frequency modulated signals, not clearly time or frequency localized. They noted that spectrogram matching methods (fixed time-frequency windows) are intolerant to time or frequency shifting, or stretching of any sort. Spectrograms also do not provide an intuitive way to extract feature vectors for characterization or classification. For these reasons, the authors, of [7] opted to use wavelet decomposition to overcome these issues while gaining some time-shift invariance and feature vector compression when used as a pre-processor for their neural network classifier.

While there are many choices for the orthonormal wavelet basis, the Daubechies Real Biorthogonal Most Selective (DRBMS) wavelet was chosen for their work. This wavelet has some attractive features, highlighted by the authors:

Time-Invariance Time-series shifting of the signal results in only wavelet packet shifts.

Fast Computation Fractal-like structure leads to fast wavelet transform techniques.

Sharp Transition Bands This minimizes edge effects of between frequency bands.

Furthermore, the authors attempted to reduce noise in the coefficients by thresholding using the Donoho-Johnstone estimator, optimized for this purpose.Their feature vector consisted of a Teager cepstrum for each wavelet packet decomposition that contains part of a whale call. Teager energy takes into account both kinetic and potential energy. This energy measure is considered to be a far more accurate measure as compared to the commonly used measure which takes into account only kinetic energy. The Teager energy cepstrum is often used to obtain feature vectors in noisy environments for speech recognition. It has been shown by [9] that the Teager energy gives a good measure of signal energy in a sub-band in the presence of *colored-noise*. For each frequency band, the energy is computed from the lowest level of the wavelet packet decomposition:

$$e_l = \frac{1}{N}\left|\sum_{n=1}^{N} \Omega_{n,l}(t)^2 - \Omega_{n,l}(t-1)\Omega_{n,l}(t+1)\right| \tag{1.9}$$

Where $\Omega_{j,i}$ corresponds to the subspace of level j, $l = 0, \ldots, 2^n - 1$ and $N = \frac{N_s}{2^n}$, n is the lowest level of decomposition, N_s is the length of the signal (therefore N is the number of samples in each sub-band. In their experiments, $N_s = 512$, $n = 6$ therefore $N = 16$. The Teager cepstrum is obtained from the discrete cosine transform of the log of Teager energy spectrum:

$$TC(k) = \sum_{l=0}^{2^n-1} \log(e_l)\cos\left(\frac{k(l-0.5)\pi}{2^n}\right) \tag{1.10}$$

for $k = 1, \ldots, 12$ since twelve points were used to encode the Teager energy spectrum (it was noted in their work that using more points did not affect classification results).

The resulting feature vector extraction was tested with two neural network classifiers, one simple back-propagation network (BP) and the other was a self-organizing map with learning vector quantization. Interestingly, they found little difference between the performance between these two types of network. This gives a strong hint as to the quality of the feature vector. Their results are summarized below:

Network	Training data correctly classified	Test data correctly classified
BP	89%	86%
SOM-LVQ	91%	86%

The success of this system shows that even with a relatively high SNR, time-shift and frequency-shift invariant systems can be designed for classification of signals.

1.3.4 Remarks

While the work surveyed concentrated mostly on anomaly detection and localization as separate concerns, this work is dedicated to using both of these theoretical influences for the the spatial localization of source anomalies. Where *anomaly detection* in this survey sought to signal significant signal departure from a historic baseline, this work recognizes that this baseline is subject to change over time and that anomalies are not just simply an abrupt change in signal characteristics but a sudden deviation of context in the acoustic scene as a whole across all sub-bands. With some notion of context change, it is the attempt of this work to localize a detected anomaly in a time-frequency window, and subsequently in space.

1.4 Conceptual Contributions

This work contains several published contributions [10] that are outlined here. They include signal characterization by energy content using the modulating source assumption, reshaping the energy distribution of Laplacian distributed audio to enable the detection energy outliers, use of a high-breakdown estimator to detect energy outliers using robust Mahalanobis distances, major and minor anomaly detection, anomaly localization as well as a lexicon of terms relevant to this area of research.

1.4.1 Modulating Source Assumption

Signal characterization by energy content for the purposes of anomaly detection requires a means for capturing both short and long duration energy changes. To detect an anomalous signal event, a baseline of what is considered normal is first required, from which energy deviation can be measured. Classical energy measurement does not measure instantaneous energy, making the detection of short-term energy deviations difficult, if not impossible. If we assume that all normal signals of interest have been modulated somehow, then demodulation would expose features of the signal that could help characterize it, such as amplitude in the presence of a constant frequency, or frequency in the presence of constant amplitude. To improve demodulation, complex sources are spectrally decomposed and demodulated in each sub-band. Termed the *modulating source assumption*, the Teager energy operator is used to provide AM, FM, and AM-FM demodulation in each sub-band. The demodulation property of the Teager energy operator provides an instantaneous measure of both potential energy from amplitude, and kinetic energy from frequency, or a combination of both. Because the total energy is measured in each decomposed sub-band, the total energy of the wide-band source is accounted for.

1.4.2 Reshaping the Teager Energy Distribution for Laplacian Distributed Audio

On the observation that the moving average of the Teager energy operator is log-Gaussian for a Gaussian input, Laplacian distributed sub-band audio data is transformed into Gaussian data. With the estimated mean and variance of the Laplacian audio, a non-linear function is designed using an inverse cumulative distribution method that will produce Gaussian distributed data with arbitrary parameters. Since the audio signals of interest are assumed to be changing constantly, fixed parameters for the target distribution cannot used. To make the target distribution dependent on the input distribution, the mean and variance for the Gaussian distribution are chosen to be the same as the estimated values of the Laplacian audio. In this way, the changing parameter estimates of Laplacian distributed audio can be used to specify a Gaussian redistribution, coupling them. The windowed Teager energy of this new signal is log-Gaussian distributed, and can also be redistributed into a signal that is Gaussian distributed using another non-linear function designed using the same method as the one already. Since the target distribution is Gaussian, a high-breakdown estimator can be used to characterize the signal's energy, even in the presence of outlier energy.

1.4.3 Use of a High-Breakdown Estimator to Detect Energy Outliers using Robust Mahalanobis Distances

Through successive random variable transformation, a Laplacian audio sub-band's total energy variable is rendered Gaussian, although unparameterized. The MCD is a highly robust mean and scatter estimator that provides reliable estimates with up to 25% of the data consisting of outliers (when the author's suggested default algorithm parameters are used). With parameters estimated for the modified Teger energy distribution, robust Mahalanobis distances can be computed for all energies. Given that the Mahalanobis distribution for a Gaussian variable are Chi distributed, a threshold can be established (given a confidence level) to determine when an energy does not belong to the distribution for which parameters were estimated. In this fashion, total energies that are too high or too low compared to the norm established by the majority of the sub-band's energy can be identified.

1.4.4 Major & Minor Anomaly Detection

With the modulating source assumption, complex acoustic sources containing narrow band anomalies can be characterized after spectral decomposition. This improves the chance of detecting a narrow band energy anomaly that may be hidden in a wide band signal and allows for baseline energy characterization in each spectrally decomposed band. With robust energy characterization in sub-bands, energy outliers are are detected and labeled as minor anomalies occurring in the signal sampling period. As a complex acoustic source changes over time, its total energy will change accordingly. The contributing sub-band energies will also change over time, demonstrating trends which are used to characterize the entire acoustic signal over the signal sample period. With the expectation that sub-band energy trends will not change radically over successive signal sampling periods, major sub-band trend deviations provide an indication that minor anomalies within the period containing the deviations and are of greater importance. Minor deviations detected during radical sub-band deviations are called major anomalies. While minor anomalies occur in a single signal sampling period, major anomalies occur over successive sampling periods providing both short and long term sensitivity to signal energy changes.

1.4.5 Anomaly Localization

This work provides a strategy for the localization of major anomalies. Once detected, an event is isolated within a time-frequency window. Since the energy of the anomaly was detected in a particular sub-band where it's signal to noise ratio is improved compared to what it would be in the wide band, the extraction of spatial information is done in only that band. For stereo acoustic localization, the time delay for a wavefront from a single distant point source to reach a second microphone after having reached a first reveals the direction of the source relative to the position of both microphones. The basic cross-correlation method is used to estimate this delay in each sub band using only the major anomalies (if present). Using acoustic wave propagation properties, the delay estimations are translated into azimuth estimates and the position of the anomaly is resolved from the median azimuth angle across the sub-bands containing anomalies.

1.4.6 Technical Lexicon

Research in anomaly detection continues and is gaining technical importance. With the goal of promoting discussion and research, a relevant technical proposed. This works makes use of the following terms and concepts which are discussed in this work:

anomaly

minor anomaly

major anomaly

stress

context

acoustic scene

attention span

anomaly localization

spatio-spectral sources

While used in the context of this work, the concepts presented here are meant for general use in the context of anomaly detection and localization.

Chapter 2

Theoretical Background

2.1 Sound Localization

The problem of *acoustic localization* is to determine the direction of arrival of a wavefront emanating from a an acoustic source relative to an acoustic sensor away. By taking advantage of the propagation delay of the wavefront in the air medium, phase differences in the signals from spatially separated acoustic sensors can be estimated and then translated into directional information. Because wavefronts will travel radially outwards from a point source, the distance from the sensor array to the source will affect the perceived *shape* of the wavefront. In the *near-field*, where the source is very close to be sensor array, the spherical wavefront's characteristic curvature is pronounced to the sensor array, manifesting itself in the phase shifts perceived by the sensors. In the far-field, the sensor array is sufficiently distant from the source that the wavefront shape appears to be almost planar to the sensor array. This section concerns itself with the localization of random acoustic point-sources in the *far-field* where location estimation is simplified because of the approximation.

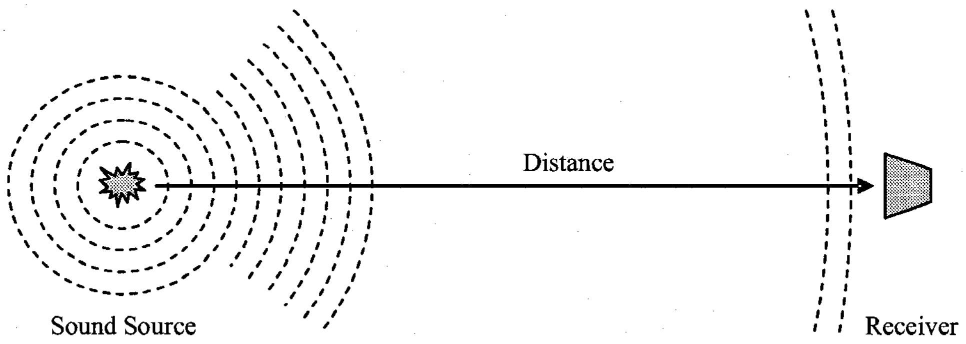


Figure 2.1: A spherical wave-front emanating from a point source appears planar to a receiver that is sufficiently far away.

2.1.1 The Far Field Assumption

The process of localization depends on the propagation delay imposed in the acoustic wave front by the medium in which it travels. The geometry of the sensor away also plays a critical role in this process. While a sophisticated phase model can be developed, this would only serve to complicate the phase estimation process. To avoid this, a simplifying assumption is made that will not have a severe impact on the phase estimates. Consider that in a relatively non-turbulent chamber, sound waves travel spherically outwards from the source as shown in Figure 2.1. For small distances, the pronounced curvature of the spherical wavefront complicates phase estimation, especially where the distance between the source and the center of the array approaches the average distance between the sensors in the array. While this complication of near-field operation is resolvable in a general solution, it is done at the expense of a more difficult analysis. If we assume that the sensor array is sufficiently far from the source then the wave-front geometry perceived by the sensor array will be approximately planar. Given that the ideal near-field solution is only valid in an unrealistic non-turbulent environment, the planar wave approximation is attractive. In this work it is assumed that the distance of the sensor array is sufficiently far from the source that a planar wave-front is perceived by the array. This is the *far-field* *assumption.*

2.1.2 Wavefront Propagation and Sampling

Stereo localization is a spatio-temporal problem whose solution parameters are resolved in both space and time. Array geometry therefore very important. For a simple stereo microphone pair, the radial distance between sensors is crucial for the determination of location from the inter-sensor wavefront propagation delay. This also plays a critical role in establishing the Nyquist frequency for this simple array. For discrete time-processing therefore, we can consider the relationship between sampling frequency and sensor spacing. The propagation delay τ (in seconds) of a wavefront from one stereo sensor to another separated by a distance d (in meters) is given by the following fundamental relation:

$$\tau = \frac{d}{c} \tag{2.1}$$

Where c is the wavefront velocity through a specific medium. For the *air* medium, this can be approximated by:

$$c = 20.5 \cdot \sqrt{273.15 + T} \tag{2.2}$$

Where T is the ambient air temperature in °C. The fundamental period τ_o of the stereo acoustic array is established as the propagation delay of a wavefront from the position of one sensor directly to the other:

$$\tau_o = \frac{d}{20.5 \cdot \sqrt{273.15 + T}} \tag{2.3}$$

To prevent aliasing, the sampling frequency would have to be at least twice the fundamental frequency $f_o = 1/\tau_o$ for this array. This is to say, the sampling period

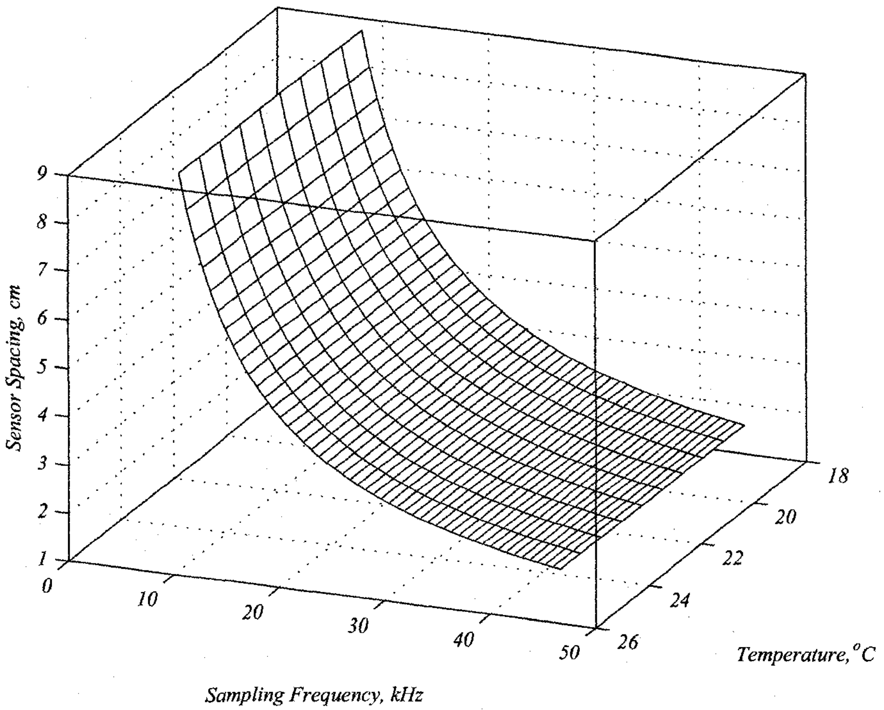


Figure 2.2: This plot shows that the effect of temperature is not as significant as the effect of the sampling frequency on inter-sensor distance.

τ_s must be less than half the fundamental period ($\tau_s < \frac{1}{2}\tau_o$):

$$\tau_s < \frac{20.5 \cdot \sqrt{273.15 + T}}{2d} \tag{2.4}$$

Figure 2.2 shows the negligible effect of temperature and the minimum distance for a chosen sampling frequency. If we assume an ambient temperature of 25°C and have a stereo acoustic array with an inter-sensor spacing of 1.6cm then we find that the sampling period is $\tau_s = 1/44100$ or $f_s = 44.1$kHz.

2.1.3 Sampling Frequency & Inter-Sensor Spacing

A point acoustic source located far from the stereo acoustic array will not necessarily yield perfect time shifted samples in the sensor data. Small turbulent vortices can occur between the sensors in even relatively calm air. This will introduce distortions

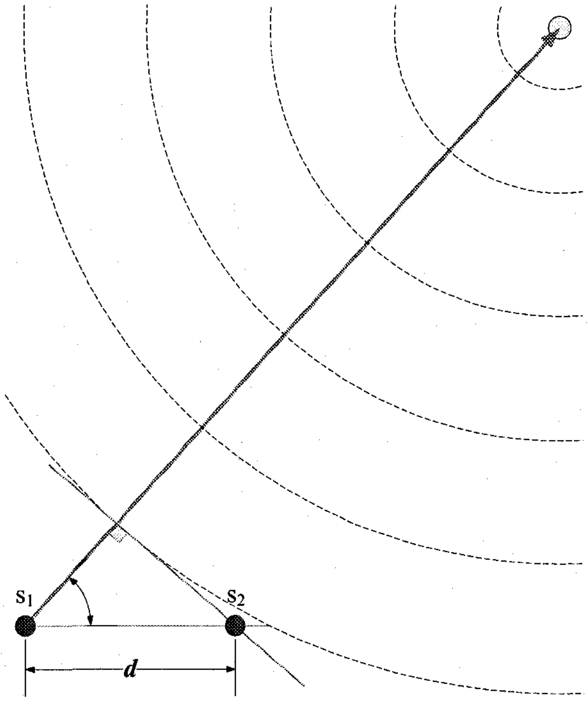


Figure 2.3: The far-field assumption assumes a planar wave emanating from the acoustic source. Here, the angle of incidence $\theta = \arccos \frac{\tau c}{d}$.

in the traveling wavefront that will be manifested as minor perturbations between between samples. Because of this sometimes chaotic behavior of the air between the sensors, we neglect this behavior knowing that the detectable differences will be minor, especially in the far-field. We assume that the air is perfectly still causing only a phase difference, or delay between sensor samples. Furthermore, we maintain the far-field assumption which implies that no correction will need to be made for the difference perceived by the spatially separated sensors. Depicted in Figure 2.3, this assumption of a planar wave allows us to trivially relate the wavefront's angle of incidence θ with the inter-sensor distance and the propagation delay of the planar wavefront across sensors s_1 and s_2:

$$\theta = \arccos \frac{\tau c}{d} \tag{2.5}$$

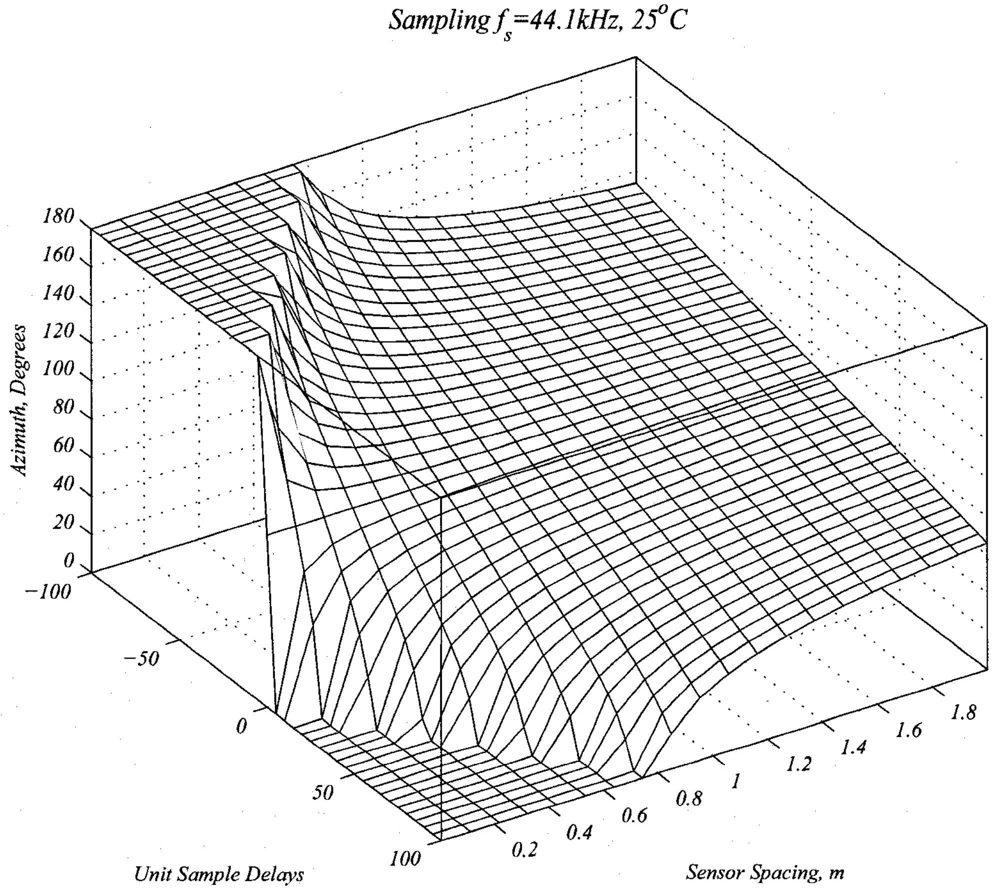


Figure 2.4: The surface shown represents θ for differing values of d and n. Clearly, the sensor spacing determines the number of samples required to cover $0° \leq \theta \leq 180°$. Consequently, spacing will determine the number of quantization levels.

Both d and c are taken as constant, allowing this simple relation to determine the azimuth of an acoustic source relative to a stereo acoustic array. Discrete sampling will quantize θ since τ is in units of a fixed τ_s;

$$\theta = \arccos \frac{n\tau_s c}{d} \tag{2.6}$$

Figure 2.4 shows the azimuth for various sample delays and inter-sensor distances. For smaller values of d, the values of θ span their full range although have a greater number of quantization levels. Similarly, we note that large d require a large number of samples for the same range coverage. For Figure 2.4, we can determine the optimal distance for a set quantization level reflected in the number of sample delays n. Consider the following example with $n = \pm 99$ and $f_s = 44.1\text{kHz}$ at 25°:

$$\begin{aligned}
\cos\theta &= \frac{n\tau_s c}{d}\Big|_{\cos\theta=\pm 1} \\
d &= \pm n\tau_s c \\
&= \pm 99\left(\frac{1}{44100}\right)(353.97) \\
&= 0.7946
\end{aligned}$$

Given these parameters, the optimal distance therefore is 0.7946m. Naturally, this can be repeated for other values of n offering a control over quantization levels in θ. Similarly, for a fixed distance and sampling frequency (and temperature is invariant), we can determine the number of samples that will be required in order to cover the full range of the azimuth.

2.1.4 Delay Estimation

Since no estimation model and no *a posteriori* probability density function exists for an uncharacterized acoustic source, standard estimation techniques such as maximum-likelihood and maximum a posteriori methods cannot be used to estimate the propagation delay τ. We can examine this estimation problem in both the time and frequency domains.

A signal that is both first and second-order stationary is said to be *wide-sense stationary.* For such signals, the Weiner-Kinchin theorem relates the poser spectral density (2.7) of a signal to its autocorrelation function (2.8) as a Fourier transform pair:

$$\begin{aligned}
S_X(\omega) &= \int_{-\infty}^{+\infty} R_{XX}(\tau)\exp^{-\jmath\omega\tau} d\tau \qquad (2.7) \\
R_{XX}(\tau) &= \frac{1}{2\pi}\int_{-\infty}^{+\infty} S_X(\omega)\exp^{\jmath\omega\tau} d\omega \qquad (2.8)
\end{aligned}$$

The autocorrelation function has a single maxima located at $\tau = 0$ and is perfectly symmetric about this point $(R_{XX}(\tau) = R_{XX}(-\tau))$. For time delayed signals, the maxima of the cross-correlation function is is shifted by τ_o, which corresponds to the signal delay time. For a two sensor array, the following is used to estimate the cross-correlation function:

$$\mathrm{C}(\tau) = \frac{1}{T} \int_{t-\frac{T}{2}}^{t+\frac{T}{2}} s_1(t) s_2(t+\tau) dt \qquad (2.9)$$

Where T is the estimation period, and $s_1(t)$ and $s_2(t)$ are the signals from each of the two sensors in the simple array. Estimation of the delay time involves finding the value of τ that maximizes $\mathrm{C}\tau$:

$$\tau_o = \max_{\tau} \mathrm{C}(\tau) \qquad (2.10)$$

Figure 2.5 shows an example of localization by peak-finding in $\mathrm{C}(\tau)$. Stereo acoustic sensors were spaced 10cm apart at 25°C. The discretely sampled signals yielded a peak that was shifted by 8 samples. At a sampling frequency of 44.1kHz, this corresponds to $\tau_o = 0.18141\mu s$ from a wave front with angle of incidence $\theta = 45°$.

Certainly, peak finding methods can be used to find τ_o however, some cross-correlation functions can be very deceptive for them yielding inaccurate estimates. The Weiner-Kinchin theorem suggests that signals with a relatively flat spectrum such as white noise will have an impulse-like autocorrelation function. Spectral whitening of sensor data will therefore improve the performance of peak-finding methods which consequently improves estimates for τ_o and therefore θ. Spectral whitening can be considered as an optimizing step and is not a requisite for coarse localization. This work, while it makes use of simplified localization from the cross-correlation alone, can be subjected to optimizations which are not the focus of this work. Spectral whitening is mentioned here only for the sake of completeness.

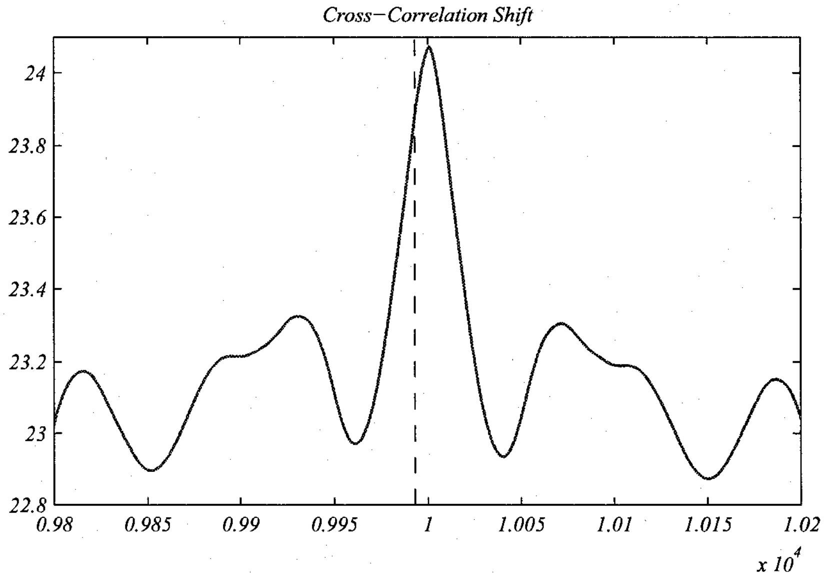


Figure 2.5: The unique peak of the cross-correlation function C(τ) is perfectly symmetric where the signals are identical. Delayed signals shift the peak from the center location at $\tau = 0$. In this case, an audio sample from a microphone pair with a spacing of 10cm and a sampling rate of 44.1kHz at 25°C results in an 8 sample shift of the maxima to τ_o. This corresponds to a 45° angle of incidence. The dotted line indicates the expected line of symmetry at $\tau = 0$, however the units on the abscissa correspond to the discrete sample index of the cross-correlation function.

2.2 Teager Energy

Under certain conditions, complicated random signals can be statistically characterized. Although very useful for very basic detection of signal amplitude deviation, this type of characterization gives very little information about signal content. Providing only a measure of scatter, central tendency and perhaps bandwidth, the signal's power and frequency are neglected. Measuring energy deviation is another way of detecting anomalies as it takes into account a *signal's strength* over a time period. By detecting when energies deviate from some established baseline, we can interpret this as an indication that the signal being monitored has undergone some fundamental change in its level of activity. Both of these measures correspond to measurements of *potential* and *kinetic* energy respectively in a signal. The Teager energy operator provides a means for measuring both of these quantities simultaneously for the source of certain types of systems. Sensitive to both amplitude and frequency, the Teager energy operator can be used to detect deviations in *total source energy.*

2.2.1 Signal Amplitude *vs.* Energy

Characterizing signals by their kinetic energy content takes into account the fact that the signal is a dynamic quantity and that amplitude variations that cancel each other out still need to be accounted for. Consider that the *instantaneous power* observed in a simple electric circuit is defined as either of the following time-domain functions:

$$p(t) = \frac{|v(t)|^2}{R} \tag{2.11}$$

$$p(t) = i(t)^2 R \tag{2.12}$$

Where $v(t)$ is voltage, $i(t)$ is current, and R is resistance of a trivial electrical system. Normalizing the resistance ($R = 1\Omega$), we observe that the power is simply the square of the input signal, regardless of whether voltage or current is being measured:

$$p(t) = |s(t)|^2 \tag{2.13}$$

The classic definition for *signal energy* and *total signal energy* over some time period are given by Equation 2.14, and 2.15 respectively:

$$\begin{aligned} E &= \int_{-T}^{T} |s(t)|^2 dt && (2.14) \\ E_T &= \lim_{T \to \infty} \int_{-T}^{T} |s(t)|^2 dt && (2.15) \end{aligned}$$

These definitions will certainly measure the activity in a signal, but it is unclear from them how signal frequencies affect the energy measures. In physics and engineering, *Parseval's theorem* is written as:

$$E_T = \int_{-\infty}^{\infty} |s(t)|^2 dt = \int_{-\infty}^{\infty} |S(f)|^2 df \tag{2.16}$$

Where $S(f)$ is the *Fourier* transform of the signal. We can interpret this as follows: *The total energy contained in the signal* $s(t)$ *across all time is equal to the total energy of the signal's Fourier transform* $S(f)$ *accrues all of its frequency components*[11].

2.2.2 Measuring Total Energy of a Source

By attempting to model the source system that generated $s(t)$ as a *spring-mass system*, we find that the energy function of that system as it *generates* a sinusoidal signal varies as a function of both amplitude *and* frequency which is quite different from what is stated in (2.16). It is this *source modeling* that is fundamental to the definition of the *Teager energy operator* and is used in this context for characterizing signals by amplitude and frequency. Consider the simple spring-mass source model

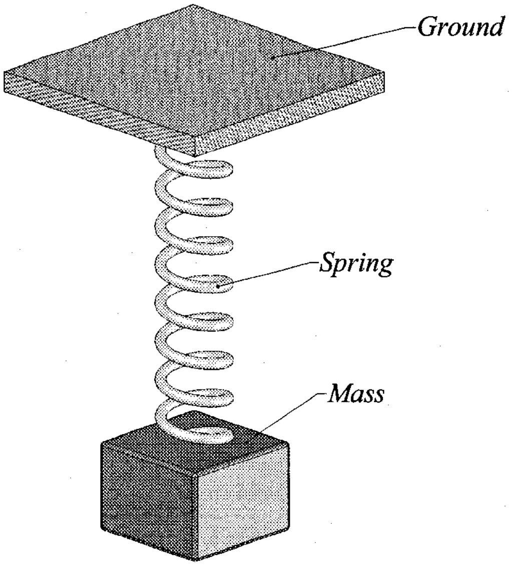


Figure 2.6: A physical spring-mass system as it corresponds to the mechanical circuit used for defining Teager Energy

shown in Figure 2.6, which is expressed in the Laplace domain as:

$$sCX(s) + \frac{X(s)}{sL} = 0 \tag{2.17}$$

$$s^2CX(s) + \frac{X(s)}{L} = 0 \tag{2.18}$$

Substituting C and L for their mechanical system counterparts ($C = m$ and $L = 1/k$), we obtain a second-order differential equation for position in the time-domain (x) which is the starting point for the Teager energy operator:

$$\frac{d^2x}{dt^2} + \frac{k}{m}x = 0 \tag{2.19}$$

Note that this simplistic model incompletely describes a mechanical-acoustical system. While accounting for mass oscillation which creates pressure waves its medium, the medium itself *is not* described by this model [11]. The periodic sinusoidal oscillation of the mass is observed from the solution to (2.19) which has the following form:

$$x(t) = A\cos(\omega t + \phi) \tag{2.20}$$

$x(t)$ is the position of the mass at time t, A is the amplitude of the oscillation, $\omega = \sqrt{k/m}$ is the frequency of the oscillation and ϕ is the initial phase. When $\phi \neq 0$, the system is not in initially in equilibrium.

Newtonian physics describes the total energy in the spring-mass system as the sum of both the spring's potential energy and the mass's kinetic energy:

$$E_T = \underbrace{\frac{1}{2}kx^2}_{\textit{Spring's Potential Energy}} + \underbrace{\frac{1}{2}mv^2}_{\textit{Mass' Kinetic Energy}} \tag{2.21}$$

Substituting the solution of (2.20) and velocity $v = \frac{dx}{dt}$ into (2.21), we obtain the following after simplification:

$$E_T = \frac{1}{2}m\omega^2 A^2 \tag{2.22}$$

The total energy of this system is clearly a function of both the amplitude of the oscillation (A) and the frequency of oscillation (ω).

2.2.3 Definitions of the Teager Energy Operator

Omitting the derivation from the spring-mass model, the definition of the *continuous* Teager energy operator followed by the *discrete* Teager energy operator are introduced [11]:

$$\Psi\left(\,(x(t)\,\right) = \dot{x}^2(t) - x(t)\ddot{x}(t) \tag{2.23}$$

$$\Psi\left[x_n\right] = x_n^2 - x_{n-1}x_{n+1} \tag{2.24}$$

While actually an estimate [11], the discrete version of the Teager energy (2.24) also has a more generalized definition [12] where a *lag parameter* M that is used to resolve closely spaced tones:

$$\Psi\left[x_n\right] = x_n^2 - x_{n-M}x_{n+M} \tag{2.25}$$

For a sinusoidal excitation $x(t) = A\cos(\omega t)$ we can clearly see again that the continuous Teager energy is a function of amplitude and frequency (after simplification):

$$\Psi\left(\,x(t)\,\right)\Big|_{x(t)=A\cos(\omega t)} = A^2\omega^2 \tag{2.26}$$

It is this sensitivity that makes this form of energy measure interesting for the detection of deviations in both amplitude and frequency.

2.2.4 Demodulation Properties of the Teager Energy Operator

By setting either the frequency or the amplitude constant, it is clear from (2.26) how this operator can be used for the demodulation of AM, FM, or AM-FM signals since both the continuous and discrete forms of the Teager energy operator are sensitive to a signal's amplitude and frequency. Consequently, for a fixed frequency, Teager energy is sensitive to amplitude and can be used to for demodulation of AM signals where the carrier frequency is constant. Substituting into the continuous form of the the operator (2.23) on the understanding that the discrete form retains the the same properties, we have the greatly simplified result in 2.28 [11]:

$$\begin{aligned} s_{AM}(t) &= a(t)\cos(\omega_c t) && (2.27) \\ \Psi\left(s_{AM}(t)\right) &= a^2(t)\omega_c^2 + \cos^2(\omega_c t)\Psi(a(t)) && (2.28) \end{aligned}$$

For a simple sinusoidal baseband $a(t) = A\cos(\omega t)$, Figure 2.7 shows both the AM signal and its corresponding *instantaneous* Teager energy. Clearly, the measured energy

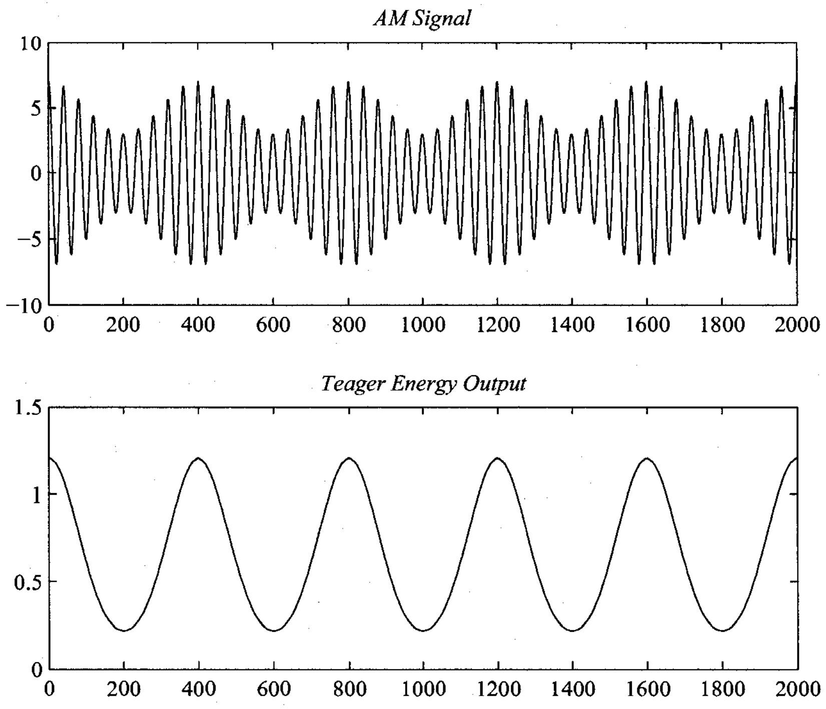


Figure 2.7: An AM signal with its Teager Energy Output

greatly resembles the envelope of the AM signal which is the modulated baseband ($a(t)$).

The Teager energy operator has FM signal demodulation properties as well (2.30) [11]:

$$\begin{aligned} s_{FM}(t) &= A\cos(\phi(t)t) && (2.29) \\ \Psi\left(s_{FM}(t)\right) &= A^2\left(\dot{\phi}^2(t) + \ddot{\phi}(t)\frac{\sin(2\phi(t))}{2}\right) && (2.30) \end{aligned}$$

Where $\phi(t)$ is the baseband of the FM signal. Figure 2.8 shows an example of FM demodulation where the baseband is again a simple sinusoid. We see again that the *instantaneous* energy measured has a great resemblance to the baseband signal.

Since the same operator can be used for either AM or FM demodulation, AM-FM demodulation is a reasonable prospect. The Teager energy for an AM-FM signal is

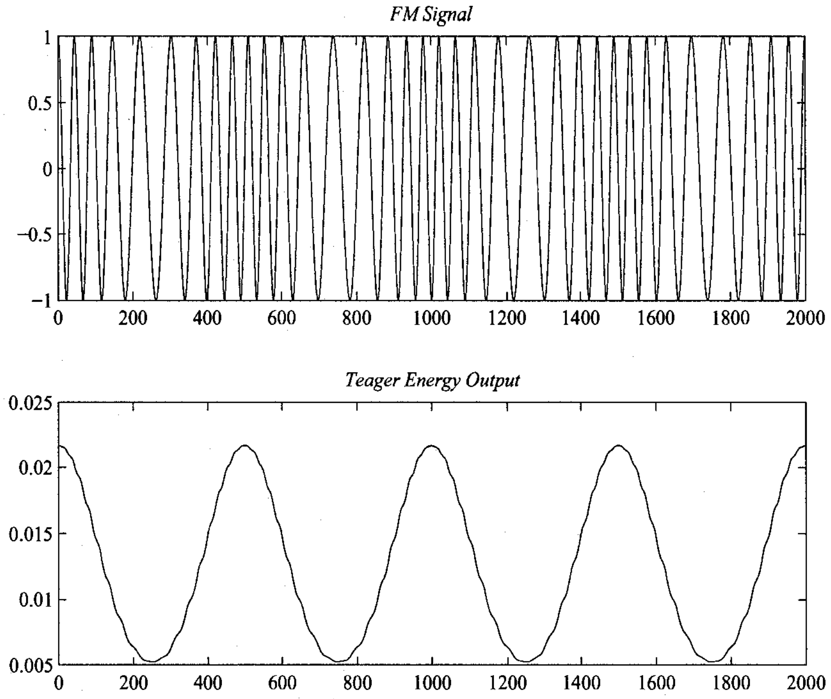


Figure 2.8: An FM signal with its Teager Energy Output

given by the following expression [11]:

$$\Psi\left(s_{AM-FM}(t)\right) = \left(a(t)\phi(t)\right)^2 + \underbrace{\frac{1}{2}a^2(t)\ddot{\phi}(t)\sin(2\phi(t))}_{FM} + \underbrace{\cos^2(\phi(t))\Psi(a(t))}_{AM} \qquad (2.31)$$

Figure 2.9 shows an example of an AM-FM signal with its *instantaneous* Teager energy. Upon close inspection of the modulated signal depicted, the Teager energy measure again greatly resembles the baseband signal.

While demodulation is not the goal of this work, demonstrating the demodulation properties of the Teager energy operator shows its sensitivity to both frequency and amplitude together. There are many signals that can be characterized using these parameters and by using the Teager energy operator to monitor the sources's *activity*, we are not restricting measurement to amplitude change alone.

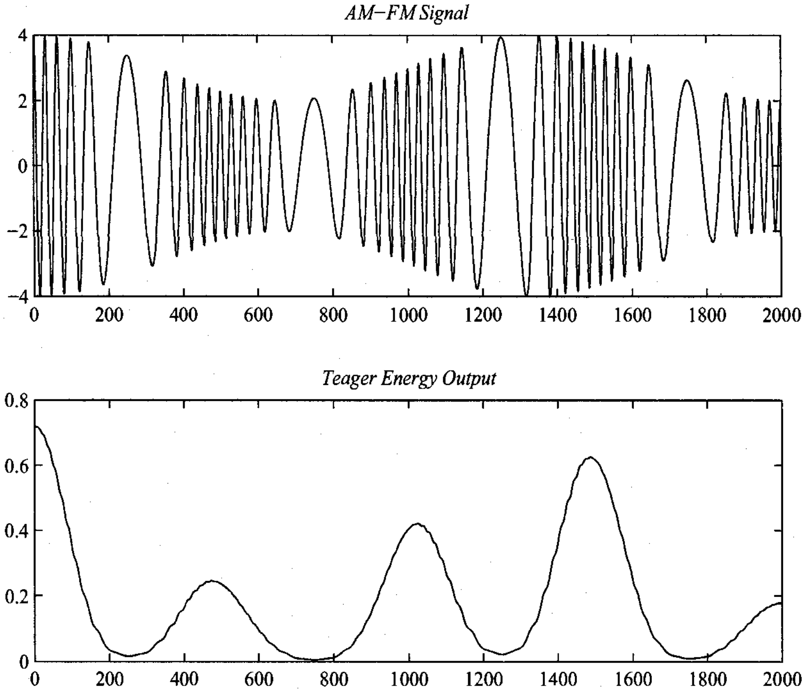


Figure 2.9: An AM-FM signal with its Teager Energy Output

2.2.5 Negative Teager Energy and Operator Noise Sensitivity

While Teager energy has interesting properties that are of use for characterizing signals, it is not without drawbacks. From the discrete operator definition in 2.24, we clearly see that its behavior is non-causal. While this can be overcome by acceptance of a one sample delay in the instantaneous energy computation (M samples for the generalized form in (2.25)), other properties may pose a problem for *signal characterization.* Notably, the problem of *negative Teager energy,* and *operator sensitivity to noise.*

Energy is a positive quantity, and a negative quantity measurement is strange indeed. For certain types of signals, *Teager energy yields negative energy* which is a strange behavior for *any* energy operator. Figure 2.10 depicts a signal containing two mixed sinusoids where the frequency of one is greater than the other, but with

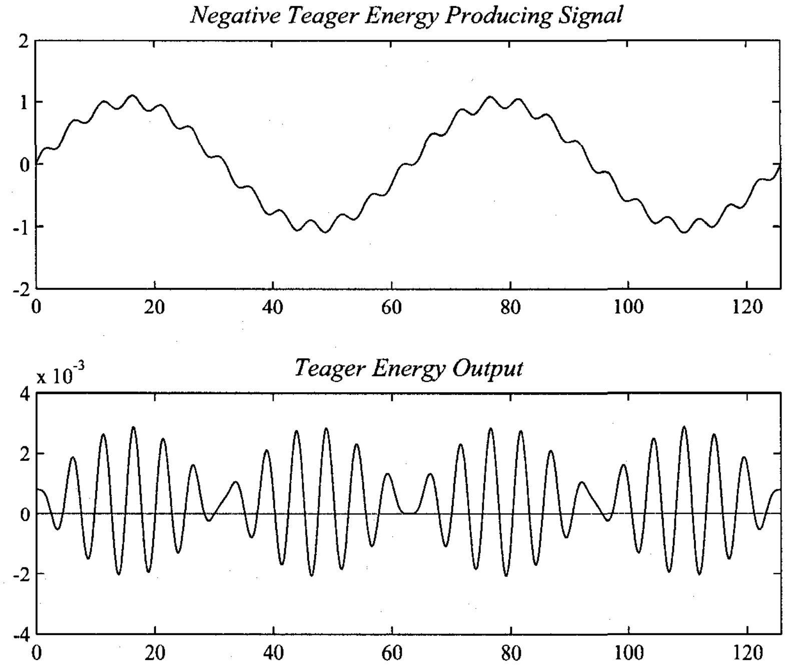


Figure 2.10: Certain signals yield negative Teager energy. This is one such signal. Notice how the signal resembles a noisy sinusoid although it is a deterministic function.

a much smaller amplitude (similar to a noisy sinusoid). In this signal, some of the energy measured will be negative. Although there are many other signals that will fall into this class, most real-valued signals do not. A detailed explanation of how to guarantee positivity of the energy measurement is not appropriate here (see [11]), but we should recall that the Teager energy model tries to model the energy of the *source* and not the signal although we speak colloquially to the contrary. The author of [11] suggests that if we consider the observed signal in Figure 2.10 was generated by *two* sources, each generating a sinusoid with one farther away and with higher frequency then Teager energy measurement will be based on an incorrect assumption of a single source system (2.18). This is a very reasonable explanation.

The second term of the discrete Teager energy definition (2.24) is essentially a *discrete differentiator* which (by definition) is very sensitive to abrupt variations. Noise can be viewed as rapid variation superimposed onto an otherwise smooth signal.

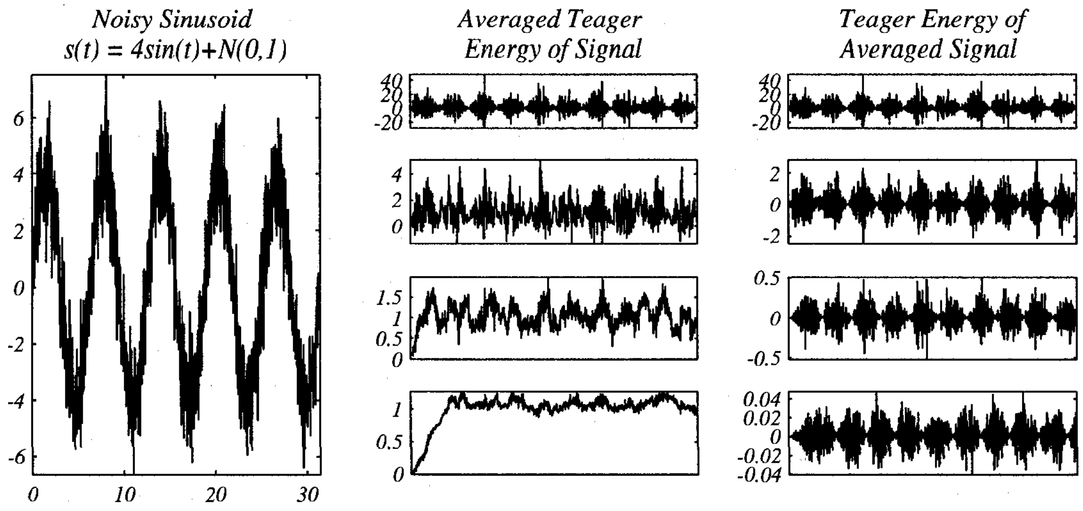


Figure 2.11: $s(t)$ is a simple monochromatic signal with additive zero-mean Gaussian noise of unity variance. Over 1500 samples, moving averages with window sizes $W \in \{1, 10, 50, 200\}$ are shown respectively in rows for $\overline{\Psi}(s(t))$ and $\Psi(\overline{s}(t))$.

Teager energy of a noisy signal will also be noisy and may yield negative energy. *Smoothing*, or *low-pass filtering* is one good solution to reduce operator noise, and more importantly to reduce the tendency of some noisy signals to produce negative energy. Figure 2.11 shows a monochromatic signal with additive noise on the left. The next column shows a windowed (moving) average of its instantaneous Teager energy ($\overline{\Psi}(s(t))$). As the window size increases, the operator tends to become more positive. The assumption of a single source for Teager energy does not account for the noise which appears as a secondary source and yields negative energy. As the window size increases, the effects of the negative energy are mitigated as the sinusoid becomes the more dominant component within the averaging window. Low-pass filtering of the Teager energy operator is one way of dealing with numerical differentiation noise. If the signal were pre-filtered in an attempt to remove noise using before Teager energy measurement ($\Psi(\overline{s}(t))$), we can see that residual noise will still produce negative energy that decreases with an increase in size of the averaging window. A windowed average of the Teager energy appears to produce an output that reflects the energy of the most dominant single source while pre-filtering of the source before Teager energy measurement appears to highlight the imperfections in the signal filter. Both averaging methods have the same goal which is to emphasize the more dominant *single* energy source in the presence of a secondary, low-amplitude, high-frequency contaminant source. Pre-filtering or post-filtering in this sense becomes an implementation detail that affects performance of the resulting Teager energy measurement of a noisy signal.

2.3 Outlier Identification

In order to quantify inherent behaviors of an experimental process, constraints are put into place so that sampled data can be analyzed according to a known model that reflects a behavior of interest. Distribution parameter estimation and model fitting methods can be very accurate, however anomalous samples may inadvertently appear that do not come from the process of interest. Generally from a completely different distribution model, these *outlier samples* cause the model fitting process to yield large residual errors and statistical parameter estimation to produce very poor confidence intervals. The goal of robust statistics is to account for outlier samples and produce good model parameter estimates for the majority of the sample data. This section describes Gaussian parameter estimation and more importantly, robust dissimilarity measures and decision criteria used for outlier identification.

2.3.1 Maximum-Likelihood Estimation

The non-robust estimation of distribution parameters for a sample set will require the minimization of a cost function or maximization of some goal function. Given a parametrized distribution model, and a sample set, *maximum-likelihood estimation* seeks to maximize a likelihood function to estimate distribution parameters. The likelihood function describes the probability that the entire sample set belong to a distribution with a given set of parameters. A general discussion of this method follows as it serves to highlight issues that are critical for robust outlier identification.

Given independent, identically distributed samples $x_1, x_2 \ldots x_N$, we wish to infer the parameters $\theta_1, \theta_2 \ldots \theta_k$ for a given distribution$f(\cdot)$:

$$f(x_1, x_2 \ldots x_N | \theta_1, \theta_2 \ldots \theta_k) \tag{2.32}$$

Because the samples are independent and from the same distribution, we compute the *likelihood function* $L(\cdot)$ as well as the *log-likelihood function* $\Lambda(\cdot)$:

$$L(x_1, x_2 \ldots x_N | \theta_1, \theta_2 \ldots \theta_k) = \prod_{i=1}^{N} f(x_1, x_2 \ldots x_N | \theta_1, \theta_2 \ldots \theta_k) \tag{2.33}$$

$$\Lambda(x_1, x_2 \ldots x_N | \theta_1, \theta_2 \ldots \theta_k) = \sum_{i=1}^{N} \ln f(x_1, x_2 \ldots x_N | \theta_1, \theta_2 \ldots \theta_k) \tag{2.34}$$

Because of the monotonic properties of the logarithmic function, we may estimate the parameters of interest by maximization. The estimate is for a given parameter θ_i is obtained by:

$$\frac{\partial \Lambda(x_1, x_2 \ldots x_N | \theta_1, \theta_2 \ldots \theta_k)}{\partial \theta_i} = 0 \tag{2.35}$$

Let us consider the estimation process for a univariate Gaussian function, whose estimators are very familiar. We wish to estimate the mean μ and the standard deviation σ for a Gaussian model from independent, identically distributed samples $x_1, x_2 \ldots x_N$:

$$f(x_1, x_2 \ldots x_N | \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}} \tag{2.36}$$

As explained earlier, the likelihood, and log-likelihood function are evaluated as:

$$L(x_1, x_2 \ldots x_N | \mu, \sigma) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}} \tag{2.37}$$

$$\Lambda(x_1, x_2 \ldots x_N | \mu, \sigma) = -\frac{N}{2} \ln 2\pi - N \ln \sigma - \frac{1}{2} \sum_{i=1}^{N} \left(\frac{x_i - \mu}{\sigma} \right)^2 \tag{2.38}$$

We maximize the monotonic $\Lambda(\cdot)$ with respect to μ in order to obtain our estimator which we now call $\hat{\mu}$:

$$\begin{aligned} \frac{\partial \Lambda(x_1, x_2 \ldots x_N | \mu, \sigma)}{\partial \mu} &= 0 \\ \hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} x_i & \end{aligned} \tag{2.39}$$

This *unbiased estimator* is the familiar sample mean. Similarly, we compute an estimator for the variance where the true mean is known *a priori*:

$$\begin{aligned} \frac{\partial \Lambda(x_1, x_2 \ldots x_N | \mu, \sigma)}{\partial \sigma} &= 0 \\ \hat{\sigma}^2 = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2 & \end{aligned} \tag{2.40}$$

The *asymptotically unbiased estimator* of variance is used when the mean is not known:

$$\hat{\sigma}^2 = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \mu)^2 \tag{2.41}$$

This estimator yields the true value of the variance where the number of samples N is large. Normally we expect parameter estimates and true parameter values to be the same where all of the samples are from a single parametrized, but unknown distribution. For all of its usefulness, the maximum-likelihood method takes into account *all* samples to perform its estimates including some samples that may have appeared from another distribution. This lack of discrimination is problematic for the estimators and in the case where the outliers carry sufficient statistical leverage, it can render them highly inaccurate.

2.3.2 Estimator Bias

While the degree of bias in these estimators is not directly of concern in our discussion of outlier leverage, the metric used to compute bias is. Consider the expected values for each of these estimators:

$$\mathrm{E}\left\{\hat{\mu} - \mu\right\} \tag{2.42}$$

$$\mathrm{E}\left\{\hat{\sigma}^2 - \sigma^2\right\} \tag{2.43}$$

Where μ and σ are the true parameter values, while $\hat{\mu}$ and $\hat{\sigma}^2$ are the parameter estimates. The *breakdown point* of an estimator is the maximal amount of model misspecification they can stand before their bias becomes arbitrarily large. The breakdown point for the p-variate maximum-likelihood estimators has been shown to be at most [13]:

$$\frac{n}{p+1} \tag{2.44}$$

Therefore, 900 samples drawn from a contaminated process that produces bivariate Gaussian data can contain at most $\frac{900}{2+1} = 300$ outliers before the maximum-likelihood estimators become unusable [14], with (2.42) and (2.43) becoming arbitrarily large instead of approaching zero for large sample sets. Maximum-likelihood estimators is the maximal amount of model misspecification they can stand before their bias becomes arbitrarily large and is at most $\frac{1}{p+1}$ for p-variate data [13]. For example, samples drawn from a contaminated process that produces bivariate Gaussian data can contain at most $\frac{n}{p+1}$ outliers before the maximum-likelihood estimators become unusable [14], with (2.42) and (2.43) becoming arbitrarily large instead of approaching zero for large sample sets.

2.3.3 Statistical Leverage in Maximum-Likelihood Estimates

As previously stated, maximum-likelihood estimation of density function parameters, while useful, makes use of all available data. Problems arise when some of the samples are taken from a different distribution. During the observation of a controlled process, presence of these *outlier* samples are typically due to the existence of a process that was unforeseen. Because they lie well outside of the range of other samples, outliers can introduce large errors in the parameter estimates for the assumed distribution model, altering the characterization of the sample set completely. Also, called *leverage points*, outliers can have a dominating effect on the distribution model estimates an should therefore be identified for removal before estimation techniques can be employed.

Considering the mean as a measure of central tendency of a sample, any values that are significantly far away from the majority of samples will have a very strong effect on $\hat{\mu}$. This can be clearly seen from a simple arithmetic example:

$$\mathrm{S} = \{0.8\ ,\ 0.5\ ,\ 0.7\ ,\ 0.9\ ,\ 0.3\ ,\ 0.6\ ,\ 0.7\ ,\ \mathbf{1000}\}$$

$$\mu = \tfrac{1004.5}{8} = 125.5625$$

Clearly, the last value in the sample set S is the only element that extends the range of the data set making it an outlier. If the other values were slightly modified without significantly extending or reducing the range of this subset, then we would see little change in the mean μ. Changing the single outiler value would have a significant effect on μ making it most sensitive to the outlier which explains why it is also called a *leverage point*. Removing the outlier reveals the true mean, the one that characterizes the majority of the samples:

$$\check{\mathrm{S}} = \{0.8\ ,\ 0.5\ ,\ 0.7\ ,\ 0.9\ ,\ 0.3\ ,\ 0.6\ ,\ 0.7\}$$

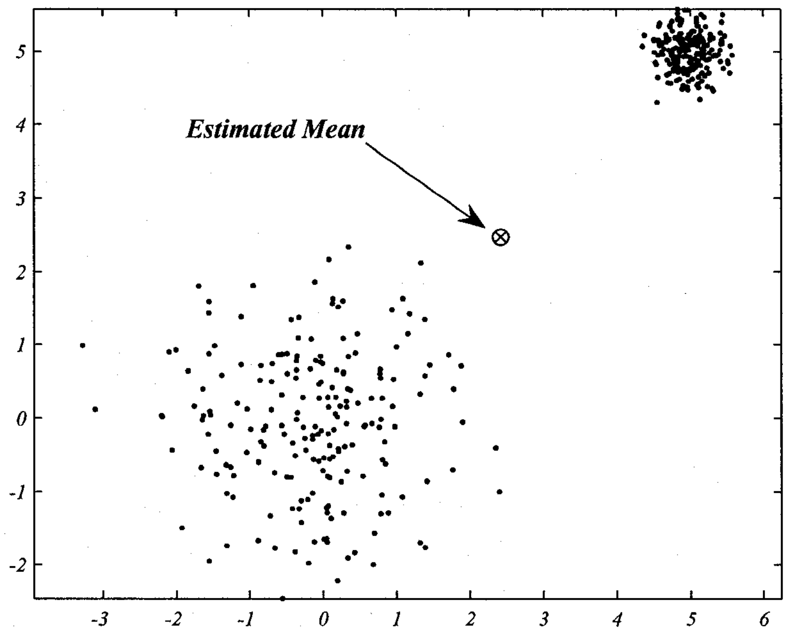


Figure 2.12: The maximum-likelihood mean estimation for a Gaussian mixture does not yield a descriptive result for this particular sample set.

$$\breve{\mu} = \frac{4.5}{7} = 0.64\ldots$$

Therefore, if we wish to characterize a sample set that may be contaminated with outliers, we should seek to identify these outliers so that they can be removed. Once removed and we have some confidence that the data is from a single distribution, we can perform the necessary parameter estimation to characterize the data. Figure 2.12 also demonstrates this point. The estimation model assumes that samples were taken from a single Gaussian Distribution. Using the derived estimators, on this Gaussian mixture yields an ambiguous result that does little to describe the data set adequately. Only outlier identification and removal will improve the estimates. Robust estimation will therefore require some sort of decision criteria that distinguishes between outliers and extreme values given some measure of confidence. Given a parametrized Gaussian distribution, and a new sample, the *Mahalanobis distance* provides a metric for the degree of membership of the sample to the known distribution.

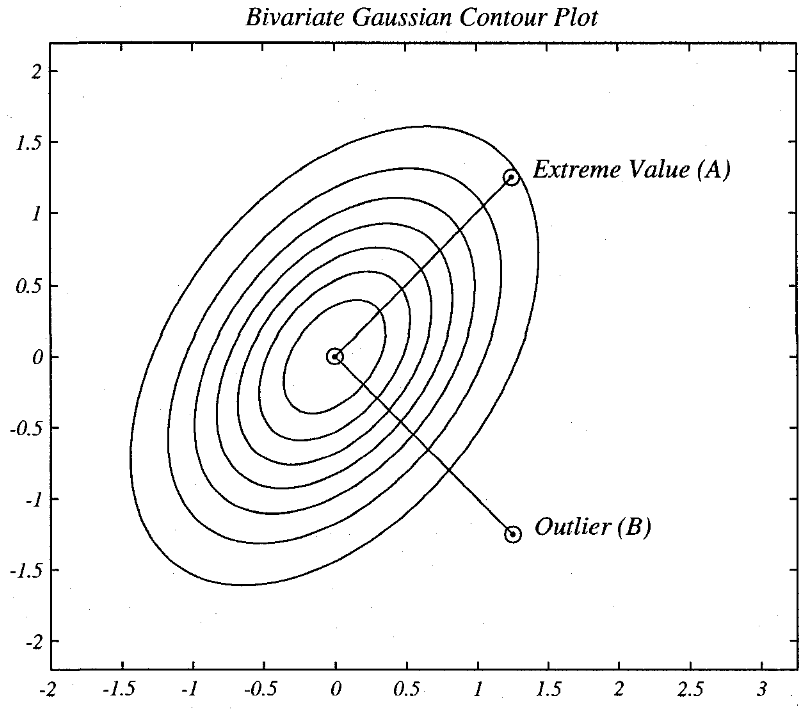


Figure 2.13: The Euclidean distance from the mean to both A and B is identical while the Mahalanobis distance is not. The suspected outiler A has a much higher Mahalanobis distance than the other extreme value B.

2.3.4 Mahalanobis Distance

Distance measurement is an intuitive way to measure the closeness of items. Based on correlations between variables, the *Mahalanobis distance* is a useful way of measuring similarity between a known and unknown sample set and unlike Euclidean distance, it is scale-invariant. Figure 2.13 shows the isolines of a bivariate Gaussian distribution with sample points A, and B. While they both have an identical Euclidean, distance from the mean, their Mahalanobis distance greater for B than it is for A indicating a greater dissimilarity between the sample and the distribution it was assumed to come from.

Given two sample sets $\vec{x}$ and $\vec{y}$ from the *same distribution*, with their covariance matrix Σ we define their dissimilarity measure with the Mahalanobis distance which is defined as:

$$\mathrm{d}(\vec{x},\vec{y}) = \sqrt{(\vec{x}-\vec{y})^{\mathrm{T}}\Sigma^{-1}(\vec{x}-\vec{y})} \tag{2.45}$$

In the case where Σ is a diagonal matrix, we have what is sometimes called the *normalized Euclidean distance*:

$$\mathrm{d}(\vec{x},\vec{y}) = \sqrt{\sum_{i=1}^{N} \frac{(\vec{x}_i - \vec{y}_i)^2}{\sigma_i^2}} \tag{2.46}$$

Where σ_i is the standard deviation over the x_i in the sample set. While (2.45) is the normal, more general definition that is used, from (2.46) we can see clearly that the Mahalanobis distance between two vectors is the length of the difference between them scaled in each dimension by the standard deviation. It is this scaling that normalizes the distance in each dimension and consequently in the overall distance measure. For a parametrized multivariate Gaussian describing a sample set with mean $\vec{\mu} = (\mu_1, \mu_2 \ldots \mu_N)^{\mathrm{T}}$ and covariance Σ, we can measure the dissimilarity between the sample set and an arbitrary sample:

$$\mathrm{D_M}(\vec{x}) = \sqrt{(\vec{x} - \vec{\mu})^{\mathrm{T}}\Sigma^{-1}(\vec{x} - \vec{\mu})} \tag{2.47}$$

While Figure 2.13 intuitively shows that $\mathrm{D_M}(\vec{A}) < \mathrm{D_M}(\vec{B})$, it does not indicate which of the two, if any, is an outlier. Because of the greater Mahalanobis distance, we suspect that B is an outlier. This may in fact not be the case. Recall that some probability density functions, such as this multivariate Gaussian are asymptotic. This means that the distributions have an infinite support region therefore values extremely far from the mean can theoretically appear, however unlikely.

Given good estimates for $\vec{\mu}$ and Σ, the squared Mahalanobis distance $\mathrm{D_M}^2(\vec{x})$ is a scalar value that is used to determine if $\vec{x}$ is part of an outlier set:

$$\textit{Outlier Set} = \left\{\vec{x} \in \Re^p \mid (\vec{x} - \vec{\mu})^{\mathrm{T}}\Sigma^{-1}(\vec{x} - \vec{\mu}) > T\right\} \tag{2.48}$$

The decision criteria and threshold T for determining whether a sample is an *outlier* or an *extreme value* with some specified confidence level is discussed in Sections 2.3.5 and 2.3.7.

2.3.5 Mahalanobis Distance Sensitivity to Covariance

The general Mahalanobis distance in (2.47) clearly shows the Euclidean distance between an arbitrary sample $\vec{x}$ and a Gaussian distribution mean $\vec{\mu}$ is scaled by the distribution covariance matrix Σ. This can pose a problem. In order to use this dissimilarity measure to detect outliers, these quantities must be known *a priori*. Given an unknown sample set, if we attempt to estimate these parameters and outliers are present, then the Mahalanobis distances will not be relative to the true distribution of the sample majority because of the outlier leverage on the estimates. Naturally, poor estimates will result in poor scaling making dissimilarity measures using the Mahalanobis distance in the presence of outliers unreliable (a masking effect that gets worse with the number of outliers). To circumvent this situation, an outlier-robust estimation scheme is required for the implicit parameters of the definition in (2.47). Clearly, too many outliers can be dangerous or classical statistical methods.

2.3.6 Robust Parameter Estimation

The minimum covariance determinant (MCD) algorithm is a highly robust mean and scatter estimator. The objective of this algorithm is to find a subset of observations whose covariance matrix has the lowest determinant. Hampered by its computational speed, the Fast-MCD algorithm [15] offers a great improvement in speed. For p-variate data, its objective is to find a set of h out of n observations whose covariance matrix has the lowest determinant. The tolerance ellipsoid with the smallest volume that covers h samples where $\frac{n}{2} \leq h < n$. It's breakdown is $\frac{n-h}{n}$.

The method considers a subsets of size $p + 1$ within the h observations to find the

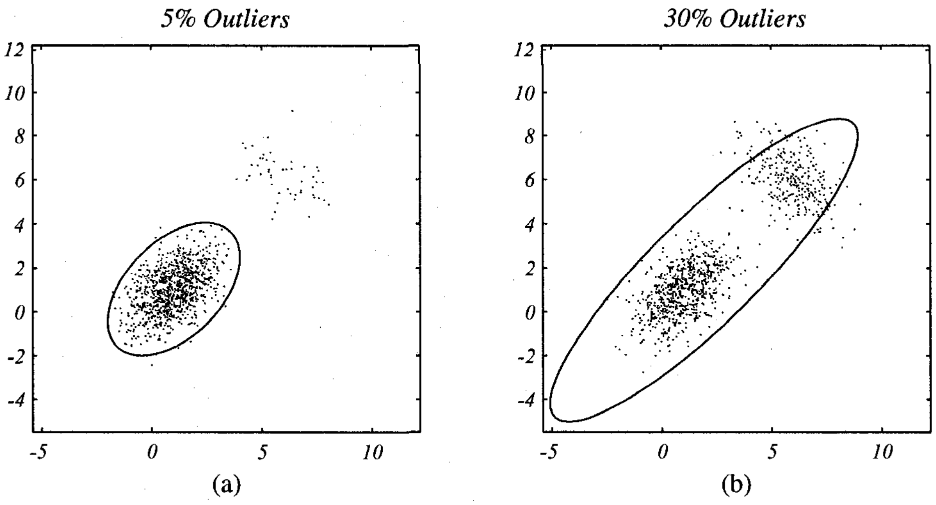


Figure 2.14: With a breakdown value of 25%, the Fast-MCD produces a robust estimate in (a) and a poor estimate in (b).

determinant with the lowest covariance. The value of h is chosen as:

$$\frac{n+p+1}{2} \leq h \leq n \tag{2.49}$$

with $h = \frac{n+p+1}{2}$ by default. Where we expect that less than 25% of the sample are outliers, we may consider $h = 0.75n$ as a compromise between the breakdown value and and statistical efficiency [15]:

$$\left.\frac{n-h}{n}\right|_{h=0.75n} = 0.25 \tag{2.50}$$

Figure 2.14 shows two sample sets with $n = 1000$ samples drawn from the same bivariate Gaussian distribution but with differing proportions of outliers which were obtained from *another* distribution. With the compromise value chosen for h, we expect the Fast-MCD estimates to be robust where only 5% of the samples are outliers. This is confirmed by the 97.5% tolerance ellipse shown in (a). The tolerance ellipse in (b) confirms the breakdown we expect since 30% of the samples are outliers and this proportion exceeds the breakdown proportion in (2.50).

Because the Fast-MCD algorithm provides a robust estimation for location and scatter ($\vec{\mu}$ and Σ), *robust distances* can be computed as well. With little or no outlier leverage, the sensitivity of the Mahalanobis distance is mitigated and the dissimilarity measure gains a more intuitive meaning where outliers are present. Since the estimated parameters represent a majority of the samples, the robust distances reflect dissimilarities from this norm. It should be clearly stated that the parameters are being estimated for the asymptotic Gaussian distribution therefore we cannot distinguish between *outliers* and *extreme values* from the same distribution without first determining the value of T from the decision criteria in (2.48). A large robust distance certainly indicates dissimilarity with the majority of other samples, but values far from the mean can most certainly arise regardless of how low this probability is.

2.3.7 Outlier Detection in Gaussian Distributions

Section 2.3.4 described the Mahalanobis distance as a dissimilarity metric. Given a known parametrized distribution, a degree of membership or distance can be established for a new sample. The distance measure in (2.47) has the following property:

$$\left\{\forall \vec{x} \in \Re^p \ , \ \mathrm{D_M}(\vec{x}) \in [0, \infty^+) \mid \vec{\mu}, \Sigma\right\} \tag{2.51}$$

A trivial observation, *any* vector in $\Re^p$ has a distance that can be measured to a specified a p-variate distribution. In this sense, no distinction can be made between outliers and extreme values for samples that are far from the distribution without some confidence. Those samples whose distance is within a specified confidence interval, are considered to belong to the distribution. Those that are close to the interval boundary but are still within it are considered extreme values although this is a subjective distinction. Those samples that are outside the interval are outliers and

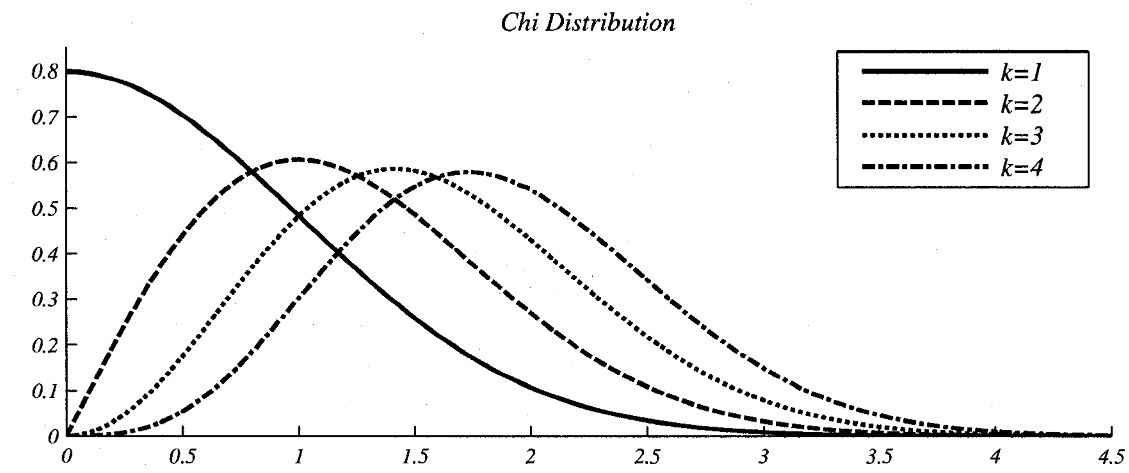


Figure 2.15: The Chi Distribution ($\chi_{\mathrm{k}}(x)$) shown with selected degrees of freedom $k = 1, 2, 3, 4$.

are considered not to belong to the distribution. A confidence interval is therefore crucial to make this distinction and should be reflected in the threshold chosen for T in (2.48).

Given vectors $\vec{x} = [x_1, x_2, \ldots, x_k]^{\mathrm{T}}$ and $\vec{y} = [y_1, y_2, \ldots, y_k]^{\mathrm{T}}$, consider the following statistic where all vector components are zero-mean gaussian distributed with variance 1 ($x_i \sim \mathrm{N}(0, 1)$ and $y_i \sim \mathrm{N}(0, 1)$):

$$Z = \sqrt{\sum_{i=1}^{k} \frac{(x_i - y_i)^2}{\sigma_i}} \qquad (2.52)$$

This statistic is *Chi distributed* with k degrees of freedom ($\chi_{\mathrm{k}}(\mathrm{x})$)as shown in Figure 2.15: Figure 2.16 shows the *Chi-square distributed* which is obtained from the statistic Z^2, also with k degrees of freedom. The Mahalanobis distance in (2.46) carries the same form as the Z statistic in (2.52). We can therefore conclude that for vectors whose components are independently distributed, $\mathrm{D_M}$ is Chi distributed, and $\mathrm{D_M}^2(\cdot)$ is Chi-square distributed. With this constraint observed, a confidence interval can be used to determine a threshold for the distances. For example, for a given sample set in $\Re^3$ in which we have confidence that 70% of the observations belong to a specified distribution $\mathrm{N_k}(\vec{\mu}, \Sigma)$. We can compute a *cut-off value* for $\mathrm{D_M}^2$ as shown in Figure

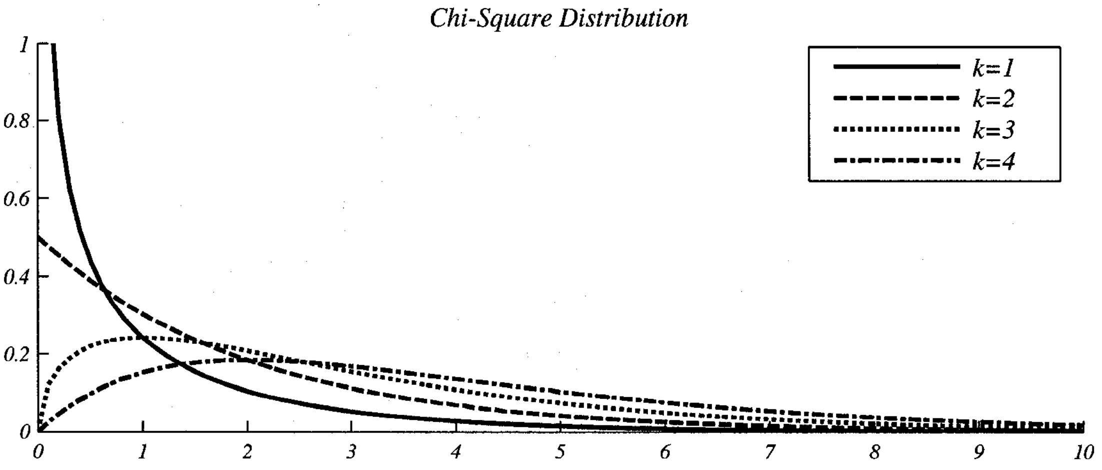


Figure 2.16: The Chi-Squared Distribution ($\chi_k^2(x)$) shown with selected degrees of freedom $k = 1, 2, 3, 4$.

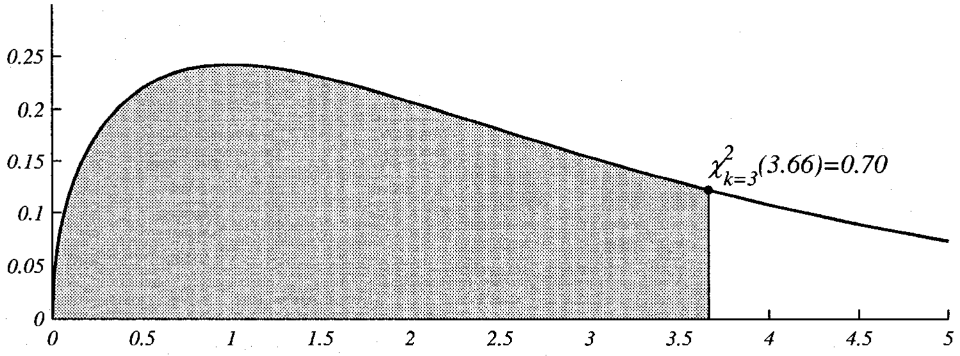


Figure 2.17: Given a 70% confidence that a sample from $\Re^3$ belongs to $N_k(\vec{\mu}, \Sigma)$, the inverse of $\chi^2_{k=3}$ gives a cut-off value for $\mathrm{D_M}^2$. Samples with $\mathrm{D_M}^2 \leq 3.66$ are considered part of the distribution while others are considered to be outliers.

2.17. With this cut-off value, we can now express (2.48) for a p-variate distribution with $T = \chi_p^{2^{-1}}(\alpha)$ where α is our degree of confidence that values with Mahalanobis distances less than T belong to a given p-variate Gaussian distribution parametrized with mean $\vec{\mu}$ and covariance Σ.

$$\textit{Outlier Set} = \left\{ \vec{x} \in \Re^p \mid (\vec{x} - \vec{\mu})^{\mathrm{T}} \Sigma^{-1} (\vec{x} - \vec{\mu}) > \frac{1}{\chi_p^2}(\alpha) \right\} \tag{2.53}$$

Similarly, we also have:

$$\textit{Outlier Set} = \left\{ \vec{x} \in \Re^p \mid \sqrt{(\vec{x}-\vec{\mu})^{\mathrm{T}}\Sigma^{-1}(\vec{x}-\vec{\mu})} > \frac{1}{\chi_p}(\alpha) \right\} \tag{2.54}$$

Clearly, we may use either $\mathrm{D_M}^2$ or $\mathrm{D_M}$. As a matter of convention, $\mathrm{D_M}^2$ is used in this text.

Recalling that $\mathrm{D_M}^2$ is sensitive to Σ, the threshold T may not be well placed according to our expected confidence α. Since T is invariant to outlier leverage we would require a robust estimate for Σ in order to detect outliers. Robust outlier detection is therefore dependent on *robust distances* which can only be obtained from robust scatter estimates such as those provided by the Fast-MCD algorithm. With outlier leverage mitigated, α is the only parameter in the decision criteria for outlier identification. Ultimately, outlier leverage is the key underlying motivation for outlier identification and the breakdown point of the estimators should be considered before forming any expectation about the sample set.

Chapter 3

Problem Statement & Technical Contributions

3.1 Introduction

When a context has changed significantly within a qualified scene, an intelligent system identifies this event as an anomaly. While scene and context features can vary across observers, the existence of a detection mechanism for significant context change is a salient feature of intelligent observers. By detecting an anomaly, an intelligent system can apply a fitting control law to accommodate the new context or initiate learning to adapt or discover a new control law that is appropriate to maintain stability the presence of the altered context without compromising previously established control laws.

The method described here for anomaly detection and localization first attempts to statistically characterize wavelet filtered sub-bands which is especially important when the narrow band power of an anomaly is insignificant when compared to that of the wide band signal. By distinguishing between extreme and outlier Teager energy values that have appeared in the sub-bands of array sensor data. The outlier data in the time-frequency window can then be used to estimate array phase data required for computing acoustic wavefront direction of arrival in the acoustic far-field.

Chapter 2 provides a theoretical foundation that is critical to the objective of this work. Adding to this foundation, this chapter describes the objective of this work in great detail through a clear statement of the problem of interest as well as a supporting solution.

3.1.1 Problem Statement

An *anomaly* is the specific event that causes the violation of a process observer's expectations about that process[10]. The problem of *anomaly detection and localization* is to determine the spatial information about an energy source that has caused a violation of a process observer's expectation about its environment. On the assumption that the environment contains multiple sources whose characteristics are unknown and whose energy output can change slowly over time, an energy sensor array is used to extract spatial information about the environment. The sources are each assumed to have energies with an arbitrarily complex time-frequency signatures which means no assumption is made about its spectral content or its duration. The problem directly concerns itself with the design of an *observer system* that retains the afore mentioned properties and operates over discrete time-series samples. The problem of *acoustic* anomaly detection and localization is specific for acoustic energy and consequently, the observation data is received from acoustic sensor array.

3.1.2 Thesis Hypothesis

The problem of *anomaly detection* can be specified as a problem in robust statistics. By assuming that all sources provide a modulating force on a spring-mass system, the total energy of the source can be determined using the Teager energy operator. A measure of both potential and kinetic energy, this operator is sensitive to both signal amplitude and signal frequency.

Since an anomaly can be caused by a wide or narrow-band energy source, spectral discrimination should improve the likelihood of detecting an anomalous energy source where its narrow band energy is insignificant to that of the wide band signal it is immersed in. With most of its energy in the lower spectral bands, linear spacing of band-pass filters is not very efficient. With more weight on the lower spectral bands, non-linearly spaced band-pass filters provide spectral decomposition of the acoustic signal whose Teager energy provides the energy contribution of that band.

Over short observation periods, statistical deviations in sub-band Teager energy samples provide some indication that there was some significant event in that sub-band. Collectively, if the energy in each of the filtered sub-bands changes significantly over a larger observation period, then the significance of events in the sub-bands are given more weight and are used for localization in the far acoustic field.

3.2 Characterizing Signals using Teager Energy

To detect a signal anomaly, a baseline for what is considered normal is first required. On the expectation that this baseline should not change significantly over short observation periods, a metric for signal deviation will provide a means for determining if the expectation for consistent normality has been violated. This section concerns itself with the use of sub-band Teager energy as a means of signal characterization over short observation periods, as well as the characteristics of an anomalous event.

3.2.1 The Modulating Source Assumption

Energy change in a signal can be induced in different ways. The classic energy definition implies that its measure is sensitive to amplitude variations. According to (2.14), a signal with *unipolar impulses* appearing with frequency f_0 can have the same energy as a bipolar signal with impulses alternating at the same frequency. A *sinusoidal signal* at this same frequency will not have the same energy. This simple example highlights that it is unreliable to characterize signal shape using classic energy measures. If a system were designed to follow energy changes in this sense, it would be insensitive to abrupt, short-duration changes in the signal shape. It is certainly possible that over successive observation periods that signals with completely differing shapes would result in similar or identical energies. Figure 3.1 shows signals that quite different yet have the same energy measure. Detecting anomalies in amplitude using this measure may pose serious difficulty if possible at all since this measure of energy is only sensitive to the mean amplitude of the absolute signal value over a time interval.

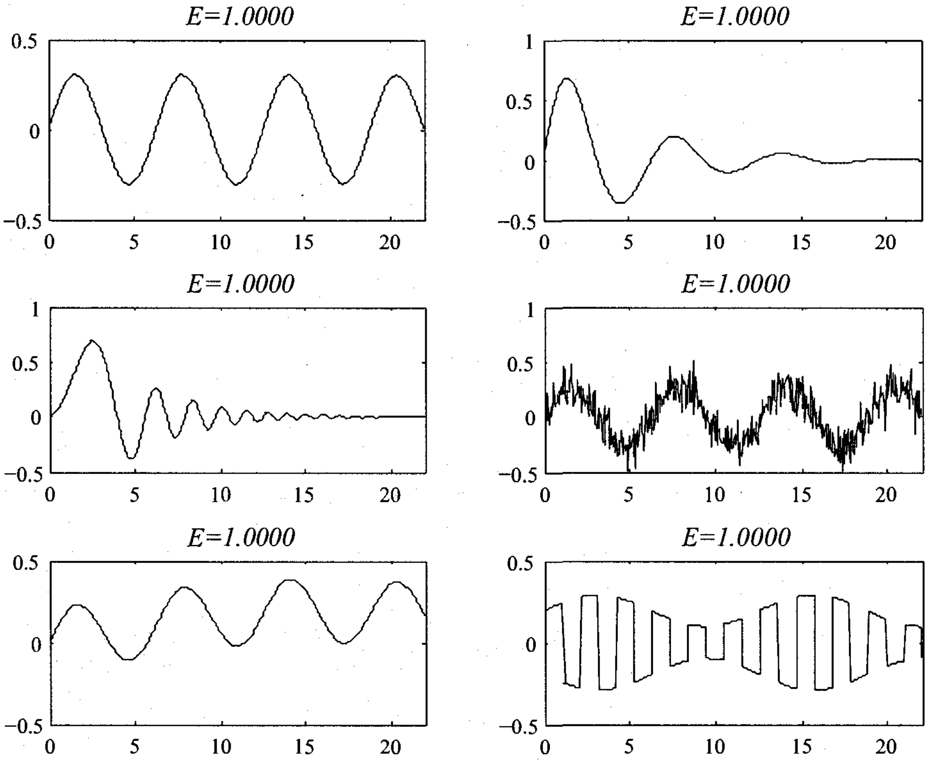


Figure 3.1: Various signals with identical, unit energy measured according to the definition of (2.14).

If we assume that all signals of interest have been *modulated* somehow, then *demodulation* would expose other features of the signal that could help characterize it.

For example, AM demodulation would expose changes in the signal's envelope for relatively constant frequencies. FM demodulation would expose changes in frequency for a relatively fixed signal envelope. Finally AM-FM demodulation would simultaneously expose both of these features. The Teager energy operator has interesting demodulation properties that can be used for amplitude and frequency feature extraction. Figure 3.2 shows the Teager energies for the signals of Figure 3.1 respectively. Each has a characteristically different instantaneous and averaged energy using the Teager energy operator. If we make the assumption that the signal source is modeled as a modulating force on a spring-mass system, we can make use of the demodulation properties of the Teager energy operator (Section 2.2.4) for joint amplitude and frequency feature extraction. We will call this the *modulating source assumption.*

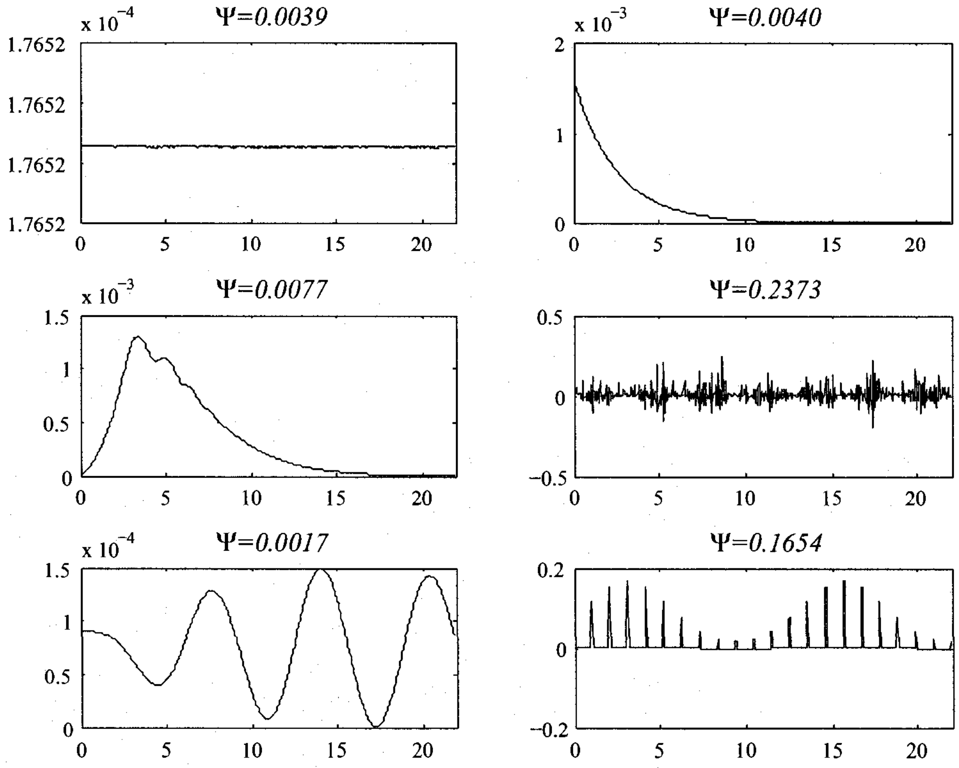


Figure 3.2: Instantaneous Teager energy of signals in Figure 3.1, with their *average* Teager energies (Ψ) in the sub-figure headings.

3.2.2 Measuring Teager Energy in Sub-Bands

A simple source is characterized rather well with the *modulated source* assumption. Complicated signals however, become increasingly difficult to characterize in this fashion because of the simplistic Teager source model. This difficulty is also experienced where multiple sources with differing properties are present.

With the presence of many differing sources, or a single complex source, the sample probability density function approaches a central attractor distribution (*the central limit theorem*). Consequently, its spectral bandwidth becomes increasingly well defined however, information about the contributing sources is completely lost with each of their individual distributions convolved with the others. Since we typically deal with few sources and cannot deal with the attractor distribution, characterization of the signal will have to be broken down somehow. On the assumption that a complicated source has multiple, simpler sources contributing in sub-bands, Teager energy characterization of sub-bands may be more representative of the complex source. We therefore consider some of the underlying concerns with sub-band Teager energy characterization.

3.2.3 Band-Pass Filtering

Among other specifications, the *filter order* is very much of concern and typically defines the behavior of the filtered signal in the stopband. Figure 3.3 shows the frequency response of an elliptic band-pass filter of order 12. Clearly, the stopbands are not ideal, but acceptable depending on the application. Ideally, we would like to have flat passbands, stopbands with a perfectly sharp transition. In this theoretical sense, we could measure the total energy(E_T) in a signal from the energies in its sub-bands(E_i):

$$E_T = \sum_{i}^{N} E_i \tag{3.1}$$

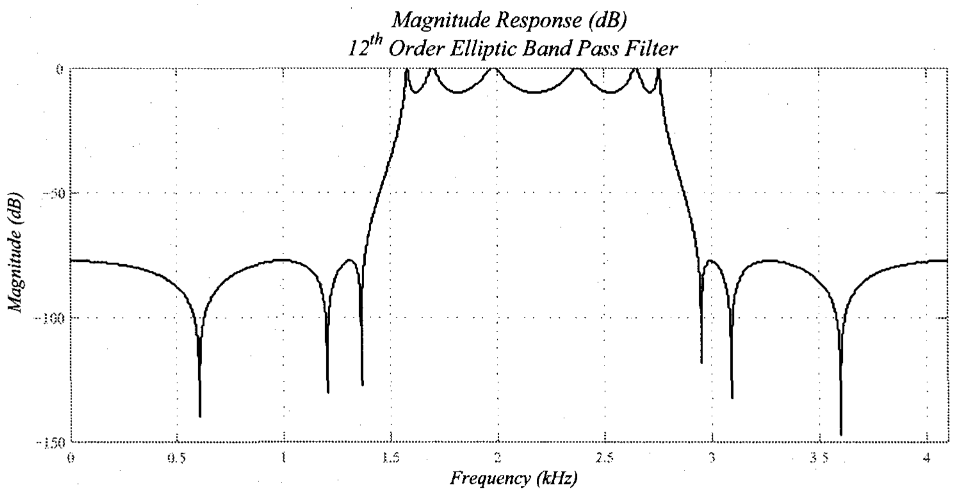


Figure 3.3: The magnitude response of a typical band-pass filter: Elliptic IIR, Order 32.

Where i would denote the index of any number of theoretical band pass filters spanning the frequency of the input signal. Superposition in this sense could not be applied if the filters were not theoretically perfect. Theoretical band-pass filters as such cannot be designed and typically display sidelobes in the stop-bands that allow undesired signal spectra to pass. Any energy measurement from non-ideal band-pass filters will contain surplus energy from the stop-band. Another problem with band-pass filtering where *total energy* is concerned, is where to place the center frequency of the filters. Since the transition bands are not perfect, this poses an extra degree of freedom that makes energy measurement far more subjective than we would like (because of the overlap of imperfect filters). While possible to obtain reasonable energy estimates, good filters come with a computational expense that increases with the number of bands required as well as the order of the filter.

3.2.4 Advantages of Wavelet Decomposition

Discrete sampling of a signal offers a great deal in the way of processing flexibility. One of note in the context of this work is the discrete wavelet transform (DWT).

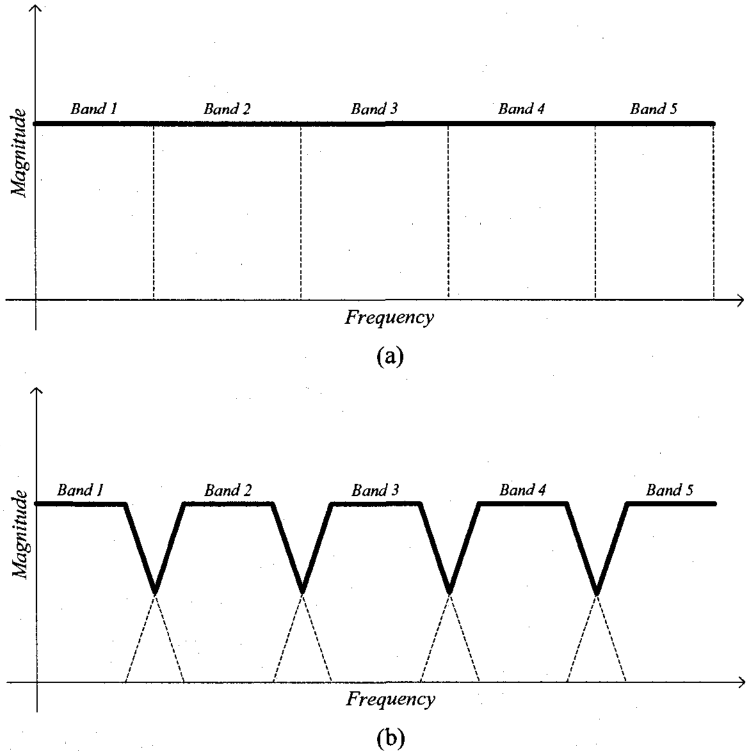


Figure 3.4: Ideal band-pass filtering would allow for all sub band energies to be accounted for with no overlap between filters and no areas of magnitude attenuation as in. In (a) the passband is ideal with no transition band. The sum of energies in each of the sub-bands is equal to the total energy ($E_T = \sum E_i$). In (b), transition bands are clearly present with overlap in the stopbands. It is not clear how much of the total energy is accounted for in each of the sub-bands with this non-ideal band pass filtering ($E_T \neq \sum E_i$).

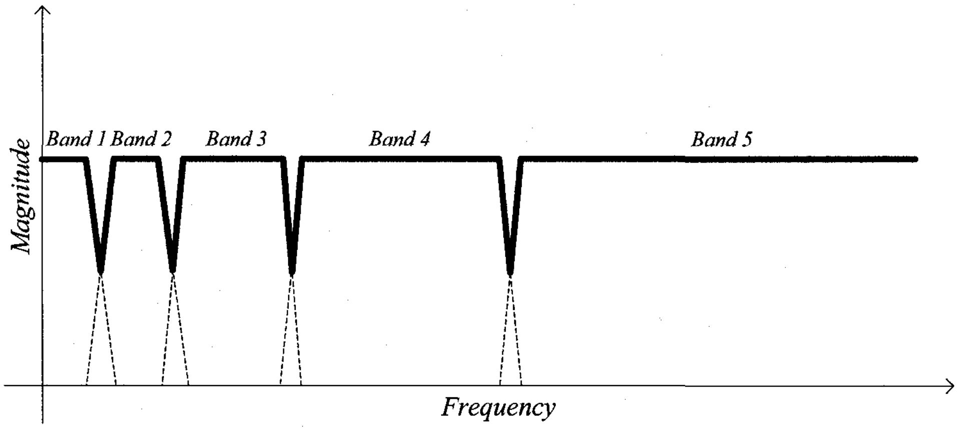


Figure 3.5: A scaled wavelet function acts as a band-pass filter, halving its bandwidth for each decomposition level. Note that the center frequency of each band is nonlinearly spaced across the signal spectrum and that the transition bands are quite sharp, although not ideal.

While typically used for time-frequency analysis of signals, it does have another use: band-pass filtering. A scaled wavelet function will act as a band-pass filter halving its bandwidth for each decomposition level as shown in Figure 3.5. A direct result of the dyadic sampling scheme used in the DWT, the center frequency of the band-pass filters are nonlinearly spaced across the sampling spectrum. The wavelet's scaling function prevents the existence of an infinite number of bands to cover the full spectrum effectively limiting the number of filters that are produced to a small number. If sub-band energy characteristics are sought, then the spectrum sampling scheme should be chosen diligently. Audio data normally has most of its power in the lower spectra therefore we would expect most of the information in the signal would appear in these bands. In this case, we would choose to have a finer spectral discrimination for this type of signal. Discrete wavelet decomposition is very good in this case since the band-pass filtering has many of the filters in the lower frequencies and fewer in the higher frequencies. In addition to the sharp transition bands offered by wavelets, the natural nonlinear spacing of the band-pass filters is a good choice for audio analysis and comes at minimal cost.

As mentioned in Section 2.2.5, the Teager energy operator is very sensitive to noise. Typically appearing as a low-amplitude, high frequency modulation of a low frequency baseband source, the Teager energy operator will tend to give more negative values. Removing noise is advantageous in this case to ensure that the energy operator produces positive values as we would normally expect. Once again, wavelet analysis can provide a good solution for this problem. With minimal computational effort, de-noising is possible by thresholding wavelet coefficients in some bands. This will improve performance of the Teager energy operator. This optional preprocessing step was not used in this work.

Optimal selection of a specific wavelet basis is highly dependent on the qualities of the for audio source and is not the topic of this work although some selection criteria is of concern. The real projection of a signal onto an orthogonal basis results in scalar coefficients. The energy in the resulting signal approximation is a sum of the energies in each basis projection. It is reasonable therefore to use a wavelet basis that has an instantaneous energy that is positive over its area of support. For this work, three wavelet families were considered. Figure 3.6 shows examples of the symlet(a), Daubechies(b), and discrete Meyer(c) wavelets. All three wavelets have negative Teager energies which may not be apparent given the scaling in plots (d) and (e). Because of the noisy bipolar nature of their Teager energy shown in (f), Daubechies wavelets were not used. A large sample of white noise was filtered using the symlet-8 and discrete Meyer wavelets to obtain an empirical impulse response for each of four band-pass filters corresponding to an increasing level in decomposition. Figure 3.7 shows the power spectral density of each filter output superimposed on the upper plot. Clearly, the stop-bands of the symlet-8 filters contain side lobes that are not desirable for this application of a band-pass filter. When the same white noise sample is filtered using the discrete Meyer wavelet, we notice right away that side lobes are minimized, and even eliminated for some bands. This filter resembles very

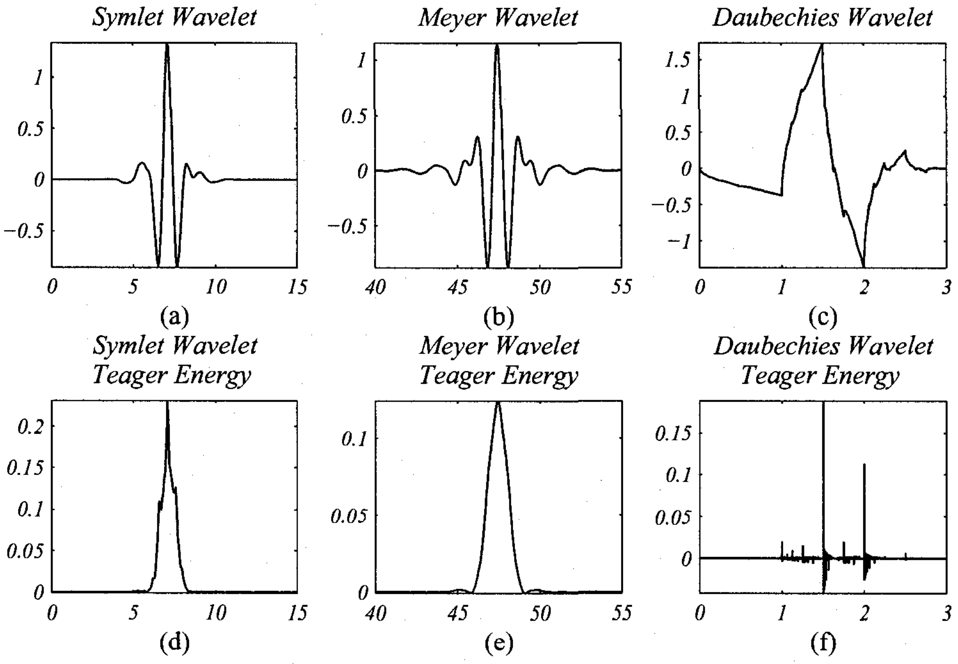


Figure 3.6: Teager Energy of selected wavelets: Symlet-8, Discrete Meyer and Daubechies-2.

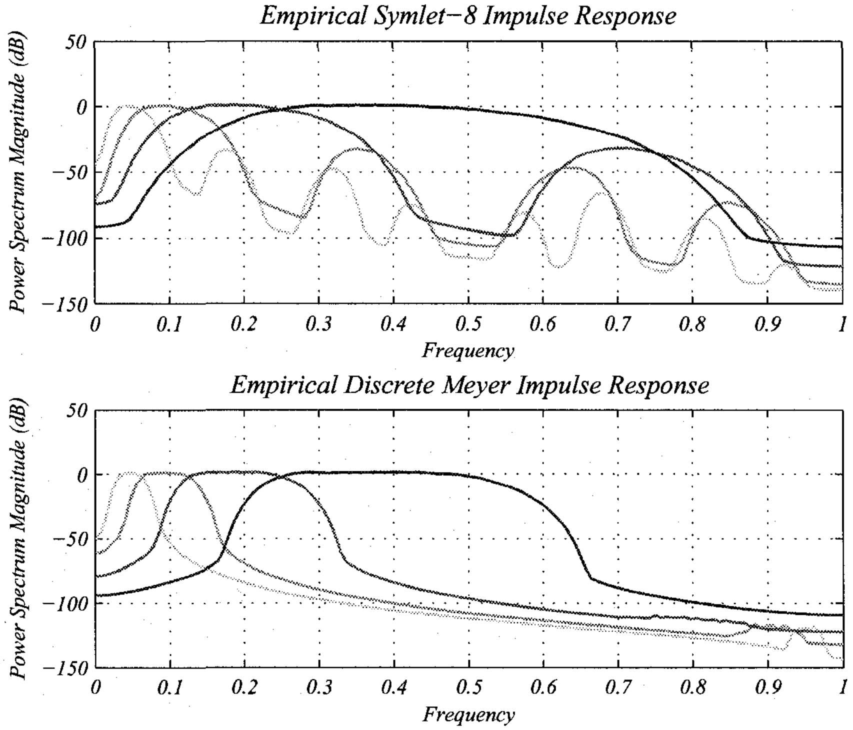
much like the desired filter of 3.5. We therefore reject the Symlet-8 wavelet in favor of the *discrete Meyer wavelet* in this work.

3.2.5 Teager Energy in Sub-bands

With appropriate band-pass filtering, energy measurement can be done in sub-bands of a signal. Our functional definitions thus far in Equations 2.23 and 2.24 define *instantaneous Teager energy.* We now introduce another measure, the *average Teager energy* for a discretely sampled signal in a particular sub-band:

$$\overline{\Psi}_W(x_n) = \frac{1}{W}\left|\sum_{n=1}^{W} x_{n,s}^2 - x_{n-1,s}x_{n+1,s}\right| \tag{3.2}$$

Where W is the number of samples, s is the index of a particular sub-band of interest, and $x_{n,s}$ is a particular sub-band sample within a sample set. Very similar in form to Equation 2.24, this non-causal definition serves three purposes:



Empirical Symlet−8 Impulse Response

Empirical Discrete Meyer Impulse Response

Figure 3.7: Empirical impulse responses obtained from wavelet decomposition of a large sample of white noise. Symlet-8 filtering produces large side lobes in all stopbands for each level of wavelet decomposition. Filtering of the same sample using the discrete Meyer wavelet produces practically no side lobes which is very desirable (see Figure 3.5).

1. The power measurement will give a single central-tendency measure for the energy levels within a localized time period reducing the amount of data to process.

2. Measuring power within a shifting time period is essentially a moving average for energy which acts as a low pass filter, mitigating problems with noisy energy processing.

3. If the negative Teager energy appears within the measurement period, the averaging function will offset the rare negative energy with the more common positive energy producing a more positive measure which is the central tendency of the Teager energy measure.

The averaging period (measured by the number of samples, W) effectively defines the *sensitivity* of our energy measurement. In choosing a value for W the goal is to try to maintain the positivity of Ψ_s while keeping as many samples as possible for statistical characterization. This can be done for any number of samples, but fewer samples can only provide a rough empirical probability density function which may not be useful for any estimation of central tendency or scatter. A large number of samples will give better estimates in general and for very large W, the density function will approach the attractor Gaussian distribution of the central limit theorem.

3.3 Teager Energy Preprocessing

A deterministic signal will have deterministic Teager energy. A discretely sampled acoustic signal in the context of this work is considered to be a stochastic time series that is wide-sense stationary. The Teager energy operator is a system that transforms the distribution of its random input to an output with another distribution. Consequently, the instantaneous Teager energy of these signals are also a stochastic time series. For Laplace distributed audio input, the Teager energy operator has an analytically complicated distribution. While it is certainly possible to find this distribution, it should be noted that for a Gaussian input, Teager energy is typically log-Gaussian. Through a successive transformation of random variable, it is possible to characterize Teager energy of a Laplace distributed input with a parametrized Gaussian distribution approximation in each sub-band. This section describes this process as a means of statistically characterizing Teager energy from a Laplacian source.

3.3.1 Approximating the Log-Gaussian Distribution for Teager Energy

The Teager energy operator can be considered as a memory system $\mathrm{T}(\cdot)$ with one random variable input X and one random variable output Y:

$$Y = \mathrm{T}(X) \tag{3.3}$$

Since this work concerns itself with characterizing Laplace distributed audio and its Teager energy in sub-bands, the distribution of Y is of interest for Laplacian X. The point of paramount concern here is that Teager energy for a Laplacian input is non-trivial. The methods discussed in Chapter 2 require a Gaussian random variable and this is certainly not the case with the non-trivial distribution. Outlier detection in this fashion is not possible.

Through experimentation, it was found that if $T(X) = \overline{\Psi}_W(X)$ (See (3.2)) for any sub-band signal, where X is a Gaussian random variable, Y in this case is approximately log-Gaussian distributed for some values of W, the averaging window length. For selected values of W, the following experiment was performed using a computer:

1. Generate uniform random variable U: $P_U\{-0.5 \leq U \leq +0.5\} = 1$

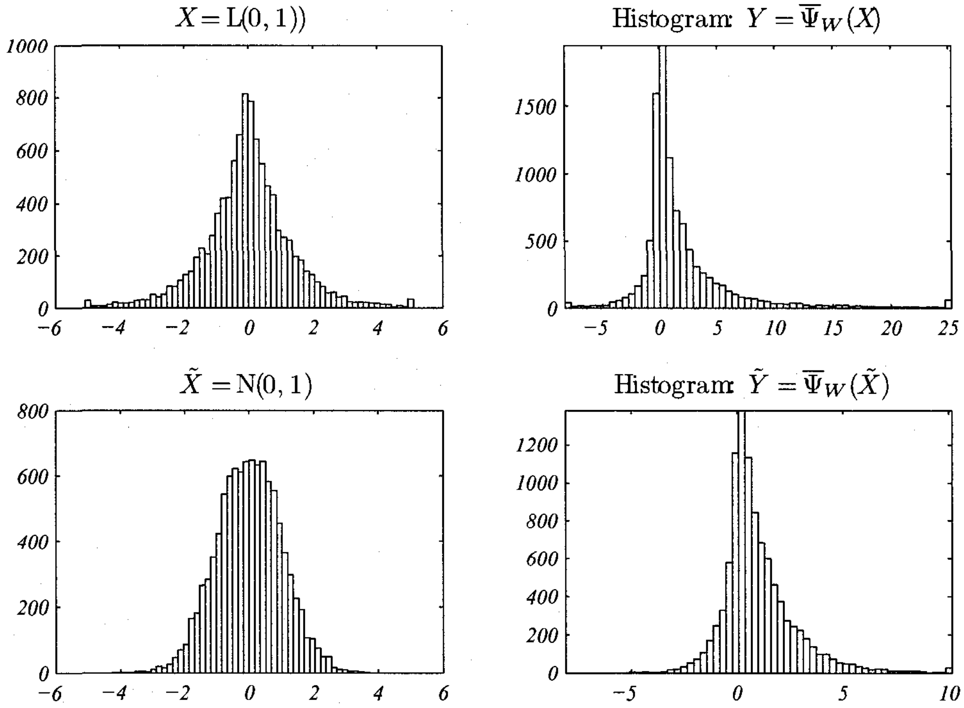


Figure 3.8: A comparison of histograms for Teager energy output. X is a Laplacian, and $\tilde{X}$ is a Gaussian distributed input to the operator. Y and $\tilde{Y}$ are the output histograms (respectively). After computer analysis, it is unclear what the resulting distribution is.

2. Generate Laplacian random variable $X = \mathrm{L}(0, 1)$, where

   $\mathrm{L}(\mu, b) = \mu - b\ \mathrm{sgn}(U) \ln(1 - 2|U|)$

3. Generate the Averaged Teager Energy $Y = \overline{\Psi}_W(X)$, where W is a selected window size.

4. Analyze Y.

The effect of W is very important in this case since it has a direct bearing on the resulting distribution. For illustrative purposes, consider the case where $W = 1$, where (3.2) essentially reduces to (2.24). For an experiment with ten-thousand samples Figure 3.8 shows the resulting histograms for a non-windowed Teager energy transformation given Laplacian random variable X and a Gaussian random variable $\tilde{X}$ for comparison. While computer analysis confirms that X is indeed Laplacian,

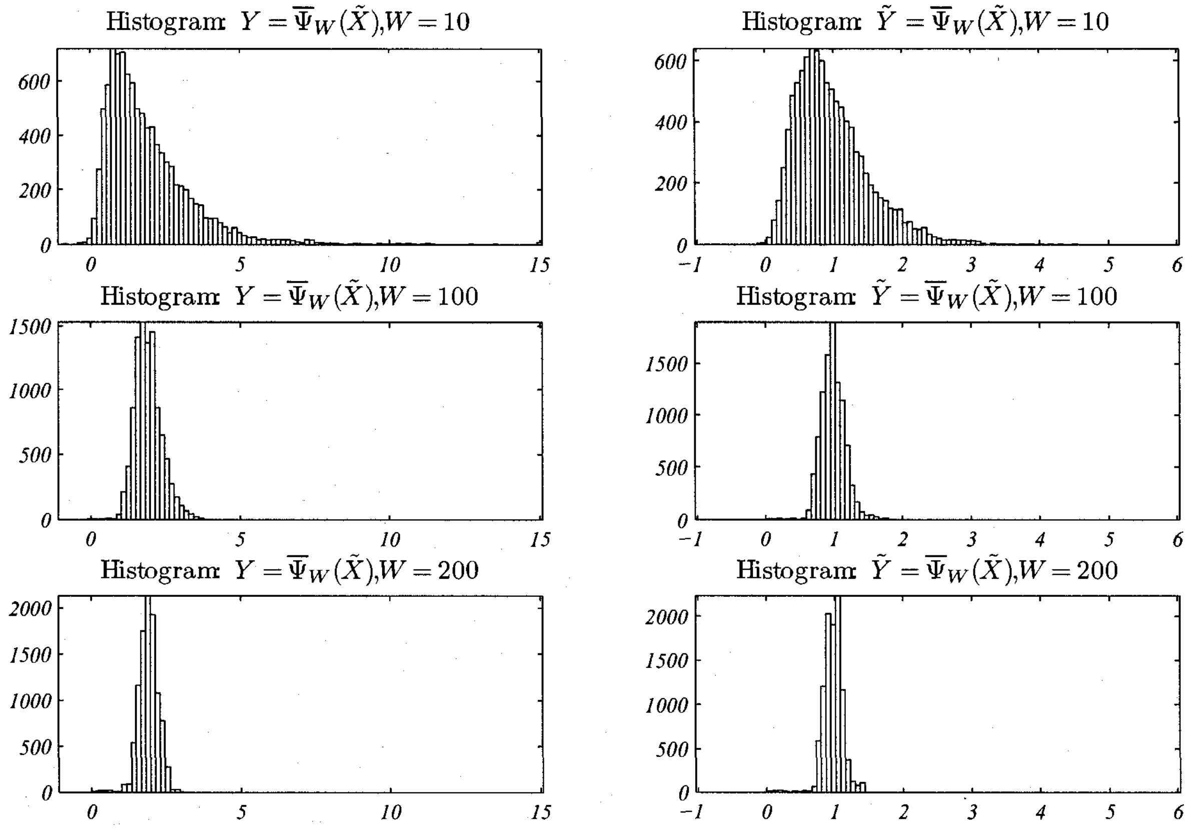


Figure 3.9: Histograms for Teager Energy outputs given Laplacian (X) and Gaussian ($\tilde{X}$) inputs. As W increases, the empirical distributions become log-Gaussian for moderate values and increasingly Gaussian for large values.

the distribution of Y was not of any standard distribution with $W = 1$ when computer analyzed against several known distributions. Large values of W succeeded in reducing the number of negative values, but reduced operator sensitivity. $\tilde{Y}$ had similar problems but for smaller W preserving sensitivity while reducing and even eliminating negative values. The comparison is shown again for differing values of W in Figure 3.9. As W increases, the histogram reflects an underlying log-Gaussian distribution with a limitation:

$$\lim_{W \to \infty} \mathrm{P}_{\overline{\Psi}_W(X)} \to N(\mu, \sigma) \tag{3.4}$$

which is expected given the central-limit theorem. A large W is not desirable since higher energy frequencies within the band are removed (low-pass filtering of the sub-band energy) and the averaged Teager energy operator becomes less sensitive to short-duration events.

Although optimal selection of W was not investigated, it is known that large values are not desirable, and moderate values result in a log-Gaussian distribution of Teager energy for a Gaussian input. In preparation for outlier detection using Mahalanobis distances, transforming the Laplacian distributed audio data into a Gaussian distribution before processing Teager energy will result in a distribution that can be easily converted to a Gaussian distribution. If the transformation $Y = G(X)$ transformation is one-to-one, then it can be used for outlier detection since:

$$\mathrm{P}\{Y = y_i\} = \mathrm{P}\{X = x_i\} \tag{3.5}$$

where x_i is a sample from the input distribution and y_i is the corresponding output[16]. This can be extended to any number of transformations provided they are all one-to-one. If successive transformation results in a Gaussian distribution, then outlier detection can be performed with a reliable Mahalanobis distance threshold.

3.3.2 Random Variable Transformation

A double-sided exponential, the Laplace distribution has much higher kurtosis than the Gaussian distribution although they are both symmetrical and asymptotic (see Figure 3.10). As explained in Section 2.3, given a confidence interval, outlier identification is possible only with a Gaussian distributions using the normalized distances. Certainly, a maximum-likelihood scatter estimator can be designed for the Laplace distribution, and Mahalanobis distances computed. The decision criteria (2.54) cannot be used however since the distances are no longer Chi distributed. In order to use the decision criteria, the data would have to be transformed into a Gaussian

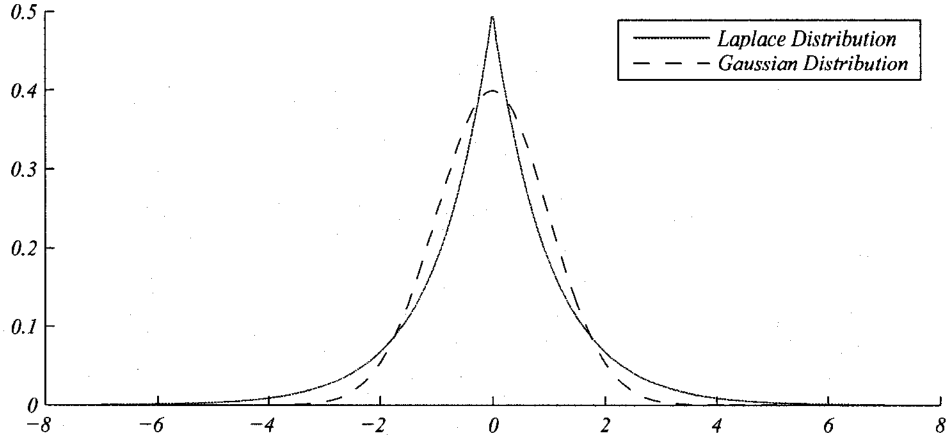


Figure 3.10: A Laplace distribution has higher kurtosis than Gaussian distribution. Shown here are L(0, 1) and N(0, 1) respectively.

distribution.

Given the known cumulative distribution $\mathrm{F}_x(\mathbf{x})$ of the random variable $\mathbf{x}$, we can find the function $\mathbf{y} = \mathrm{g}(\mathbf{x})$ for a specified cumulative distribution $\mathrm{F}_y(\mathbf{y})$. In the general case [16], we find that if $\mathbf{y} = \mathrm{F}_y^{-1}(\mathrm{F}_x(\mathbf{x}))$ then $\mathrm{P}(\mathbf{y} \leq y) = \mathrm{F}_y(\mathbf{y})$. Therefore, if $\mathbf{x}$ is Laplacian distributed audio data, it can be redistributed into a Gaussian distribution. The cumulative Laplace distribution is given in (3.6), the cumulative Gaussian distribution is given in (3.7), and the inverse cumulative Gaussian distribution is given in (3.8).

$$\begin{aligned}
\mathrm{F}_x(\mathbf{x}) &= \frac{1}{2}\left\{1 + \mathrm{sgn}(\mathbf{x} - \mu) \cdot 1 - \exp\left(-\frac{|\mathbf{x} - \mu|}{b}\right)\right\} && (3.6) \\
\mathrm{F}_y(\mathbf{y}) &= \frac{1}{2}\left\{1 + \mathrm{erf}\left(\frac{\mathbf{y} - \breve{\mu}}{\sigma\sqrt{2}}\right)\right\} && (3.7) \\
\mathrm{F}_y^{-1}(\mathbf{y}) &= \mathrm{erf}^{-1}(2\mathbf{y} - 1) \cdot \sigma\sqrt{2} + \breve{\mu} && (3.8)
\end{aligned}$$

The Laplacian random variable $\mathbf{x}$ has two parameters which are estimated from sampled audio data: the mean μ, and variance $2b^2$. For Laplacian distributed sample data, the mean is the *sample median* and the following estimator is used to find the parameter b of the variance from sample data:

$$\hat{b} = \frac{1}{N}\sum_{i}^{N} |x_i - \hat{\mu}| \tag{3.9}$$

Where $\hat{\mu}$ is the *sample median* of the sample x_i's, and $\hat{b}$ is the estimate of b, and N is the number of samples. The transfer function $\mathrm{g}(\mathbf{x})$ that will transform a Laplacian random variable $\mathbf{x}$ into a Gaussian random variable $\mathbf{y} = \mathrm{g}(\mathbf{x})$ is obtained from the following:

$$\begin{aligned}
\mathrm{g}(\mathbf{x}) &= \mathrm{F}_y^{-1}\left(\mathrm{F}_x(\mathbf{x})\right) && (3.10)\\
&= \mathrm{erf}^{-1}\left(2\cdot\frac{1}{2}\left\{1 + \mathrm{sgn}(\mathbf{x}-\mu)\cdot 1 - \exp\left(-\frac{|\mathbf{x}-\mu|}{b}\right)\right\} - 1\right)\cdot\sigma\sqrt{2} + \breve{\mu}\\
&= \mathrm{erf}^{-1}\left(\mathrm{sgn}(\mathbf{x}-\mu)\cdot 1 - \exp\left(-\frac{|\mathbf{x}-\mu|}{b}\right)\right)\cdot\sigma\sqrt{2} + \breve{\mu}
\end{aligned}$$

The direct algebraic substitution of (3.10) leaves the parameters σ and $\breve{\mu}$ from the Gaussian random variable, and may be taken as constants, although this is not very useful. For example, any Laplacian random variable with a sample-estimated mean and variance can be transformed into a zero-mean Gaussian random variable with unity variance. Fully parameterizing the target distribution with constants is not very useful to characterize signals. In order to characterize a changing signal, σ and $\breve{\mu}$ should be dependent on the input random variable. To achieve this we let $\breve{\mu} = \mu$, and $\sigma^2 = 2b^2$, both being the mean and variance of the Laplacian random variable. The target distribution is therefore $\mathrm{N}(\mu, 2b^2)$:

$$\begin{aligned}
\mathrm{g}(\mathbf{x}) &= \mathrm{F}_y^{-1}\left(\mathrm{F}_x(\mathbf{x})\right)|_{\breve{\mu}=\mu,\ \sigma^2=2b^2} && (3.11)\\
&= \mathrm{erf}^{-1}\left(\mathrm{sgn}(\mathbf{x}-\mu)\cdot 1 - \exp\left(-\frac{|\mathbf{x}-\mu|}{b}\right)\right)\cdot\sqrt{2}b\sqrt{2} + \mu\\
&= 2b\,\mathrm{erf}^{-1}\left\{\mathrm{sgn}(\mathbf{x}-\mu)\cdot\left(1 - e^{-\frac{|\mathbf{X}-\mu|}{b}}\right)\right\} + \mu && (3.12)
\end{aligned}$$

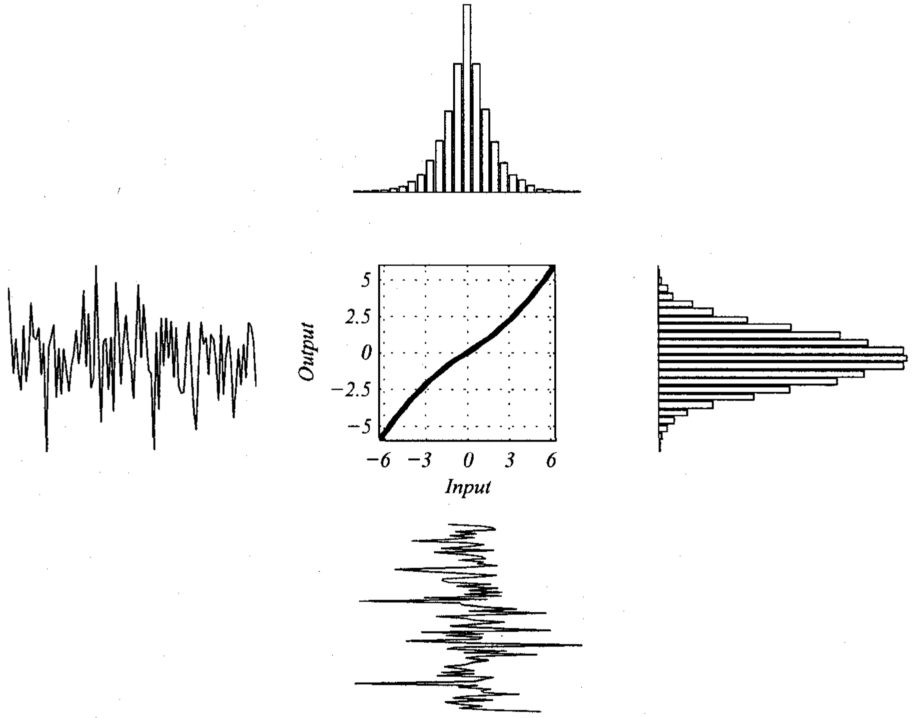


Figure 3.11: The transfer function $\mathrm{g}(\mathbf{x})$ shown in the center plot. $\mathbf{x} = \mathrm{L}(0, 1)$ is shown on the bottom plot with its transformation $\mathbf{y} = \mathrm{g}(\mathbf{x})$ on the left and its histogram on the top. The histogram of $\mathbf{y}$ clearly shows a the desired correction in kurtosis offered by a Gaussian distribution.

If we wish to *estimate* $\mathrm{g}(\mathbf{x})$ we can now make full use of the sample estimates:

$$\hat{\mathrm{g}}(\mathbf{x}) \quad = \quad 2\hat{b}\,\mathrm{erf}^{-1}\left\{\mathrm{sgn}(\mathbf{x}-\hat{\mu})\cdot\left(1-e^{-\frac{|\mathbf{X}-\hat{\mu}|}{\hat{b}}}\right)\right\}+\hat{\mu} \qquad (3.13)$$

Where $\hat{\mathrm{g}}(\mathbf{x})$ is the estimated random variable transform that will redistribute a random Laplacian variable into a random Gaussian random variable. For the sake of clarity, the notation $\mathrm{g}(\mathbf{x})$ will be used instead of $\hat{\mathrm{g}}(\mathbf{x})$ with the implicit understanding that distribution parameters are estimated from sample data. Figure 3.11 shows a Laplacian input sample $\mathbf{x}$ that is transformed into $\mathbf{y} = \mathrm{g}(\mathbf{x})$ and is shown on the left. The histogram on the top shows represents Laplacian random variable $\mathbf{x}$, while the histogram on the right highlights its transformation into $\mathbf{y}$, a Gaussian random variable. From the reasoning that led to (3.12), it follows that a log-Gaussian random variable $\mathbf{v}$ can be transformed into a Gaussian distribution using the following:

$$\begin{aligned}
\mathrm{F}_v(\mathbf{v}) &= \frac{1}{2} + \frac{1}{2}\mathrm{erf}\left(\frac{\mathbf{v}-\mu}{\sigma\sqrt{2}}\right) && (3.14)\\
\mathrm{h}(\mathbf{v}) &= \mathrm{F}_y^{-1}\left(\mathrm{F}_v(\mathbf{v})\right) && (3.15)\\
&= \mathrm{erf}^{-1}\left(2\cdot\left\{\frac{1}{2} + \frac{1}{2}\mathrm{erf}\left(\frac{\mathbf{v}-\mu}{\sigma\sqrt{2}}\right)\right\} - 1\right)\cdot\sigma\sqrt{2} + \breve{\mu} \\
&= \left(\frac{\ln\mathbf{v}-\mu}{\sigma\sqrt{2}}\right)\cdot\sigma\sqrt{2} + \mu \\
&= \ln\mathbf{v} && (3.16)
\end{aligned}$$

Given that both (3.12) and (3.16) are one-to-one functions, we may design the following transformed energy function for a Laplacian input [10]:

$$\breve{\overline{\Psi}}_W(\mathbf{x}) = \mathrm{h}\left(\overline{\Psi}_W\left(\mathrm{g}(\mathbf{x})\right)\right) \quad (3.17)$$

Where $\mathbf{x}$ is the audio input to the system, and $\breve{\overline{\Psi}}_W(\mathbf{x})$ is the redistributed Teager energy. Because of the one-to-one nature of $\mathrm{h}(\cdot)$, outliers in $\breve{\overline{\Psi}}_W(\cdot)$ are also outliers in $\overline{\Psi}_W(\cdot)$. This system is used for detecting Teager energy outliers in audio data provided robust estimates for Gaussian location and scatter parameters.

3.4 Robust Anomaly Detection with Teager Energy

Given Laplace distributed audio data $\mathbf{x}$, the Teager energy transformation from (3.17) will be Gaussian distributed. This extends to sub-bands. The more Laplacian the input, the more Gaussian the transformed Teager energy is. This not only applies to the wide band signal, but to sub-bands as well. Band-pass filtering using wavelets (Section 3.2.4) can decompose an audio signal and the transformed Teager energy of spectral bands can be used for outlier detection. Figure 3.12 shows histograms for a real audio sample that has been wavelet decomposed into sub-bands using the discrete Meyer wavelet followed by a transformation in each band by $\breve{\Psi}_W$. Note that the energy histograms in Levels 2, 3, and 4 clearly contain secondary distributions which are considered as outliers. The following section concerns itself with the detection of anomalous events once the signal has been preprocessed according to the methods discussed thus far, specifically those concerning Teager energy.

3.4.1 Minor Anomalies

Mahalanobis distances are dependent on Gaussian location and scatter parameters which are sensitive to the latter (Section 2.3.5) however, the Fast-MCD algorithm (Section 2.3.6) provides a robust estimation for these parameters. The robust Gaussian characterization of $\breve{\Psi}_W$ is very much dependent on whether or not the proportion of outliers has exceeded the breakdown point. If the breakdown point was not exceeded then the estimates are robust to outliers and consequently, *robust distances* can be used to identify outliers. For a sub-band sample buffer $\delta_{s,i}$ of length N we define the set of Teager energy outliers at time-index i in sub-band s:

$$\begin{aligned}
\delta_{s,i} &= \{x_i, x_{i-1}, x_{i-2} \ldots x_{i-N-1}\} && (3.18)\\
\tilde{\mu}_{\delta_{s,i}} &: \textit{Robust mean of all elements in } \delta_{s,i} && (3.19)\\
\tilde{\Sigma}_{\delta_{s,i}} &: \textit{Robust covariance of all elements in } \delta_{s,i} && (3.20)\\
\tilde{\mathrm{D}}^2_{\mathrm{M}}(\vec{x}\ ;\ \tilde{\mu}, \tilde{\Sigma}) &= (\vec{x} - \tilde{\mu})^{\mathrm{T}} \tilde{\Sigma}^{-1} (\vec{x} - \tilde{\mu}) && (3.21)\\
\mathbf{a}_{s,i} &= \left\{ \exists x \in \delta_{s,i} \mid \tilde{\mathrm{D}}^2_{\mathrm{M}} \left(\breve{\Psi}_W(x)\ ;\ \tilde{\mu}_{\delta_{s,i}}, \tilde{\Sigma}_{\delta_{s,i}} \right) > \frac{1}{\chi^2_k}(\alpha), k = 1 \right\} && (3.22)
\end{aligned}$$

Where $\tilde{\mathrm{D}}^2_{\mathrm{M}}(x_i\ ;\ \tilde{\mu}, \tilde{\Sigma})$ is the robust Mahalanobis distance which uses robust Fast-MCD estimates $\tilde{\mu}$ and $\tilde{\Sigma}$ instead of those specified in (2.47). Since $\breve{\Psi}_W(x_i)$ is a scalar, there is $k = 1$ degree of freedom for the Chi distribution. Figure 3.13 shows a histogram of time-series data collected over 850 samples. The $\mathrm{N}(0, 1)$ source contains contaminants

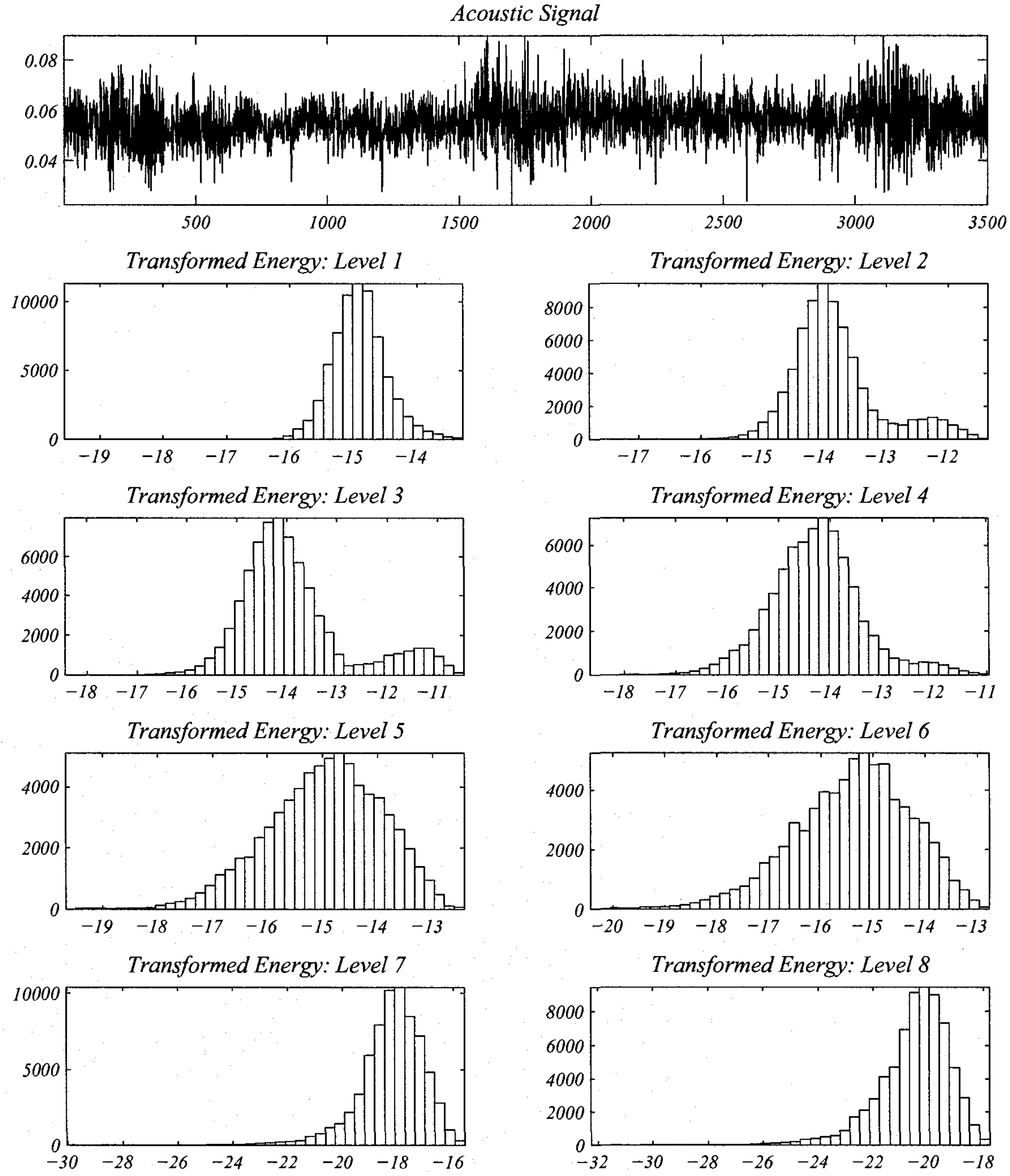


Figure 3.12: histograms for a real audio sample that has been wavelet decomposed into sub-bands using the discrete Meyer wavelet followed by a transformation in each band by $\overset{\smile}{\overline{\Psi}}_W$. The energy histograms in Levels 2, 3, and 4 contain secondary distributions which are considered as outliers.

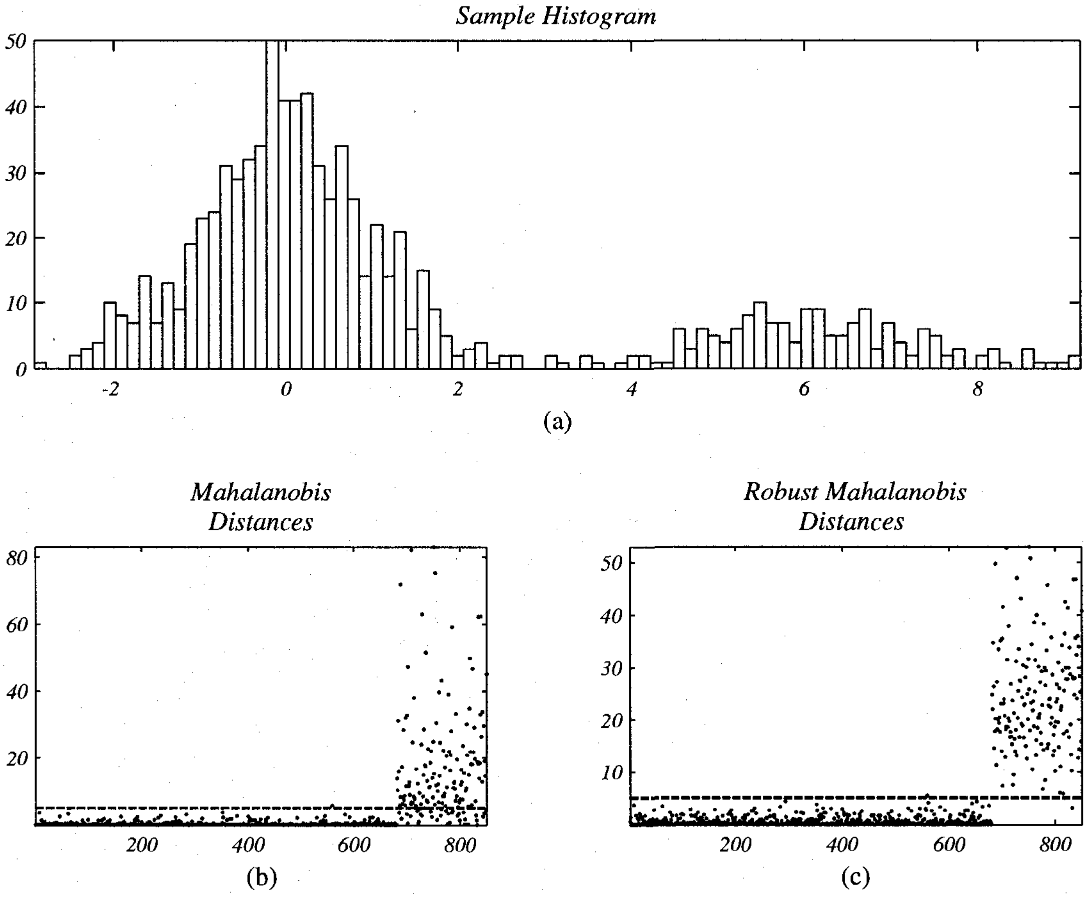


Figure 3.13: Outlier separability comparison: normal, and robust Mahalanobis distances. The data here is from two sources: N(0, 1) and N(6, 1.2) with threshold set by $\alpha = 0.975$.

from a N(6, 1.2) source for 20% of the samples. On the top is a histogram of the sample. On the bottom left, maximum-likelihood estimates for location and scatter cannot be thresholded to provide a good separation for the contaminants which are displayed to the right of the plot. On the bottom right, robust estimates are used and are easily thresholded to separate outliers.

We define any member of $\mathbf{a}_{s,i}$ as a *minor anomaly* where i is the time index for the computation over the N-length buffer. Certainly, minor anomalies can appear as single samples and this may pose a problem for localization. To mitigate this, we can introduce another parameter η which is used to reject singletons. With a default of $\eta = 1$ (singletons allowed), we place an additional restriction on $\mathbf{a}_{s,i}$: If there is a sequence of anomalies whose length is less than η, then the sequence is excluded from $a_{s,i}$. This restriction reflects the reasoning that *samples themselves are not anomalies, sample sequences are.*

As i advances in time, the proportion of outliers may increase past the breakdown point of the Fast-MCD algorithm. In this case, the estimates are very similar to those obtained by maximum-likelihood methods which account for all samples rather than a subset. This is not an undesirable effect since $\tilde{\mathrm{D}}^2_{\mathrm{M}}(x_i \,;\, \tilde{\vec{\mu}}, \tilde{\Sigma})$ will decrease as outliers increase in number because of their masking effect (caused by outlier leverage). This decrease in robust distance corresponds to a desensitization to the outlier's energy qualities. This *habituation* effect due to breakdown is desirable. Anomalies when in sufficient number, are not considered anomalies (no anomalies exist when breakdown occurs).

The *attention span* $|\delta_{s,i}| = N$ [10] clearly defines the length of the sample window used in the detection of minor anomalies. Since the robust estimates $\tilde{\vec{\mu}}$ and $\tilde{\Sigma}$ are for the *majority* of samples over that period, we note that for large N, the method becomes insensitive to localized outliers, while a small N results in s hypersensitivity to outliers. The attention span therefore is a measure of sensitivity.

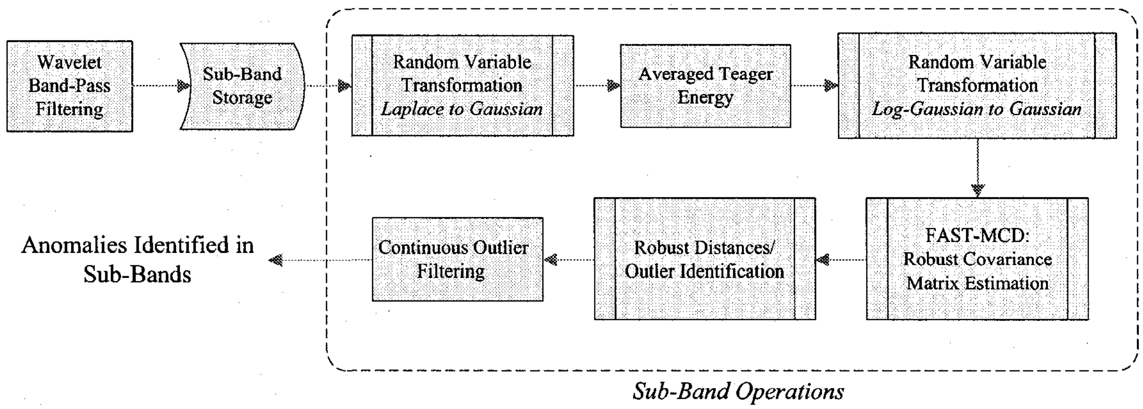

Wavelet Band-Pass Filtering
Sub-Band Storage
Random Variable Transformation *Laplace to Gaussian*
Averaged Teager Energy
Random Variable Transformation *Log-Gaussian to Gaussian*
FAST-MCD: Robust Covariance Matrix Estimation
Robust Distances/ Outler Identification
Continuous Outlier Filtering
Anomalies Identified in Sub-Bands
Sub-Band Operations

Figure 3.14: A block diagram of the sub-band minor anomaly detection system.

A summary of the sub-band minor anomaly detection system described thus far is shown in Figure 3.14.

3.4.2 System Stress

Over time, robust parameters will change for differing acoustic scenes. This implies that somehow the scene *context* has changed somehow [10]. Tracking the parameter variations can give a qualitative measure about the degree of difficulty an observer will have in detecting an anomaly in any sub-band. We refer to this degree of difficulty as the *system stress*. On the assumption that a change in the current context implies that an observer system will have difficulty to adapt, we attempt to define and follow the signal context. For a buffer Δ_i of past contexts, there is a mean context $\vec{\Xi}_{\Delta_i}$ that is subject to some variation Θ_{Δ_i}. We then compute the system stress as:

$$\mathrm{S}_i(\vec{C}_i) = \sqrt{(\vec{C}_i - \vec{\Xi}_{\Delta_i})^{\mathrm{T}} \; \Theta_{\Delta_i}^{-1} \; (\vec{C}_i - \vec{\Xi}_{\Delta_i})} \tag{3.23}$$

Where $\vec{C}_i$ is the current context containing the robust means in units of their robust scatter for each of the N_s sub-bands that were obtained from wavelet filtering:

$$\vec{C}_i = \left[\frac{\tilde{\vec{\mu}}_{\delta_{1,i}}}{\tilde{\Sigma}_{\delta_{1,i}}} \; \frac{\tilde{\vec{\mu}}_{\delta_{2,i}}}{\tilde{\Sigma}_{\delta_{2,i}}} \cdots \frac{\tilde{\vec{\mu}}_{\delta_{N_s,i}}}{\tilde{\Sigma}_{\delta_{N_s,i}}} \right]^{\mathrm{T}} \tag{3.24}$$

$\tilde{\vec{\mu}}_k$ and $\tilde{\Sigma}_k$ are the robust means in units of the robust covariance respectively for each of the $k = 1 \dots N_s$ sub-bands. Also, we define $\vec{\Xi}_\Delta$ and Θ_Δ as the unbiased maximum-likelihood Gaussian parameter estimates for the past M observations of $\vec{C}_m$ in buffer Δ_i:

$$\Delta_i = \left\{ \vec{C}_i, \vec{C}_{i-1}, \vec{C}_{i-2}, \dots, \vec{C}_{i-M} \right\} \in \Re^{N_s \times M} \tag{3.25}$$

$$= \left\{ \begin{bmatrix} \frac{\tilde{\vec{\mu}}_{\delta_{1,i}}}{\tilde{\Sigma}_{\delta_{1,i}}} \\ \frac{\tilde{\vec{\mu}}_{\delta_{2,i}}}{\tilde{\Sigma}_{\delta_{2,i}}} \\ \vdots \\ \frac{\tilde{\vec{\mu}}_{\delta_{N_s,i}}}{\tilde{\Sigma}_{\delta_{N_s,i}}} \end{bmatrix} \begin{bmatrix} \frac{\tilde{\vec{\mu}}_{\delta_{1,i-1}}}{\tilde{\Sigma}_{\delta_{1,i-1}}} \\ \frac{\tilde{\vec{\mu}}_{\delta_{2,i-1}}}{\tilde{\Sigma}_{\delta_{2,i-1}}} \\ \vdots \\ \frac{\tilde{\vec{\mu}}_{\delta_{N_s,i-1}}}{\tilde{\Sigma}_{\delta_{N_s,i-1}}} \end{bmatrix} \begin{bmatrix} \frac{\tilde{\vec{\mu}}_{\delta_{1,i-2}}}{\tilde{\Sigma}_{\delta_{1,i-2}}} \\ \frac{\tilde{\vec{\mu}}_{\delta_{2,i-2}}}{\tilde{\Sigma}_{\delta_{2,i-2}}} \\ \vdots \\ \frac{\tilde{\vec{\mu}}_{\delta_{N_s,i-2}}}{\tilde{\Sigma}_{\delta_{N_s,i-2}}} \end{bmatrix} \cdots \begin{bmatrix} \frac{\tilde{\vec{\mu}}_{\delta_{1,i-M}}}{\tilde{\Sigma}_{\delta_{1,i-M}}} \\ \frac{\tilde{\vec{\mu}}_{\delta_{2,i-M}}}{\tilde{\Sigma}_{\delta_{2,i-M}}} \\ \vdots \\ \frac{\tilde{\vec{\mu}}_{\delta_{N_s,i-M}}}{\tilde{\Sigma}_{\delta_{N_s,i-M}}} \end{bmatrix} \right\} \tag{3.26}$$

$$\vec{\Xi}_{\Delta_i} = \mathrm{E}\left\{ \vec{\mathbf{C}} \right\} \tag{3.27}$$

$$\Theta^2_{\Delta_i} = \mathrm{E}\left\{ (\vec{\mathbf{C}} - \vec{\Xi}_{\Delta_i})^{\mathrm{T}} (\vec{\mathbf{C}} - \vec{\Xi}_{\Delta_i}) \right\} \tag{3.28}$$

$$\tag{3.29}$$

Since system stress $\mathrm{S}_i(\vec{C}_i)$ is a scalar quantity that is evaluated for each sample after a delay of at least M samples, its variations are can contain vector outliers that can be detected using appropriate thresholding. The decision criteria for determining whether a context change has occurred is therefore resolved by determining an acceptable threshold T_c for $S_i(\vec{C}_i)$.

3.4.3 Major Anomalies

Over the local scope defined by $\delta_{s,i}$, minor anomalies can reveal interesting features of an acoustic signal however, its importance as an uncharacteristic event over the scope defined by Δ_i is defined by the system stress [10]. If the system is sufficiently stressed we assert that the minor anomalies encountered in sub-bands are related to context change. We define a *major anomaly* as a minor anomaly that has occurred during a significant context change.

As with outlier detection of transformed Teager energy $\breve{\bar{\Psi}}_W$, a threshold can be determined to find outliers in Δ_i. If $\vec{C_i}$ are $p = N_s$-variate Gaussian distributed (as was assumed in the estimates for Ξ_{Δ_i} and Θ_{Δ_i}) then $\mathrm{S}_i(\vec{C_i})$ is Chi-distributed. Outliers in context are therefore defined as the following set:

$$\mathbf{\Omega}_i = \left\{ \exists \vec{C} \in \Delta_i \mid \mathrm{S}_i(\vec{C}) > T_c \right\} \tag{3.30}$$

Minor anomalies that occur while $\mathrm{S}_i(\vec{C_i}) > T_c$ are considered as *major anomalies*:

$$\mathbf{A}_{s,i} = \left\{ \exists \mathbf{a}_{s,i} \mid \mathrm{S}_i(\vec{C_i}) > T_c = \frac{1}{\chi_{N_s}}(1 - \beta) \right\} \tag{3.31}$$

Where β is the confidence we have that there will be no significant change in context. It should be noted that for any given sub-band s:

$$\mathbf{A}_{s,i} \subseteq \mathbf{a}_{s,i} \subseteq \delta_{s,i} \tag{3.32}$$

Only major anomalies from $\mathbf{A}_{s,i}$ are localized in space. If $\mathbf{A}_{s,i} = \{\emptyset\}$ then the localization operation is not performed in that sub-band.

3.5 Anomaly Localization

Spatial localization of a wide band source can be done in many ways and are generally optimized for performance. The goal of this work is not to suggest a new method of localization using sensor arrays, or to enhance performance in any way. This section describes a strategy for anomaly localization given samples that appear in time-frequency windows.

3.5.1 Localization in Sub-Bands

As denoted in their subscript, major anomalies are situated in both time and frequency. An arbitrary $\mathbf{A}_{s,i}$ contains samples from a single array sensor in a specified sub-band and time index. On the assumption that the anomaly is perceivable to all sensors in a given sensor array geometry, we can define a simple mapping $L_{s,i}$ that translates major anomalies in $\mathbf{A}_{s,i}$ to $P_{s,i}$ which is the set of physical locations for each sub-band at time index i:

$$L_{s,i} \, : \, \mathbf{A}_{s,i} \longmapsto P_{s,i} \in \Re^{N_s} \tag{3.33}$$

The localization method is arbitrary but a strategy is required in order to resolve the location of the anomalous source given major anomalies in the sub-bands. Since the system implicitly runs continuously in discrete time, a *bracketing* strategy is used to account for noisy positions. Recalling (3.32), it should be understood that the localization procedure requires only a subset of $\delta_{s,i}$. In the case of very short duration anomalies, the position estimates may be poor. Furthermore, $\mathbf{A}_{s,i}$ should not be expected to be identical across all s, especially for narrow band sources adding to the variability of position estimates across all bands.

3.5.2 Sources with Spatio-Spectral Properties

Non-point sources may have spatio-spectral properties that can cause position estimates that are very dissimilar to those in other bands. A singing choir may have such properties if the performers are physically placed by their singing tone. When the performers sing together, this wide band source can be easily localized. Spectral filtering before localization will reveal differing positions. Localization of a higher band will reveal the position of the soprano of the group, while the lower band will reveal the location of the tenor. Because of an unknown source geometry, it should not be assumed that the position distribution will be symmetric. On the assumption that non-point sources may have spatio-spectral properties, skew in the position distribution should be expected.

3.5.3 Resolving Position from Sub-Band Anomalies

The anomaly position P_i^* is given as the *median* of all major sub-band anomaly positions:

$$P_i^* = \min_{\theta} \mathrm{E}\{|P_{s,i} - \theta|\} \tag{3.34}$$

This measure of central tendency will make position more robust to sources with spatio-spectral properties while providing good estimates for position of point sources.

Chapter 4

Validation

The majority of the system described in this work was designed with validation data provided from two sources: a robotic data acquisition system and from audio recordings obtained from the public domain. The former allowed for a great degree of experimentation with localization of anomalies, while the latter allowed for a great degree with differing types of anomalies. Some discussion is provided here about each of these data sources with a concentration on the latter since most of this work is relevant to anomaly detection.

4.1 Algorithm Implementation

All processing and validation was done in the , the Matlab$^{\text{TM}}$v.7.0 computing environment under the Windows XP operating system with Service Pack 2 installed. The algorithms described in this work were programmed in native Matlab, object-oriented computing language. A custom driver interface object was designed to control the robot and acquire audio data from it. Audio recordings were imported into the Matlab environment using native functions where they were processed.

4.2 Live Data

The robotic instrumentation shown in Figure 4.1 was designed specifically for this work. It is a configurable stereo acoustic array and has the following technical characteristics:

Mechanical *Custom design. Chassis and support constructed in plastic and metal. Custom plastic gearing. Two servo motors, one for each degree of freedom.*

Connectors *Cylindrical power connector: 6.5V @ 2100mA, Serial interface: DB9, Stereo audio: TRS-3.5mm stereo. Computer: USB 2.0 via in-line converter.*

Acoustic Sensitivity *Standard condenser microphone: -67dB/µbar, -47dBV/Pascal ±4dB, 50Hz-20kHz.*

Degrees Of Freedom *Two degrees of freedom: Microphone spacing* ($2\text{cm} \leq d \leq 50\text{cm}$) *and azimuth* ($-90° \leq d \leq +90°$).

Components *Stereo acoustic amplifier: Velleman MK136. Servo motor controller: Pontech SV203. USB interface: Generic, in-line serial DB9 to USB2.0 converter.*

Control *USB serial port. Custom object-oriented software interface and driver in the Matlab™mathematical programming environment. See Appendix B.2.*

Microphones were placed on the tips of each of the bars which were connected to the top of the assembly. Rotation of a servo inside the boxed assembly it rests on caused a gear to rotate in one direction while an intermeshed gear rotated in the other direction. The ends of the bars opposite to the microphones were directly connected to these gears, so rotation of the gears caused an opening and closing of the bar assembly as shown in Figure 4.2. This first degree of freedom provided a controllable separation between the microphones. The entire microphone array was

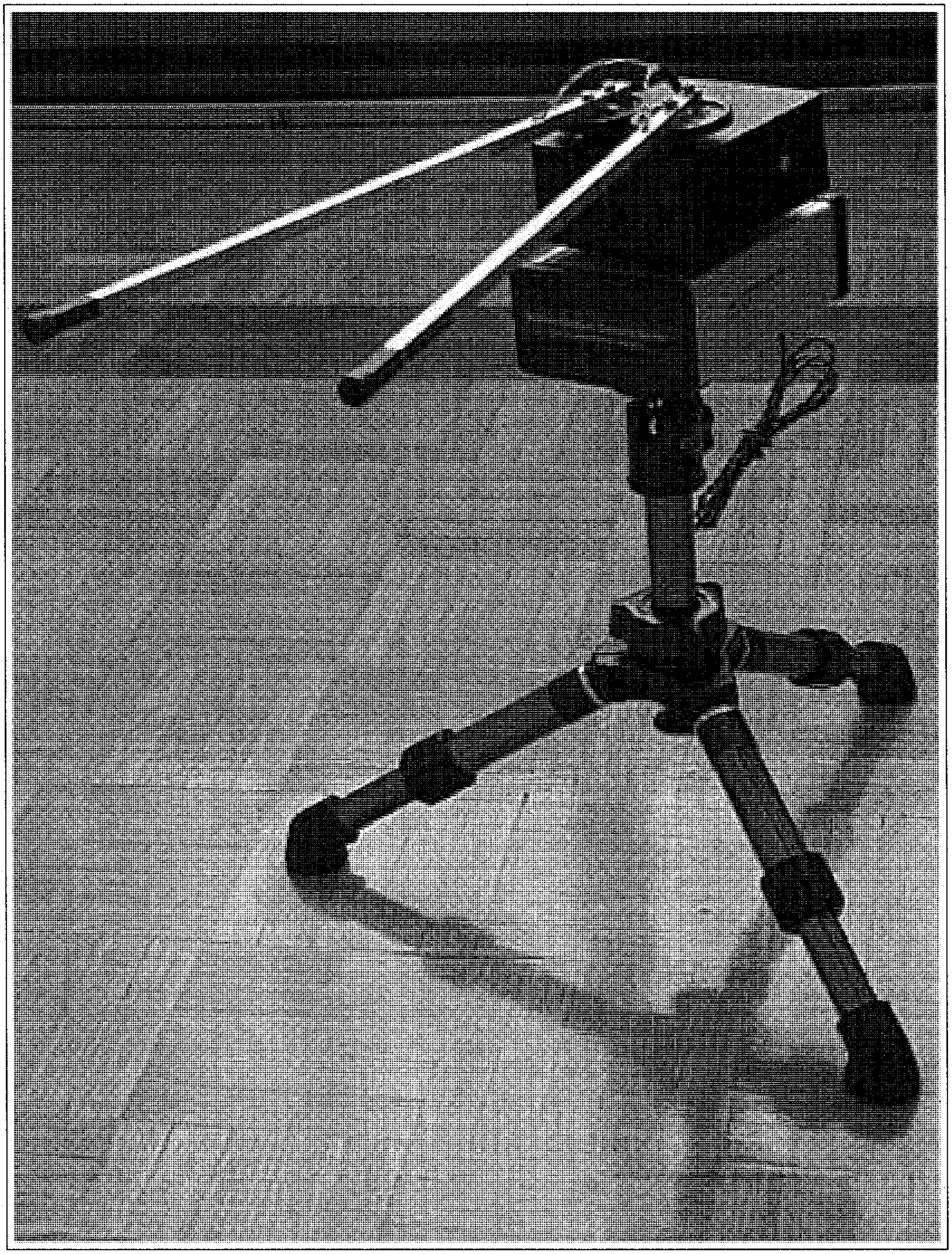


Figure 4.1: The robotic data acquisition device used for experimentation in this work.

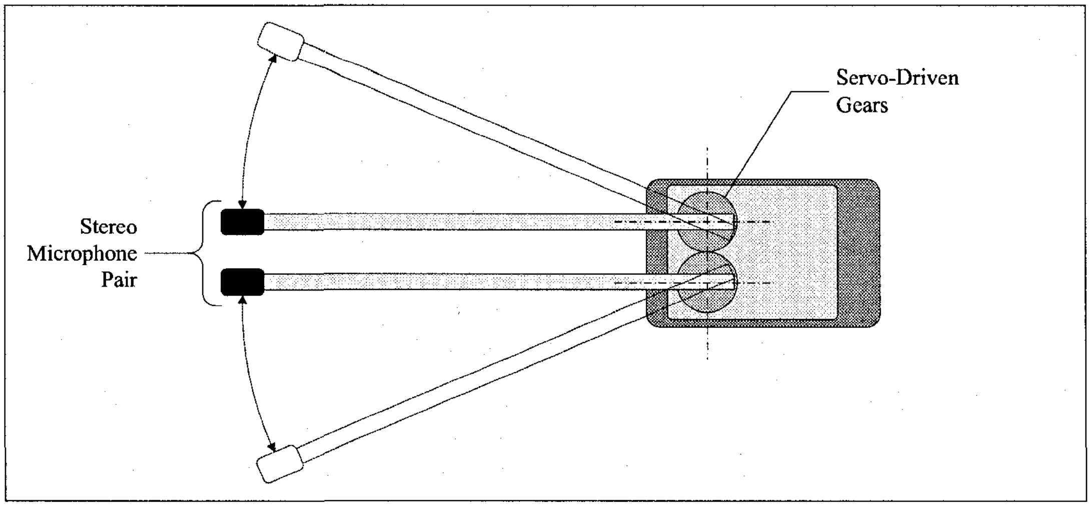


Figure 4.2: A top view of the robotic assembly. The bars are rotated with the turn of the servo-motor driven gears. The gears are intermeshed requiring torque on only one gear to rotate the other. This provides a controllable separation between the microphones.

also orientable in the horizontal plane. The array, with the separated microphones was attached to a stationary base by a direct servo-motor coupling. Rotation of the servo-motor, caused rotation of the array. The second degree of freedom was the array azimuth angle as shown in Figure 4.3. Both motors controlling each degree of freedom were interfaced to a single servo-motor controller with a serial interface (DB9). The controller was connected to a computer via an in-line serial port to USB port converter (a proprietary driver for the operating system was required). The condenser microphones were connected to a pre-amplifier with an adjustable gain. The pre-amplifier output was connected to the computer's standard sound interface allowing for automated audio sampling by the computer.

The robot was placed in acoustic scenes of many sorts. Some were very noisy with no particular subject of interest, or with complete silence. Some noisy environments had a brief, noticeable event occurring. Silent, and low noise environments were also chosen, all with events. Normally, differing types of events were introduced either accidentally, by intention, or by nature of the location. The near-immediate response of the system allowed for a great number of experiments to be performed.

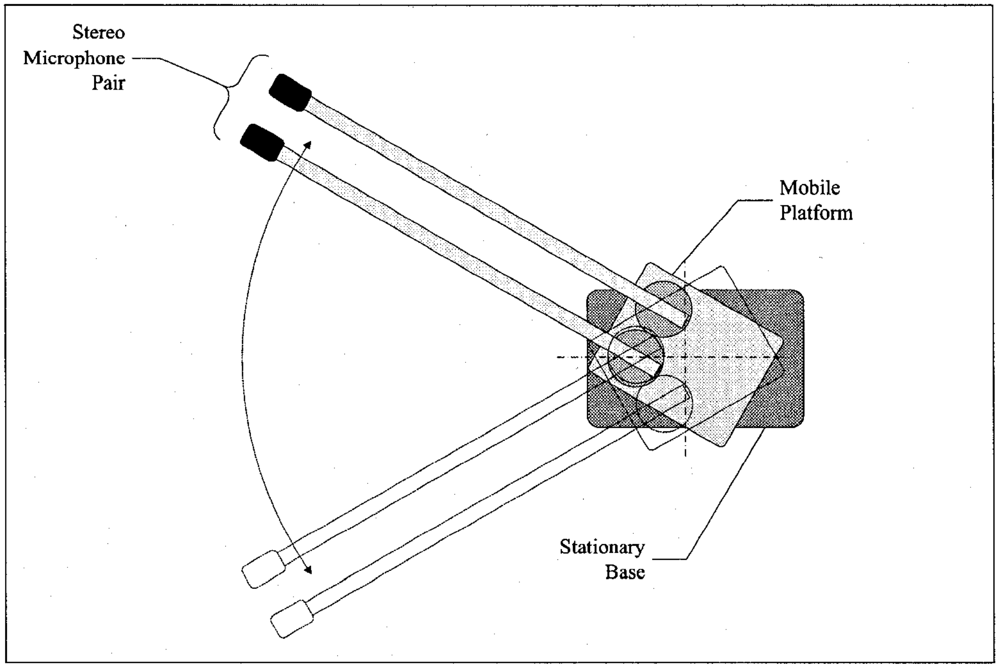


Figure 4.3: Another top view of the robotic assembly. The entire microphone array is subject to rotation in the horizontal plane from a servo-motor. This provides azimuth control of the microphone array.

When anomalies were detected, such as an object falling in an otherwise silent room, the best results were obtained from events that had low or mid tones. High tone anomalies were typically difficult to detect when the ambient noise also contained high tone sources. Low tone anomalies offered the best results.

Localization performance was good. When the robot detected a major anomaly, the unit would turn to face it. This offered a visual cue that was excellent in exploring the nature of anomalous events. Ultimately, the robot was an excellent experimental tool for this type of work, and working with live data confirmed that the system could work in many acoustic environments with good results. More will be discussed in Section 4.4.

4.3 Pre-Recorded Data

Public domain data is used here to demonstrate minor anomaly detection as well as system stress measurement. For each data sample, a brief description of the entire acoustic scene is given. Another description follows for a smaller sub-sample used in minor anomaly detection. For each test case, a brief discussion of results will follow first for minor anomaly detection and then signal stress measurement. After all data is presented, a summarizing discussion will follow highlighting key observations.

4.3.1 Data Processing

This work introduced many parameters, therefore due to the great variation in test data appropriate for this system, certain parameters are fixed so that comparisons can be easily made. For the experiments with *minor anomaly detection*, the parameters of Table 4.1 were used and in the case of system stress measurement, the parameters in Table 4.2 were used. For the results presented here, the context vector is taken to be $\vec{C_i} = \vec{0}$. The system stress described in Equation 3.23 can be considered as a normalized distance. The non-normalized stress is used for demonstration purposes as it helps to understand how the signal is changing. *Recall that for stress measurement, there is a buffering latency which is not shown in the plots.*

Parameter	**Value**	**Description**
N_s	8	Number of wavelet decomposed sub-bands.
W	100	Teager energy averaging window size.
η	100	Minimum anomaly length.
α	0.975	Data confidence level.
N	100000	Size of minor anomaly detection buffer.

Table 4.1: Parameters used for validating minor anomaly detection.

The data sets used for validation were selected primarily because of their acoustic qualities, specifically the occurrence of one or more events that a human observer would determine to be an anomaly. The data sets as well as the results of the selected

Parameter	Value	Description
N_s	5	Number of wavelet decomposed sub-bands.
W	100	Teager energy averaging window size.
η	100	Minimum anomaly length.
α	0.975	Normal data confidence level.
N	5000	Size of minor anomaly detection buffer.
M	500	Depth of the context buffer.

Table 4.2: Parameters used for measuring system stress.

experiments performed on them follow.

Three Phase Blower Motor

This is an audio sample of an initially inactive three phase blower motor which is activated for a brief period after which it is deactivated and allowed to slow down naturally (see Figure 4.4). As the motor starts up, there is a snapping sound followed by noise which sweeps from predominantly low frequencies, to predominantly high frequencies. By the end of the sample, the motor makes no sound, and there is no ambient noise. *752126 samples were acquired at a sampling rate of 44.1kHz, in 16 bits.*

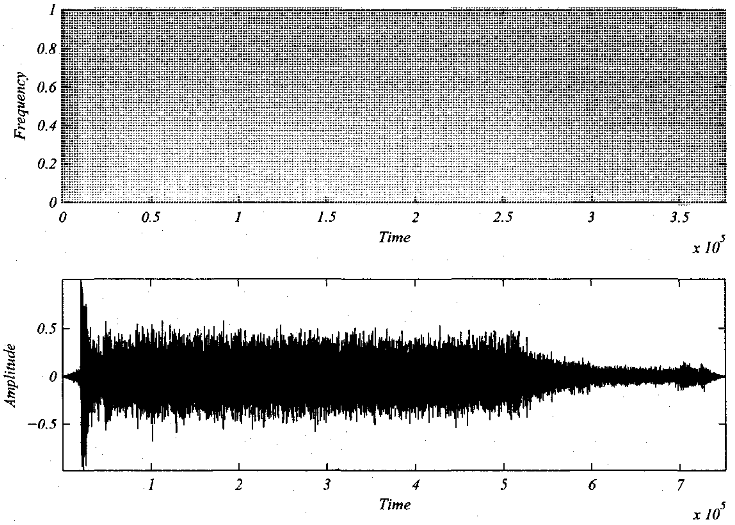


Figure 4.4: The spectrogram and amplitude plot for the entire 3 phase blower motor sample.

Signal stress measurement was performed on the entire 752126 sample signal. The system stress measure in Figure 4.5 highlights the changing sub-band trends in the signal with some latency due to buffering. Clearly split into three parts, the first hump is due to the startup of the motor from a silent period. After some stability is achieved, the stress levels drop, rising again as sounds from the motor change once again. When the motor is deactivated, the signal changes once again causing stress levels to rise.

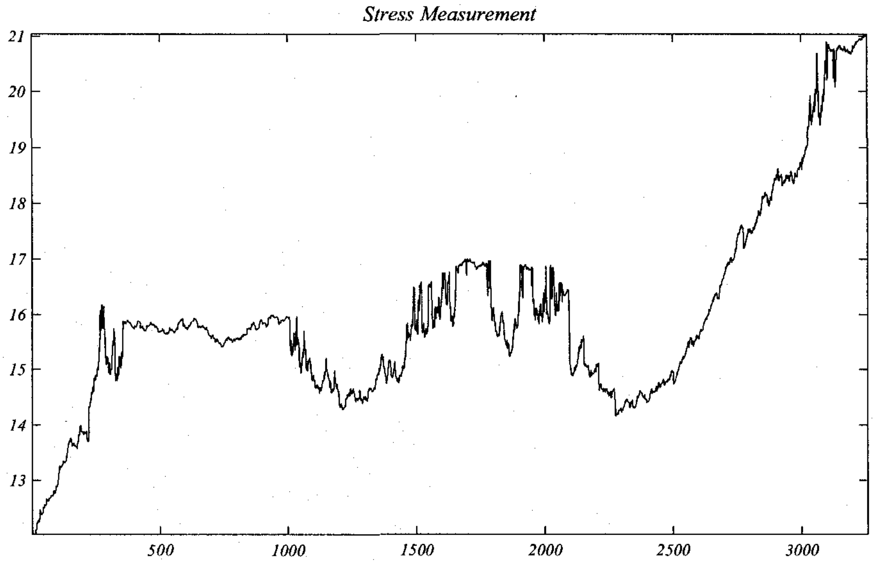
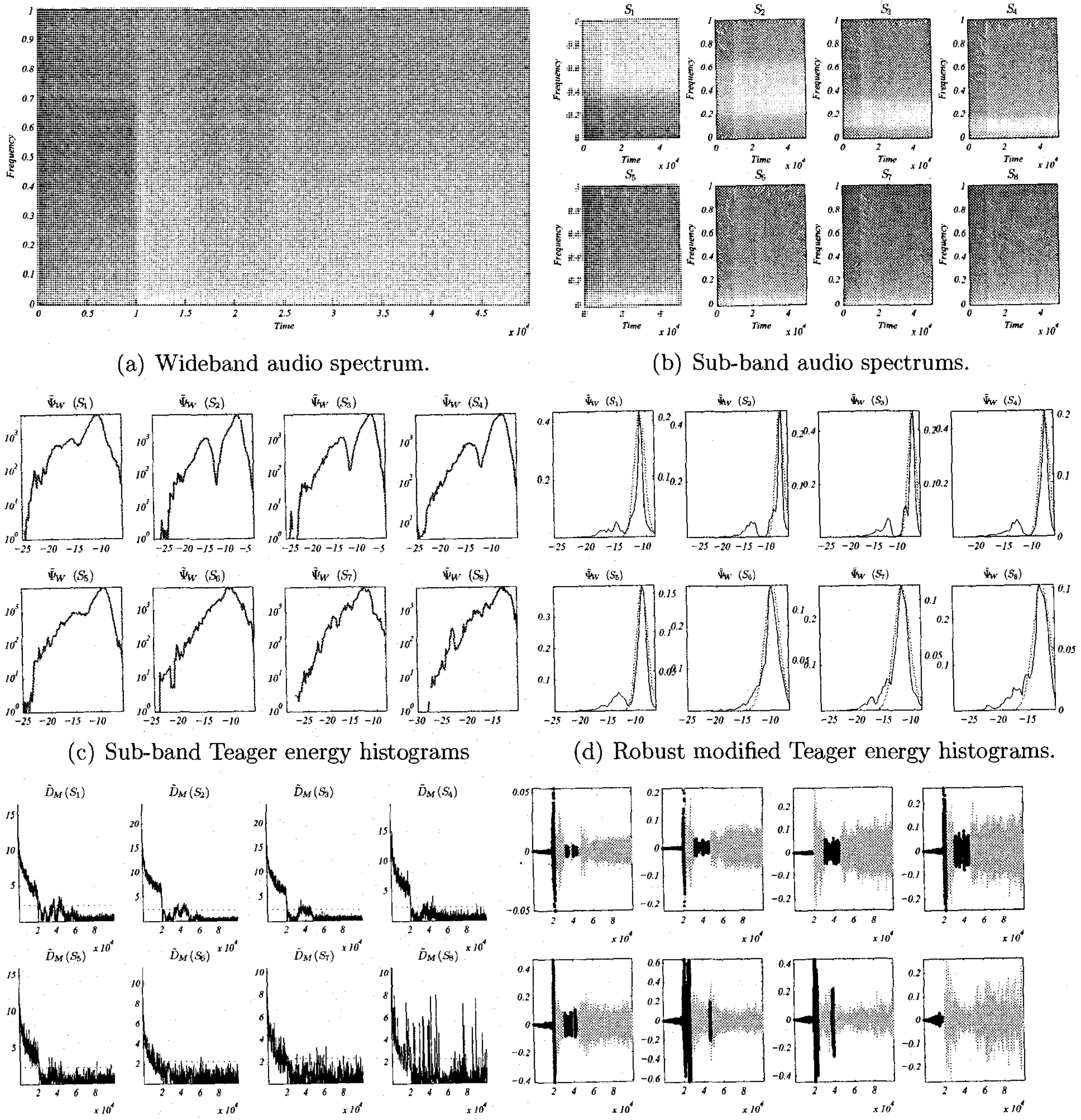


Figure 4.5: System Stress: 3 Phase motor blower.

Figure 4.6 shows the analysis results after minor anomaly detection applied to the first 100000 samples of the data. This 2.2676 second sub-sample, contains only the initial starting of the motor. Figure 4.6(d) shows the modified Teager energy histograms for each of 8 sub-bands, with the robustly estimated Gaussian distribution shown in gray. The first five sub-bands (S_{1-5}) in Figure 4.6(d) clearly show that there are a fair number of samples that lie quite far from the robust mean. The robust distances in Figure 4.6(e) confirm this and show that they occur at the start of the data. The dotted line reflects the confidence that 97.5% of the data will have robust Mahalanobis distances below this threshold. Figure 4.6(f) shows the waveforms of each sub-band with those samples detected as minor anomalies in black. They all show anomalies early on in the sample near the beginning, right where the motor is activated and where there is the snapping noise described above.



(a) Wideband audio spectrum.

(b) Sub-band audio spectrums.

(c) Sub-band Teager energy histograms

(d) Robust modified Teager energy histograms.

(e) Robust Mahalanobis distances.

(f) Minor anomaly detection

Figure 4.6: Minor Anomaly Detection: 3 phase blower motor, first 100000 samples.

Firecrackers With Human Screams

A high-pitched whistling sound from a firecracker followed by an explosion that makes a loud cracking noise as it explodes (see Figure 4.7). Following the explosion, there are children present who scream and laugh at the event. Before the children begin vocalizing, there is a small pause. There is no ambient noise, but some reverberation is heard from the explosion. *521212 samples were acquired at a sampling rate of 44.1kHz, in 16 bits.*

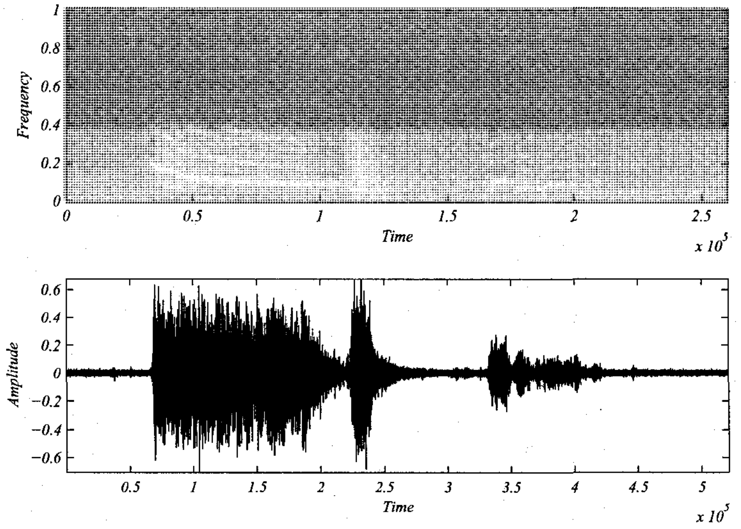


Figure 4.7: The spectrogram and amplitude plot for the entire sample with fireworks and human screams.

Signal stress measurement was performed on the entire 521212 sample signal. The system stress measure in Figure 4.8 highlights the changing sub-band trends over time. Initially, the signal consists of a high pitched whistle that remains constant as the firecracker ascends as can be seen in the plateau in the early part of the signal. After it explodes, a peak is observed. The brief silence afterward causes stress levels to decrease for its duration. When the nearby children are vocalizing, their spectral energies are *very* different from the firecracker causing the high stress levels at the end of the sample.

Figure 4.9 shows the analysis results after minor anomaly detection is applied to

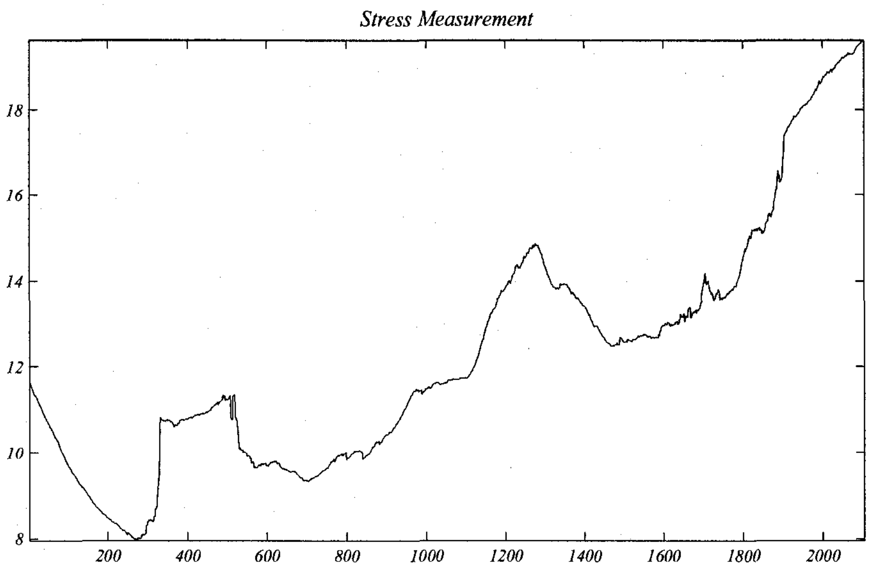
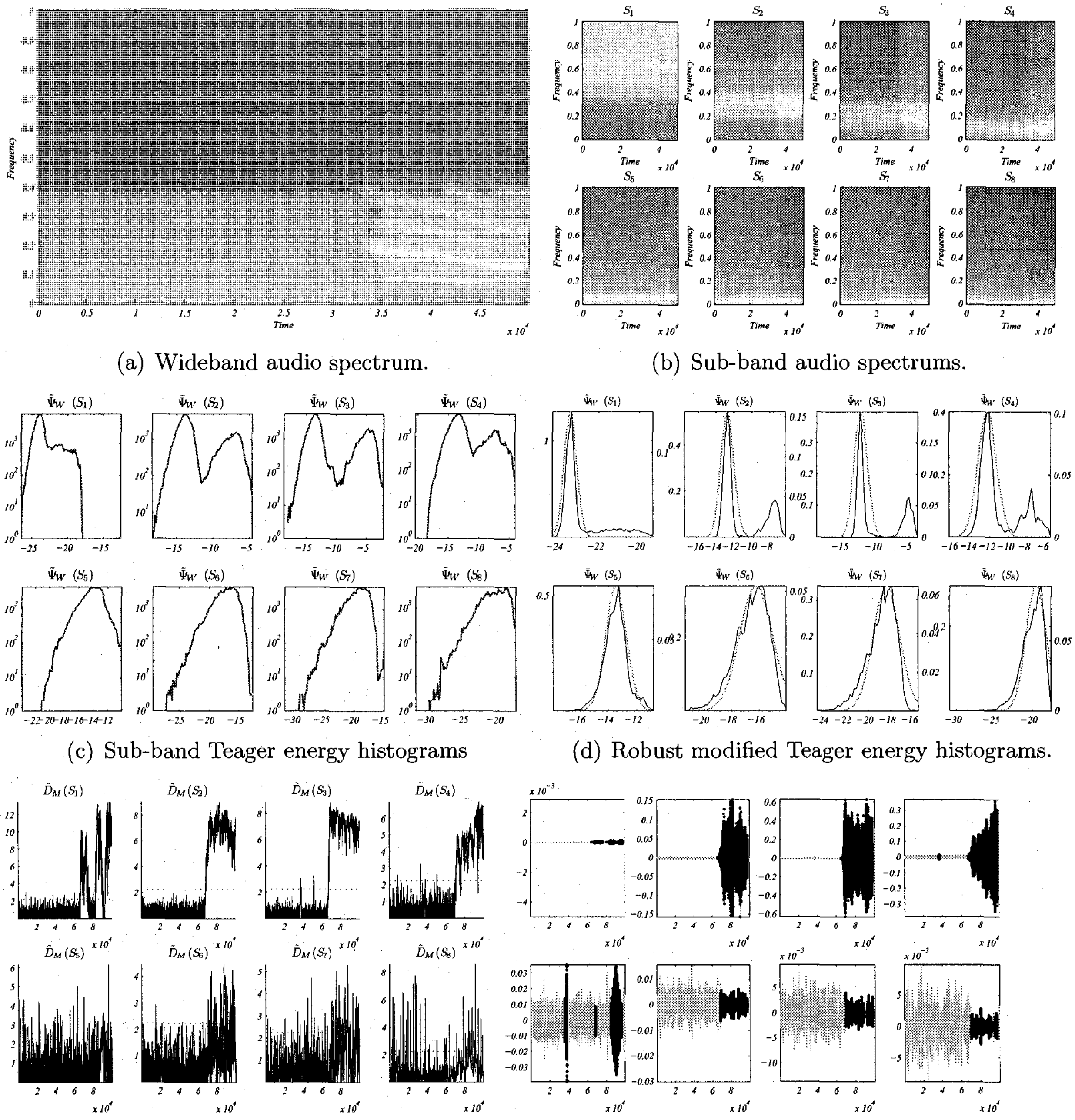


Figure 4.8: System Stress: firecrackers with human screams

the first 100000 samples of the data. This 2.2676 second sub-sample, contains a long silence followed by the sudden onset of the high-pitched whistling of the firecracker as it is being launched. Figure 4.9(a) shows the modified Teager energy histograms for each of 8 sub-bands, with the robustly estimated Gaussian distribution shown in gray. Figure 4.9(d) shows that in three sub-bands (S_{2-4}) there are a fair number of samples that lie quite far from the robust mean (there very little significant energy in the highest sub-band S_1). The robust distances in Figure 4.9(e) confirm this showing these higher than normal energies occur in the latter part of the signal which coincides with the point where the whistling firecracker suddenly appears in the acoustic scene. The dotted line reflects the confidence that 97.5% of the data will have robust Mahalanobis distances below this threshold. Figure 4.9(f) shows the waveforms of each sub-band with those samples detected as minor anomalies in black. All sub-bands have some energy anomalies, although seven out of the eight sub-bands have anomalies that coincide with onset of acoustic energy from the whistling firecracker.



(a) Wideband audio spectrum.

(b) Sub-band audio spectrums.

(c) Sub-band Teager energy histograms

(d) Robust modified Teager energy histograms.

(e) Robust Mahalanobis distances.

(f) Minor anomaly detection

Figure 4.9: Minor Anomaly Detection: Fire crackers with human screams, first 100000 samples.

Fireworks

In this sample, the scene is very quiet with the sounds from nearby birds. Predominantly consisting of very this very low intensity noise, the near silence is punctuated by two loud distant explosions for which some reverberation is heard (see Figure 4.10). *914379 samples were acquired at a sampling rate of 44.1kHz, in 16 bits.*

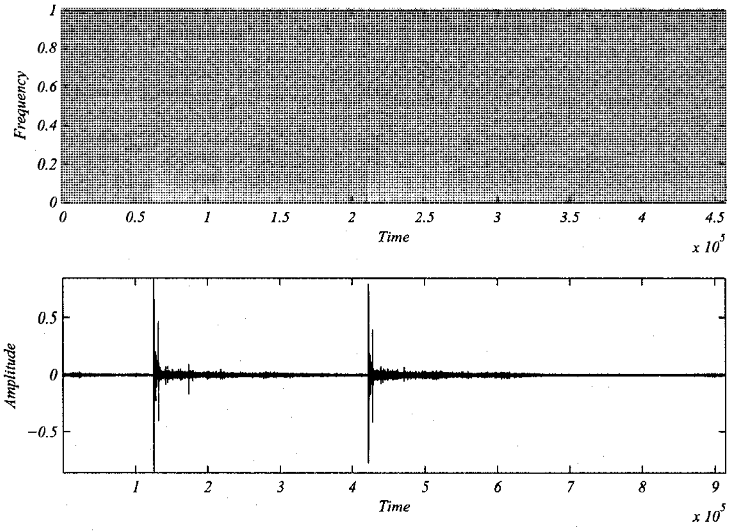


Figure 4.10: The spectrogram and amplitude plot for the fireworks sample.

Signal stress measurement was performed on the entire 914379 sample signal. The system stress measure in Figure 4.11 highlights the changing sub-band trends in the signal with some latency due to buffering. In this case, the scale of the stress measurement is important. There is very little variation over this sample indicating that the explosion events were not accounted for in this measurement. This was probably due to the short duration of the events. In other words, the short duration explosions did not contribute toward the sub-band trends.

Figure 4.12 shows the analysis results after minor anomaly detection applied to the first 100000 samples of the data. This 2.2676 second sub-sample, is of very low amplitude and is only of the birds without any explosions. In this acoustic sub-scene, many birds can be heard, although the bird call of one is quite noticeable above all others with a tweeting sound that repeats for a total of six times in this sample. Figure

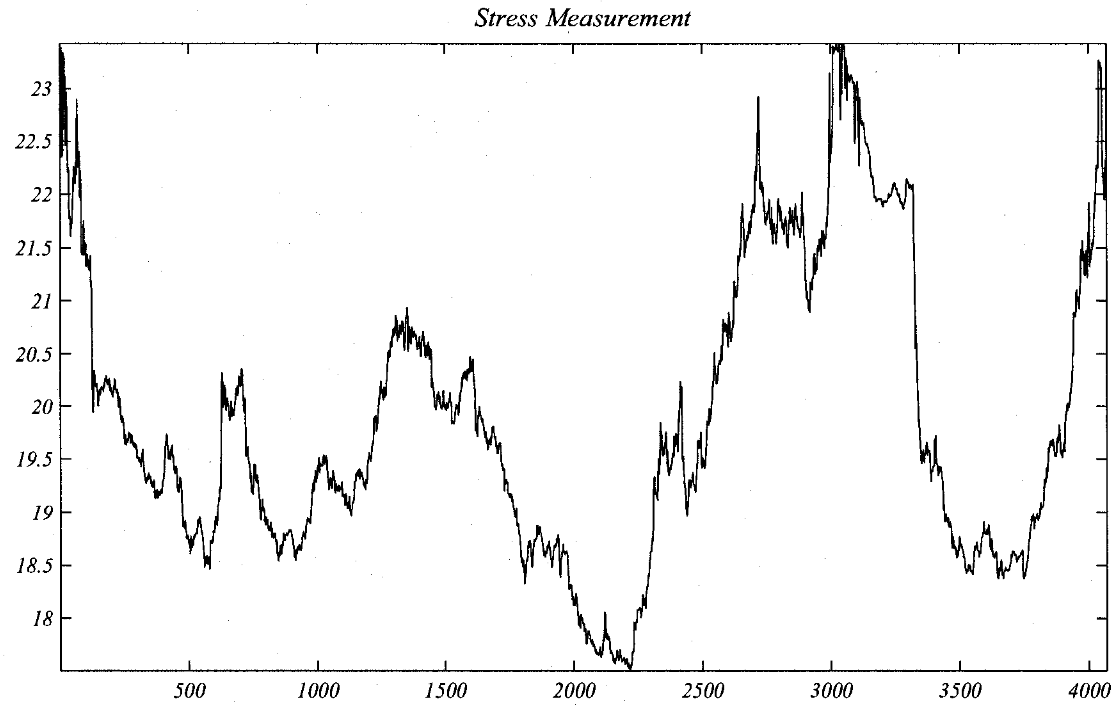
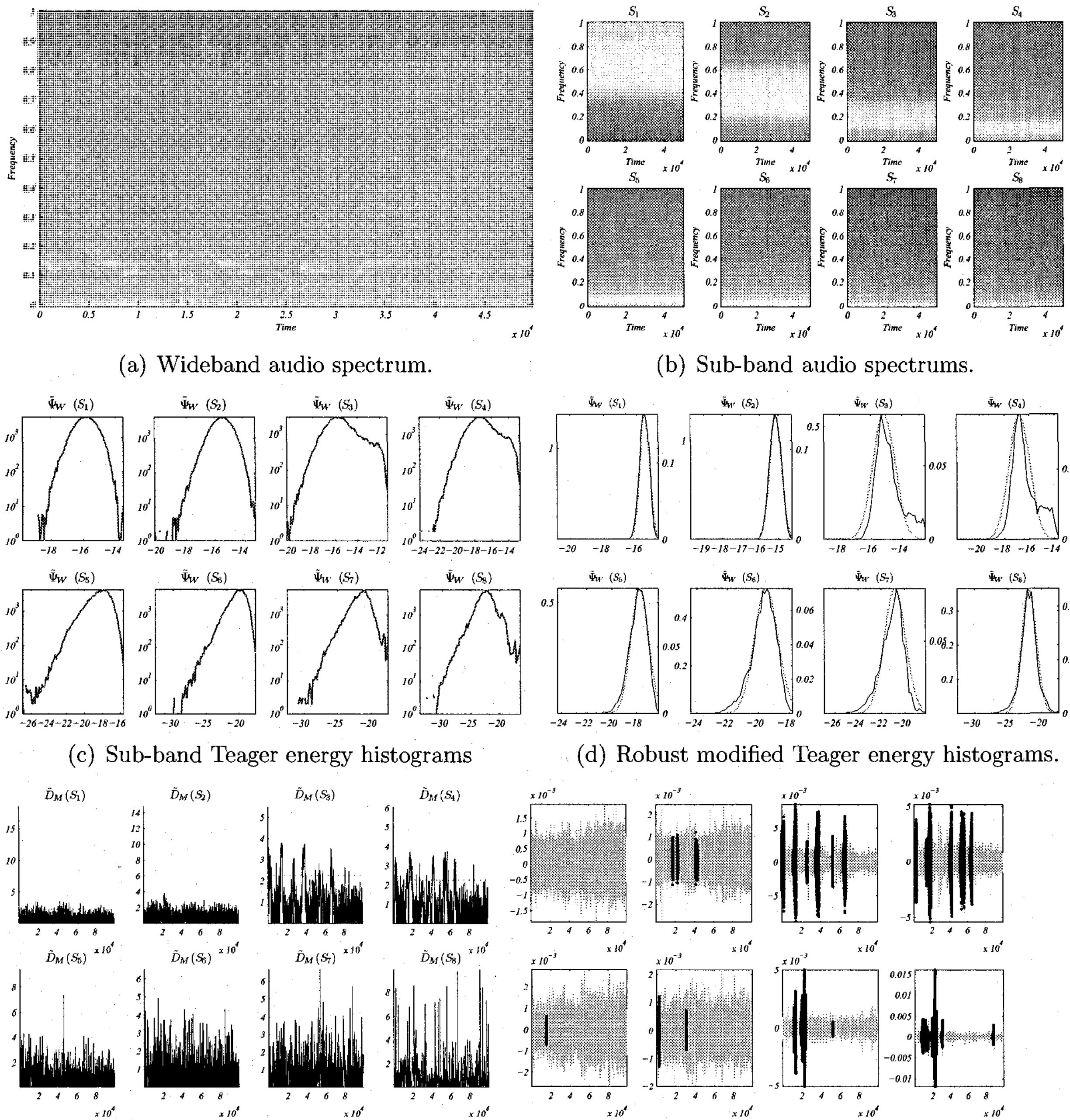


Figure 4.11: System Stress: fireworks sample.

4.12(a) shows the modified Teager energy histograms for each of 8 sub-bands, with the robustly estimated Gaussian distribution shown in gray. Note that these plots do not accurately highlight the presence of the relatively few numbers of samples with anomalous energies, although they are noticeable in the robust distances ($S_{3,4}$) in Figure 4.12(e). The dotted line reflects the confidence that 97.5% of the data will have robust Mahalanobis distances below this threshold. Figure 4.12(f) shows the waveforms of each sub-band with those samples detected as minor anomalies in black. The six tweeting sounds are clearly picked up in four sub-bands ($S_{3,4,7,8}$) coinciding with what appears in the acoustic scene. There are other anomalies detected as well although those sounds could not be accurately identified.



(a) Wideband audio spectrum.

(b) Sub-band audio spectrums.

(c) Sub-band Teager energy histograms

(d) Robust modified Teager energy histograms.

(e) Robust Mahalanobis distances.

(f) Minor anomaly detection

Figure 4.12: Minor Anomaly Detection: Fireworks data, first 100000 samples.

Fireworks at 500 Meters

From a distance of approximately 500 meters, an audio sample was obtained from a fireworks display. The acoustic scene opens from silence with a loud and sudden explosion followed by the sounds of debris particles falling to the ground (see Figure 4.13). Other fireworks that make a high frequency whistling sound when launched punctuate the acoustic scene and continue to the end of the sample with many concurrent deep sounding explosions. *720129 samples were acquired at a sampling rate of 44.1kHz, in 16 bits.*

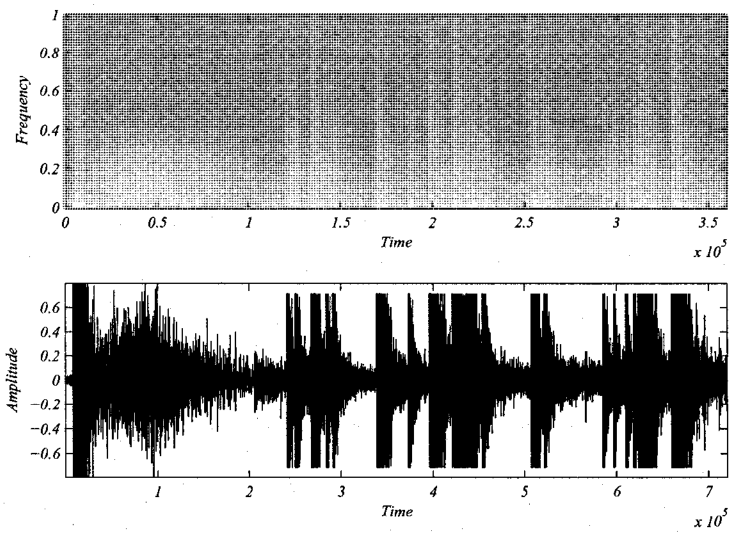


Figure 4.13: The spectrogram and amplitude plot for the fireworks at 500m sample.

Signal stress measurement was performed on the entire 720129 sample signal. The system stress measure in Figure 4.14 highlights the changing sub-band trends over time. The inital part of the signal contains a single deep explosion which raises the system stress decreasing only when debris fall gently. Concurrent explosions in rapid succession cause stress levels to decrease since they are short duration close to one another, demonstrating a trend. This causes the system stress to decrease. Near the end of the signal, a very loud and deep sounding explosion is heard with its reverberation, increasing stress levels as shown in the latter part of the plot.

Figure 4.15 shows the analysis results after minor anomaly detection is applied to

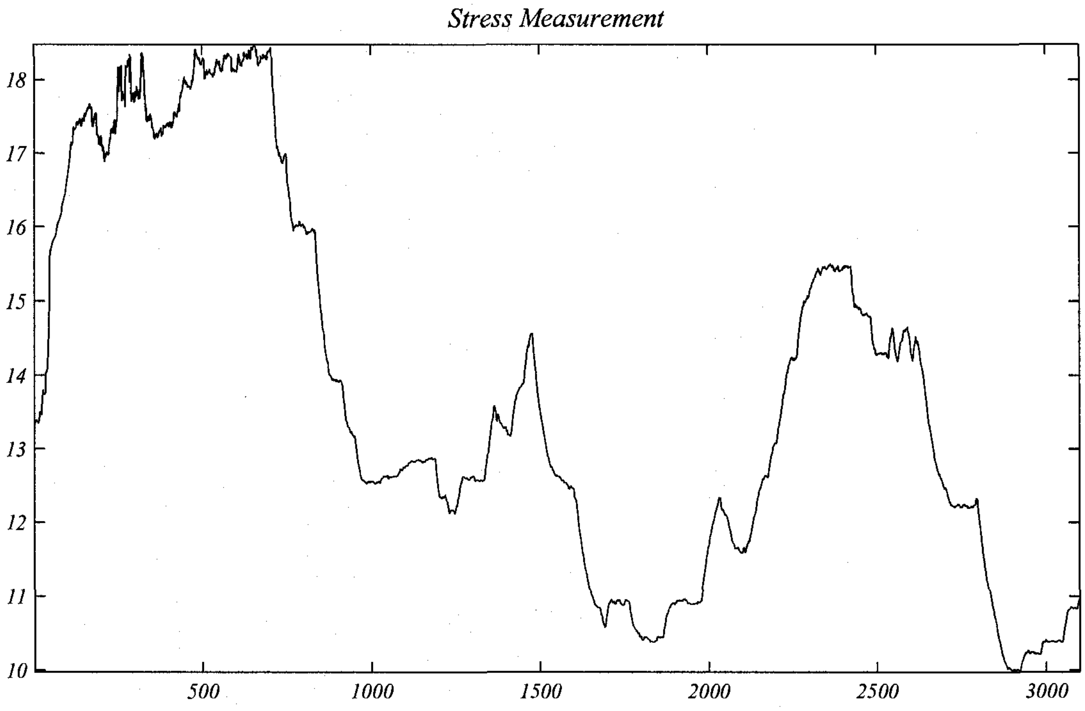
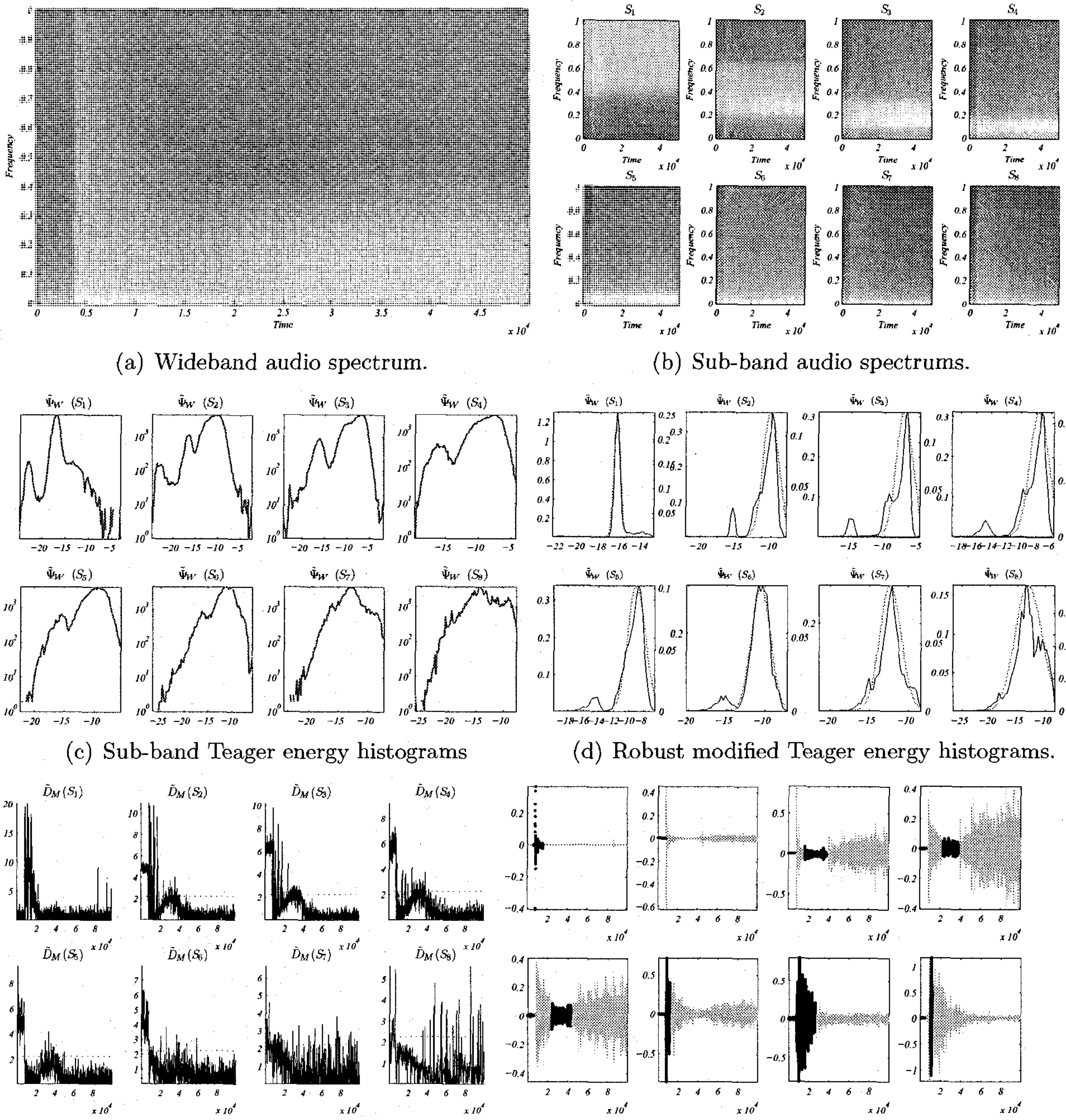


Figure 4.14: System Stress: Fireworks at 500 meters.

the first 100000 samples of the data. This 2.2676 second sub-sample, contains only the initial explosion followed by the sound of debris particles for the majority of its duration. Figure 4.15(d) shows the modified Teager energy histograms for each of 8 sub-bands, with the robustly estimated Gaussian distribution shown in gray. Some of the sub-bands clearly show that there are a fair number of samples that lie quite far from the robust mean (S_{3-6}). The robust distances in Figure 4.15(e) confirm this and show that they occur primarily at the start of the data when the explosion occurs and when the initial debris falls. The dotted line reflects the confidence that 97.5% of the data will have robust Mahalanobis distances below this threshold. Figure 4.15(f) shows the waveforms of each sub-band with those samples detected as minor anomalies in black. They all show anomalies early on in the sample where there is an explosion and the onset of falling debris.



(a) Wideband audio spectrum.

(b) Sub-band audio spectrums.

(c) Sub-band Teager energy histograms

(d) Robust modified Teager energy histograms.

(e) Robust Mahalanobis distances.

(f) Minor anomaly detection

Figure 4.15: Minor Anomaly Detection: Fireworks at 500m, first 100000 samples.

Gear-Reduced Motor Running at Low RPM

A gear-reduced electric motor is activated for the duration of this sample. From an intimal silence, the motor is activated producing a steadily rising tone which remains constant for the duration of the acoustic scene (see Figure 4.16). Near the end of the scene, the motor is deactivated producing a very quickly falling tone as it stops. *639450 samples were acquired at a sampling rate of 44.1kHz, in 16 bits.*

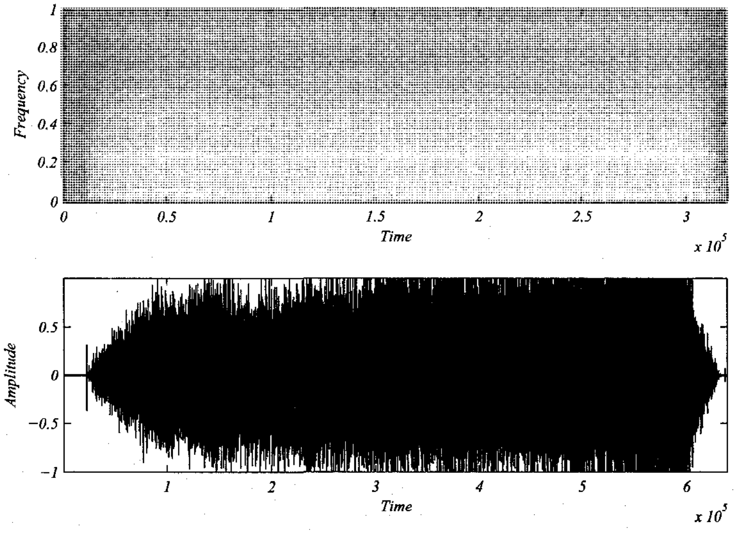


Figure 4.16: The spectrogram and amplitude plot for the entire gear-reduced motor sample.

Signal stress measurement was performed on the entire 639450 sample signal. The system stress measure in Figure 4.17 highlights the changing sub-band trends over time. The motor in this sample shows strong trends in each sub-band. The sound of the motor is very even and regular with no noticeable artifacts other than noise. The initial increase in system stress is due to the activation of the motor. The sustained stress is due to the sustained trends in sub-bands. Finally, the decrease in stress corresponds to the deactivation of the motor.

Figure 4.18 shows the analysis results after minor anomaly detection applied to the first 100000 samples of the data. This 2.2676 second sub-sample, contains only the initial explosion followed by the sound of debris particles for the majority of its

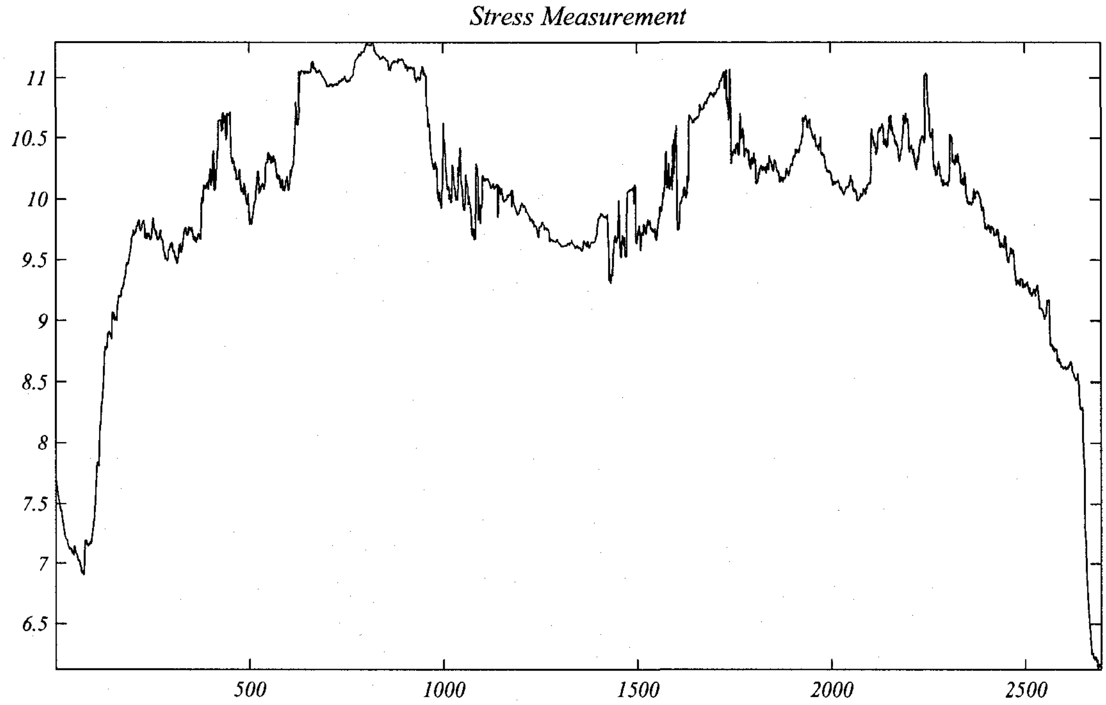
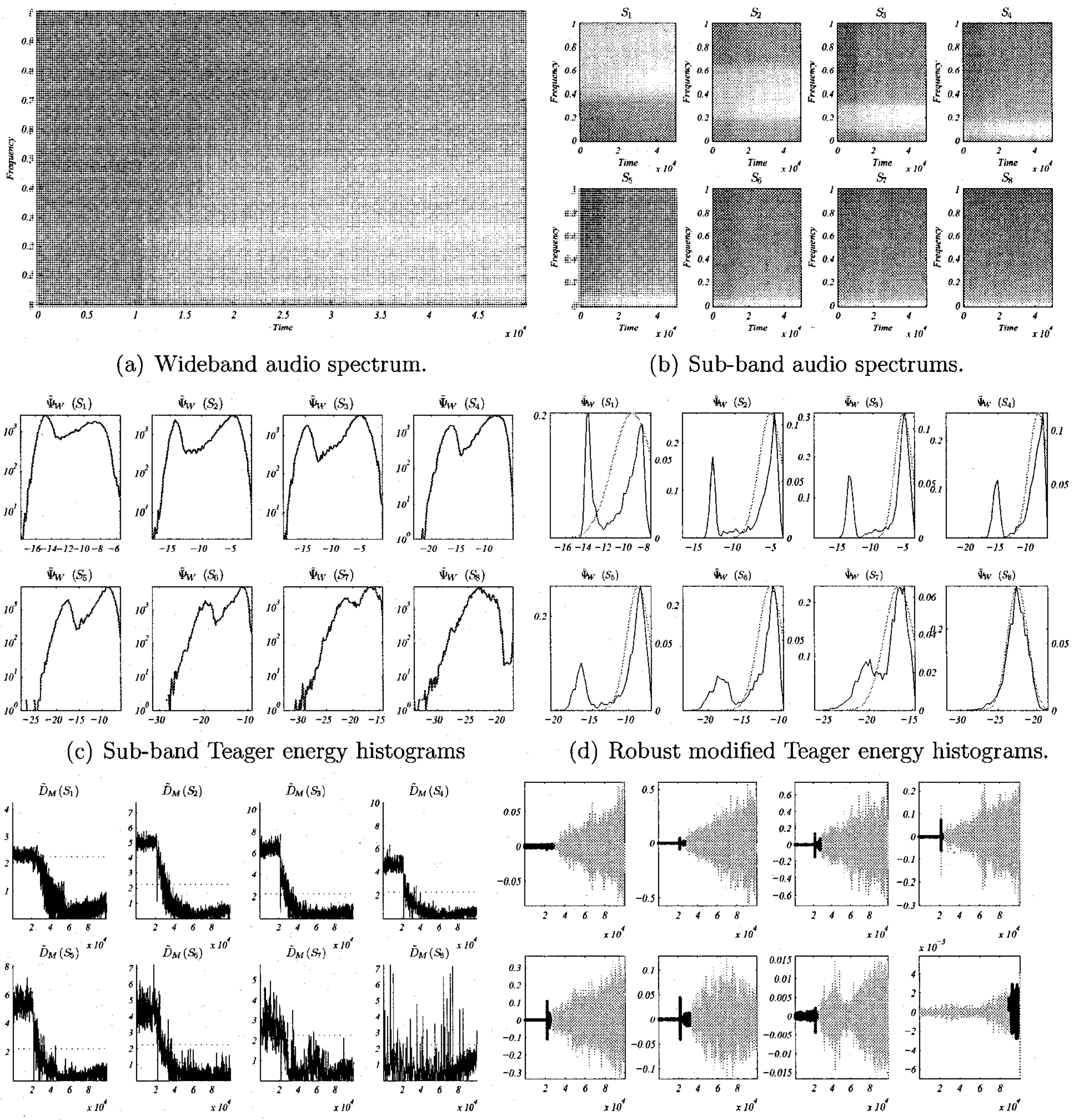


Figure 4.17: System Stress: Gear-reduced motor running at low RPM.

duration. Figure 4.18(d) shows the modified Teager energy histograms for each of 8 sub-bands, with the robustly estimated Gaussian distribution shown in gray. Almost all of the sub-bands clearly show that there are many samples that lie quite far from the robust mean (S_{1-7}). The robust distances in Figure 4.18(e) confirm this and show that they occur primarily at the start of the data when there is a silence right before the motor is activated. The dotted line reflects the confidence that 97.5% of the data will have robust Mahalanobis distances below this threshold. Figure 4.18(f) shows the waveforms of each sub-band with those samples detected as minor anomalies in black. They all show anomalies where there is an uncharacteristic silence in the data which predominantly has energy in many sub-bands. The lowest sub-band (S_8) contains an anomaly where the motor is about to reach constant speed.



(a) Wideband audio spectrum.

(b) Sub-band audio spectrums.

(c) Sub-band Teager energy histograms

(d) Robust modified Teager energy histograms.

(e) Robust Mahalanobis distances.

(f) Minor anomaly detection

Figure 4.18: Minor Anomaly Detection: Gear-reduced motor running at low RPM, first 100000 samples.

May 18, 1980 Mt. St. Helen's Eruption From 140 Miles Away

A unique sound sample of the May 18, 1980 eruption of Mount St. Helens, a volcano in Washington state of the United States of America (see Figure 4.19). The sample was obtained 140 miles away with standard audio equipment. The sounds from the volcano are not easily heard due to its low frequency, and may not be noticed by a listening observer. Upon careful examination, at least six extremely low frequency seismic events occur with great intensity and short duration although the great majority of the sample consists of low intensity ambient noise from a natural setting. *1916293 samples were acquired at a sampling rate of 44.1kHz, in 16 bits.*

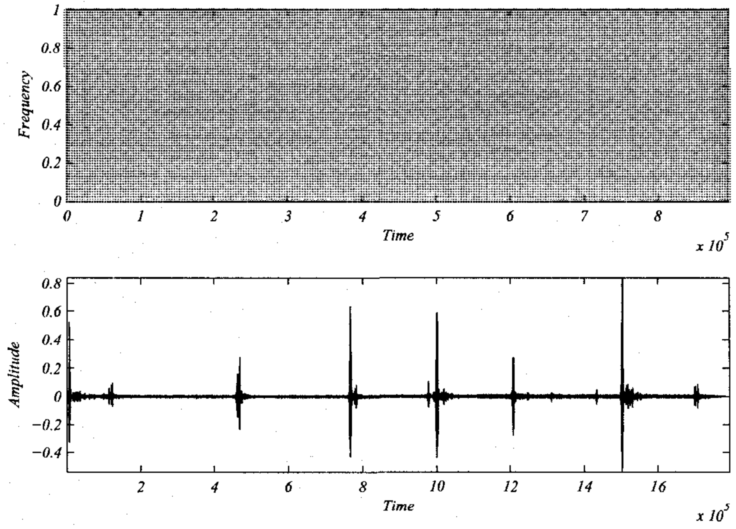


Figure 4.19: The spectrogram and amplitude plot for the Mount St. Helen's eruption.

Signal stress measurement was performed on the entire 1916293 sample signal. The system stress measure in Figure 4.20 highlights the changing sub-band trends over time. Each of the six very low frequency seismic events demonstrate trends that increase system stress each time. System stress lowers with the more regular sounds of ambient noise, in the absence of any seismic activity.

Figure 4.21 shows the analysis results after minor anomaly detection applied to the first 100000 samples of the data. This 2.2676 second sub-sample, contains a single seismic event which is barely audible amidst ambient natural sounds. Figure

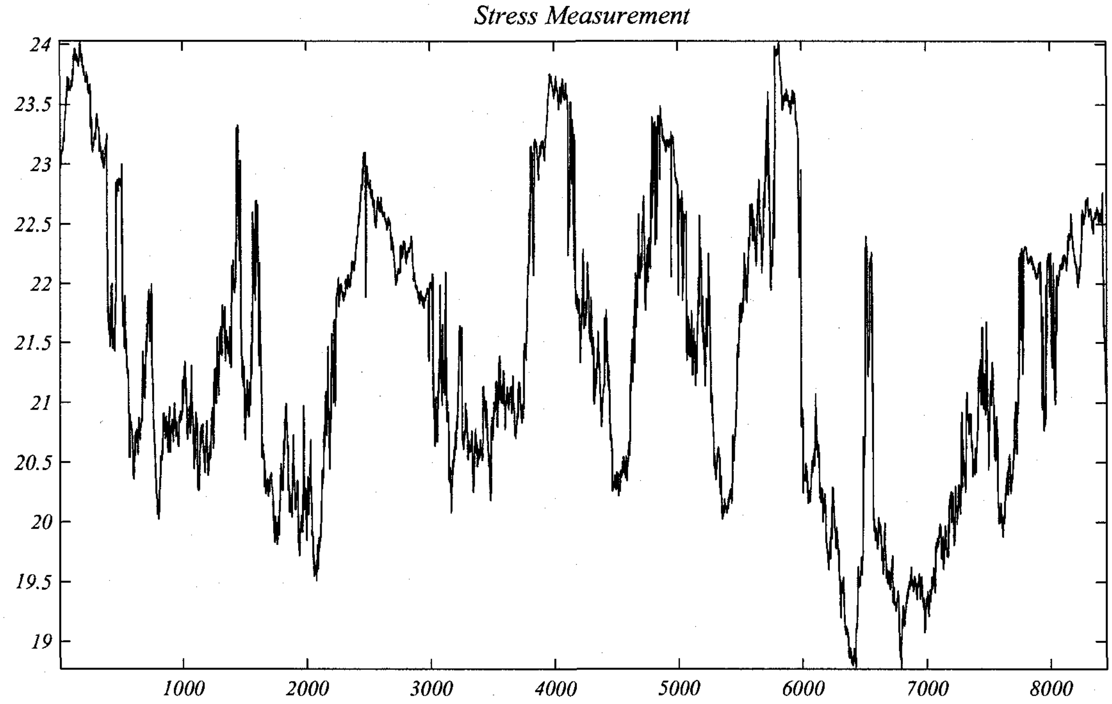
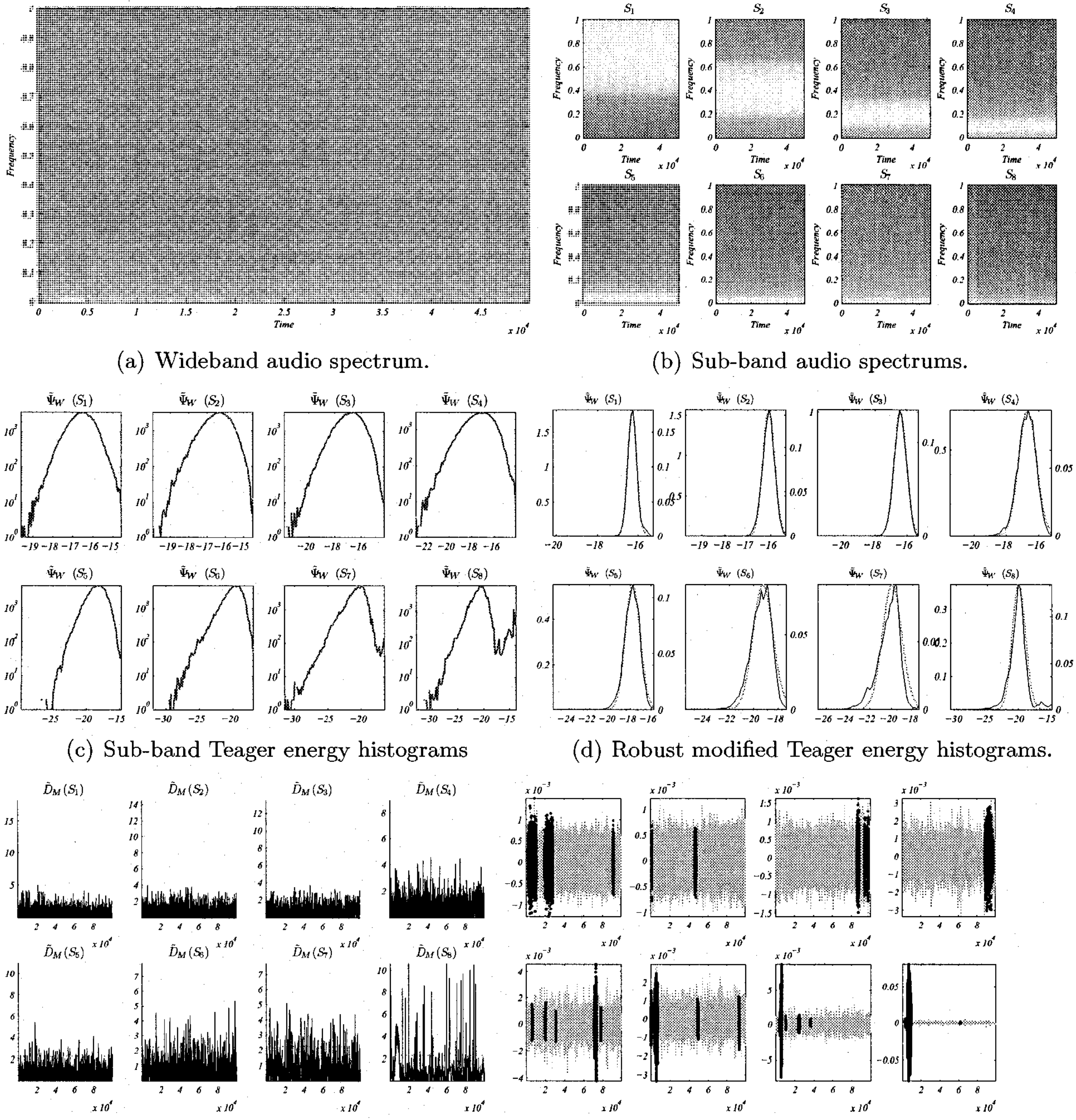


Figure 4.20: System Stress: Mount St. Helen's eruption data.

4.21(d) shows the modified Teager energy histograms for each of 8 sub-bands, with the robustly estimated Gaussian distribution shown in gray. There are some anomalies far from the mean although they are not accurately shown in Figure 4.21(d). The robust distances in Figure 4.21(e) confirm this and indicate that there are anomalies in several sub-bands at differing times. The dotted line reflects the confidence that 97.5% of the data will have robust Mahalanobis distances below this threshold. Figure 4.21(f) shows the waveforms of each sub-band. They all show anomalies however only a select few coincide with the seismic event (S_{6-8}). Initialy it could not be subjectively determined what the other anomalies were. Upon re-examination some sub-bands (S_{2-4}) were found to suddenly contain faintly audible natural sounds of birds and another unidentified source. The identified anomalies coincide with these events.



(a) Wideband audio spectrum.

(b) Sub-band audio spectrums.

(c) Sub-band Teager energy histograms

(d) Robust modified Teager energy histograms.

(e) Robust Mahalanobis distances.

(f) Minor anomaly detection

Figure 4.21: Minor Anomaly Detection: Mt. St. Helen's Eruption From 140 Miles Away (May 18, 1980), first 100000 samples.

Minolta Camera Attempting to Focus

This is an audio sample of a Minolta brand camera lens attempting to focus on a subject (see Figure 4.22). The lens motor makes a moderately high pitched tone when activated for a brief period and emits a mechanical clicking noise when the lens has reached its limit and stops moving. It is activated three times over the duration of the sample. There is no ambient noise, and there are clear silences between motor activations. *338843 samples were acquired at a sampling rate of 44.1kHz, in 16 bits.*

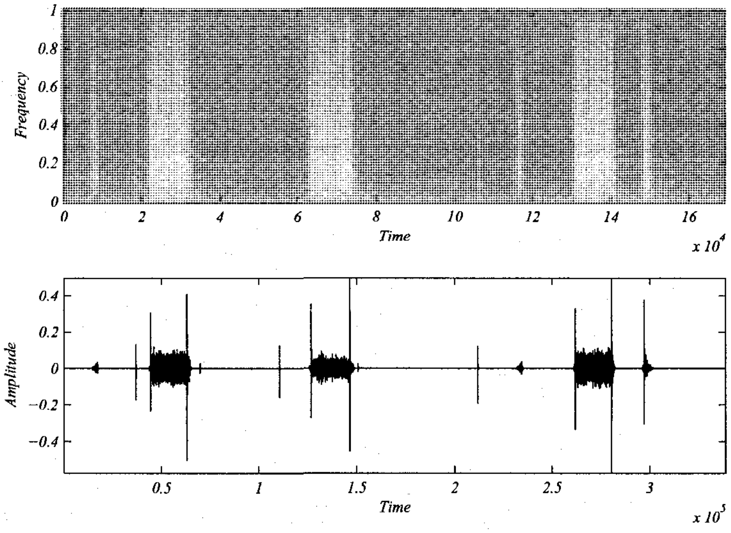


Figure 4.22: The spectrogram and amplitude plot for the Minolta camera attempting to focus.

Signal stress measurement was performed on the entire 338843 sample signal. The system stress measure in Figure 4.23 highlights the changing sub-band trends over time. In this signal, the sub-band trends are going in differing directions over the brief period of the first two consecutive events (camera motor in operation). Because of the pause before the third event, the system stress detects a strong change by the time the motor is activated in the third event, thereby increasing system stress.

Figure 4.24 shows the analysis results after minor anomaly detection applied to the first 100000 samples of the data. This 2.2676 second sub-sample, contains one focus attempt by the camera. Figure 4.24(d) shows the modified Teager energy histograms

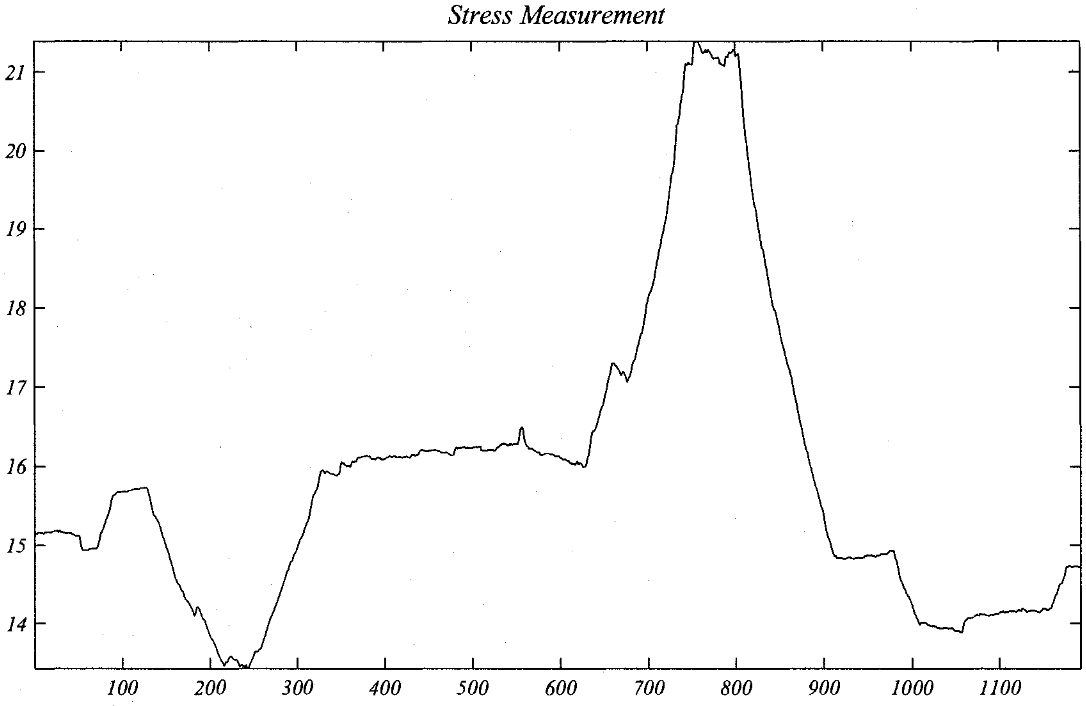
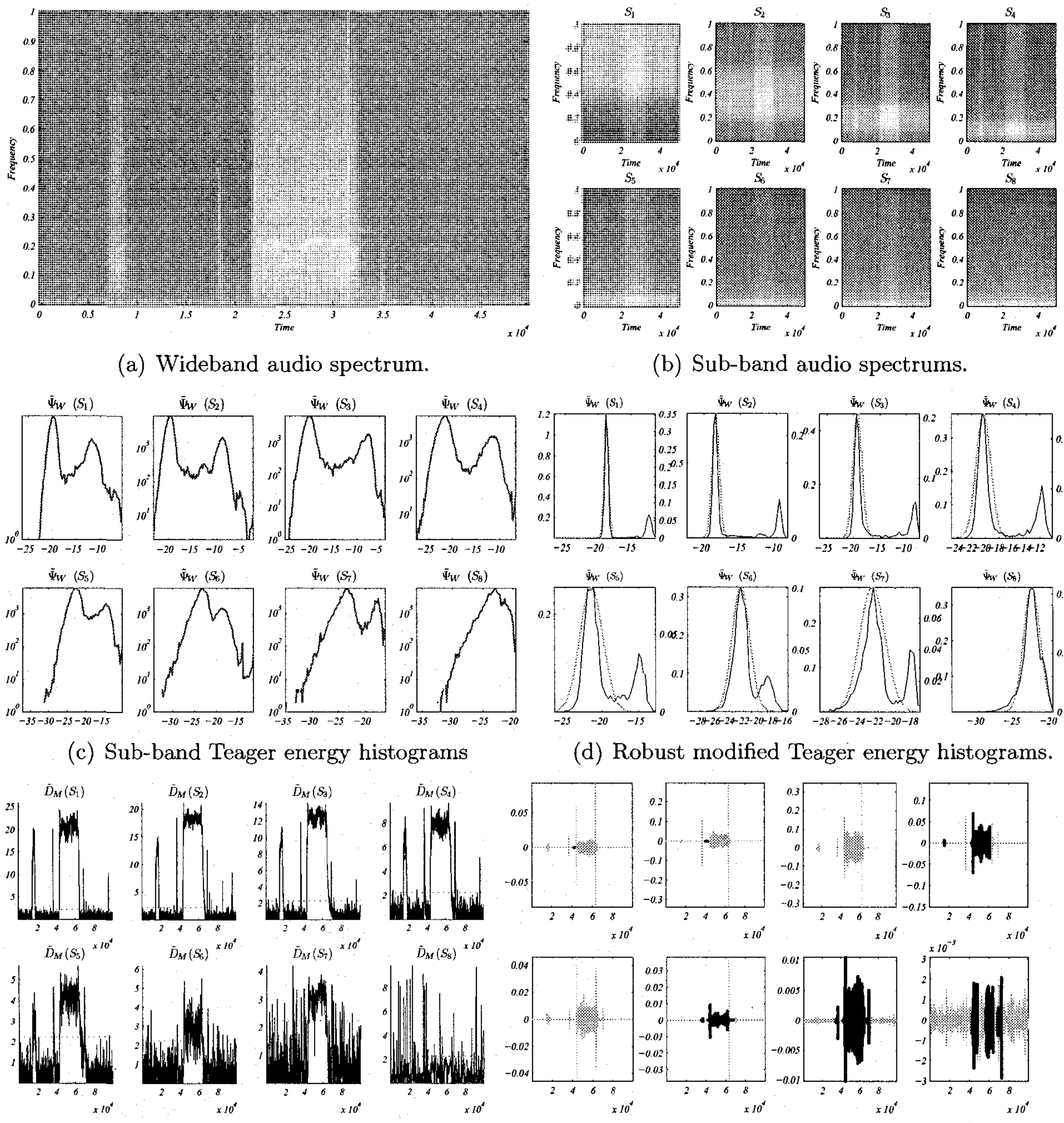


Figure 4.23: System Stress: Minolta camera focusing data.

for each of 8 sub-bands, with the robustly estimated Gaussian distribution shown in gray. It can be clearly seen that the first seven sub-bands (S_{1-7}) have significant energies far from the robust mean. The robust distances in Figure 4.24(e) confirm this and show that they occur midway through the sub-sample, coinciding with the motor activation which breaks a predominance of silence. The dotted line reflects the confidence that 97.5% of the data will have robust Mahalanobis distances below this threshold. Figure 4.24(f) shows the waveforms of each sub-band with those samples detected as minor anomalies in black. The anomaly detected coincides with the activation of the motor, breaking the silence of the acoustic scene.



(a) Wideband audio spectrum.

(b) Sub-band audio spectrums.

(c) Sub-band Teager energy histograms

(d) Robust modified Teager energy histograms.

(e) Robust Mahalanobis distances.

(f) Minor anomaly detection

Figure 4.24: Minor Anomaly Detection: Minolta camera attempting to focus, first 100000 samples.

Old Mechanical Toy

This is an audio sample of a mechanical toy. In operation, the toy's emits a clicking noise a quick pace while there is another knocking noise that occurs regularly with the clicking, a result of movement on a hard surface while in operation (see Figure 4.25). The clicking sound is very regular while the knocking sound has a longer period and is not as regular. *178080 samples were acquired at a sampling rate of 44.1kHz, in 16 bits.*

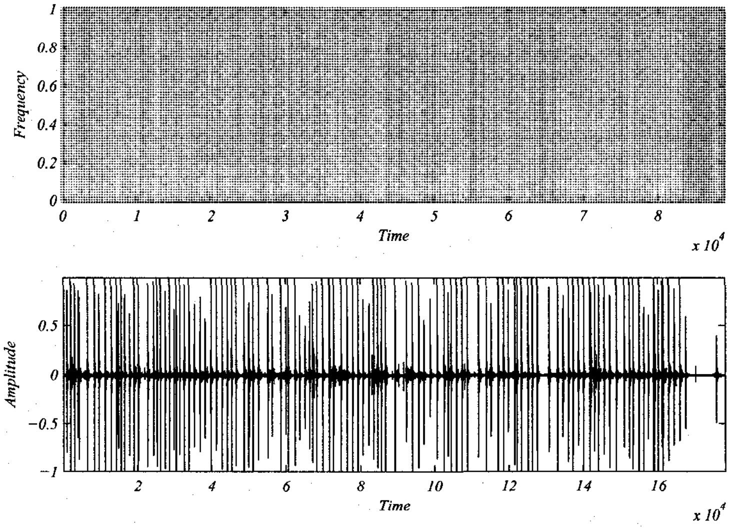


Figure 4.25: The spectrogram and amplitude plot for the old mechanical toy data.

Signal stress measurement was performed on the entire 178080 sample signal. The system stress measure in Figure 4.26 highlights the changing sub-band trends over time. In this sample, the clicking of the toy is very regular showing strong trends in each sub-band. The scale of plot shows that the system stress does not change because while there is signal variation, the strong trend is maintained.

Figure 4.27 shows the analysis results after minor anomaly detection applied to the first 100000 samples of the data. This 2.2676 second sub-sample, contains clicking from the toy as well as the knocking sounds. Figure 4.27(d) shows the modified Teager energy histograms for each of 8 sub-bands, with the robustly estimated Gaussian distribution shown in gray. Some of the sub-bands clearly show that there are samples

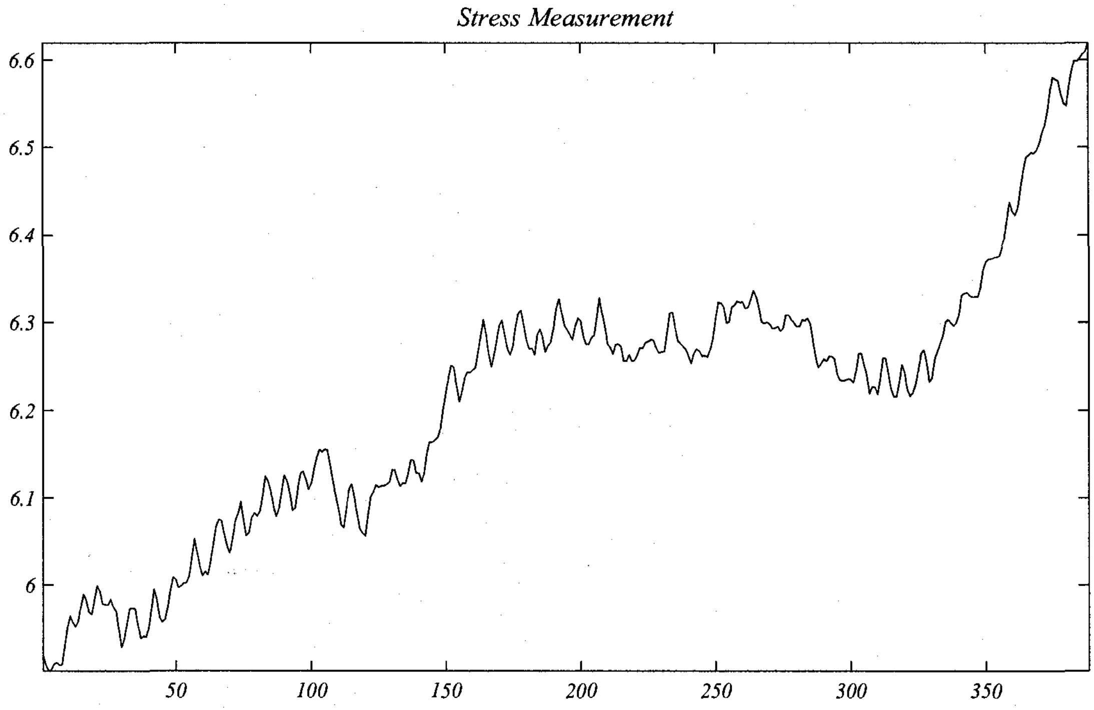
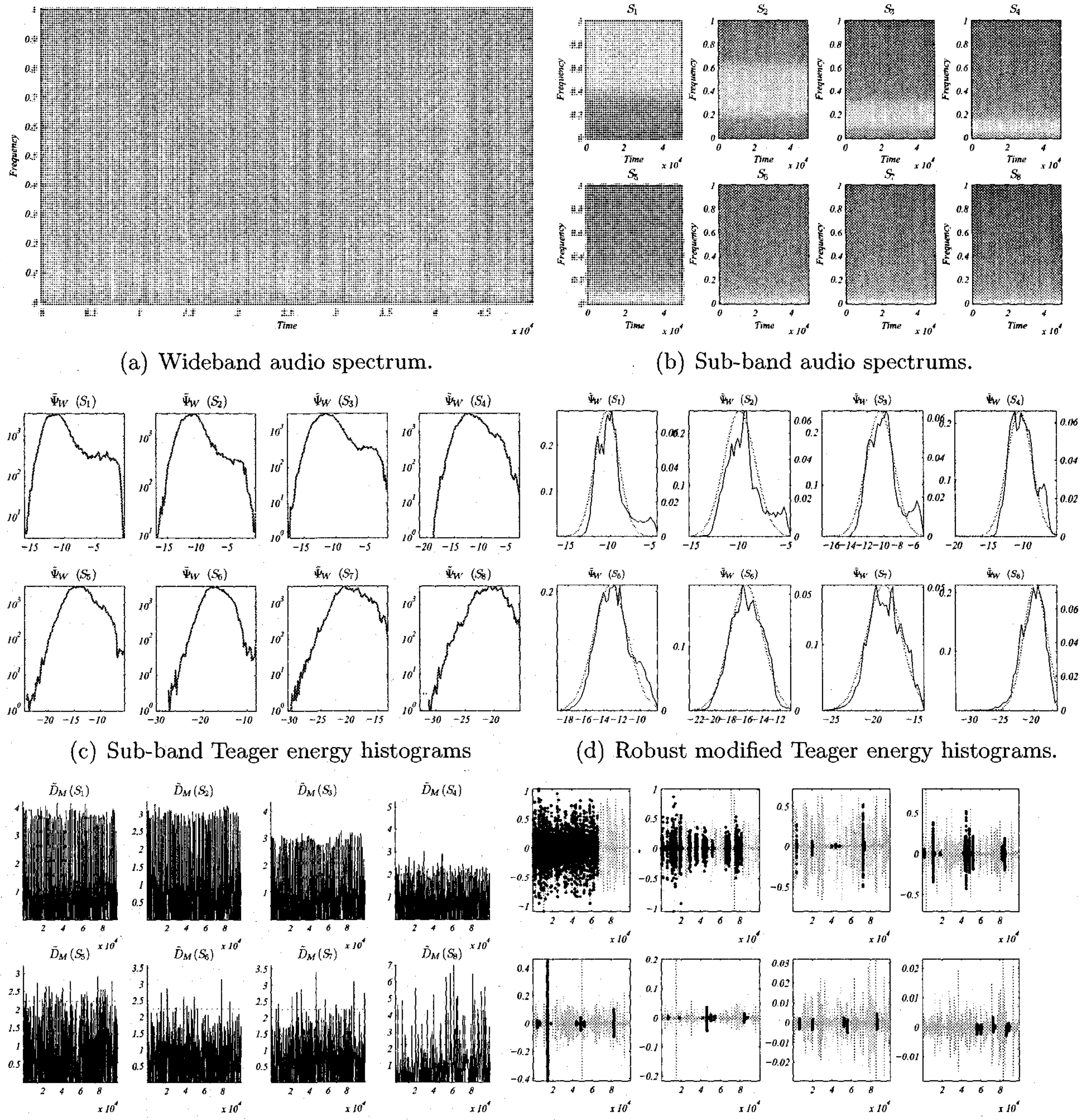


Figure 4.26: System Stress: Old mechanical toy.

that are quite far from the robust mean (S_{1-4}). The dotted line in Figure 4.27(e) reflects the confidence that 97.5% of the data will have robust Mahalanobis distances below this threshold. Figure 4.27(f) shows the waveforms of each sub-band with those samples detected as minor anomalies in black. In this case, the knocking sound coincides with some of the detected anomalies, however the anomalies in S_1 cannot be accounted for with a subjective label.



(a) Wideband audio spectrum.

(b) Sub-band audio spectrums.

(c) Sub-band Teager energy histograms

(d) Robust modified Teager energy histograms.

(e) Robust Mahalanobis distances.

(f) Minor anomaly detection

Figure 4.27: Minor Anomaly Detection: Old mechanical toy, first 100000 samples.

Rocks Hitting Each Other Very Hard

This audio sample is of two rocks hitting each other very hard followed by a long silence (see Figure 4.28). The short-duration striking sound is a high pitched cracking sound with some reverberation. *220500 samples were acquired at a sampling rate of 44.1kHz, in 16 bits.*

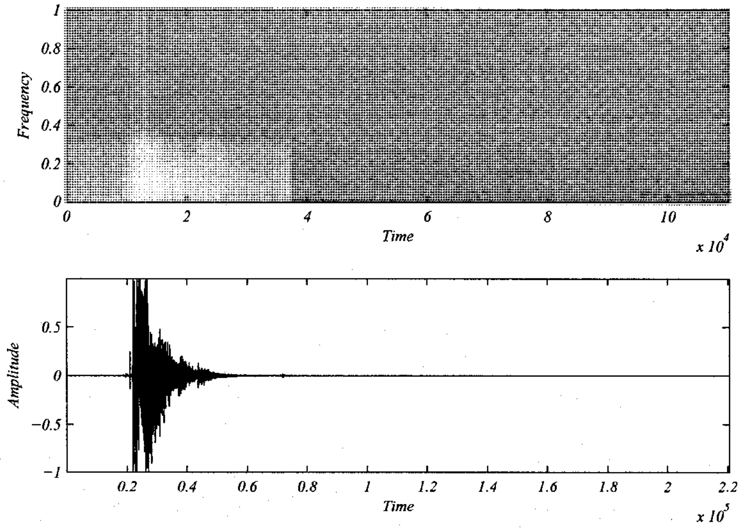


Figure 4.28: The spectrogram and amplitude plot for the rocks hitting each other.

Signal stress measurement was performed on the entire 220500 sample signal. The system stress measure in Figure 4.29 highlights the changing sub-band trends over time. This signal shows low system stress when the rocks hit each other and then high stress afterward when there is silence. This odd occurrence is due to the strong trends shown in all sub-bands when the rocks hit each other (recall the buffering latency) followed by changes as the signal tends toward silence in all sub-bands.

Figure 4.30 shows the analysis results after minor anomaly detection is applied to the first 100000 samples of the data. This 2.2676 second sub-sample, contains the initial striking sound followed by a brief silence (unlike the complete sample which has a longer silence). Figure 4.30(d) shows the modified Teager energy histograms for each of 8 sub-bands, with the robustly estimated Gaussian distribution shown in gray. Some of the sub-bands clearly show that there are a fair number of samples that lie

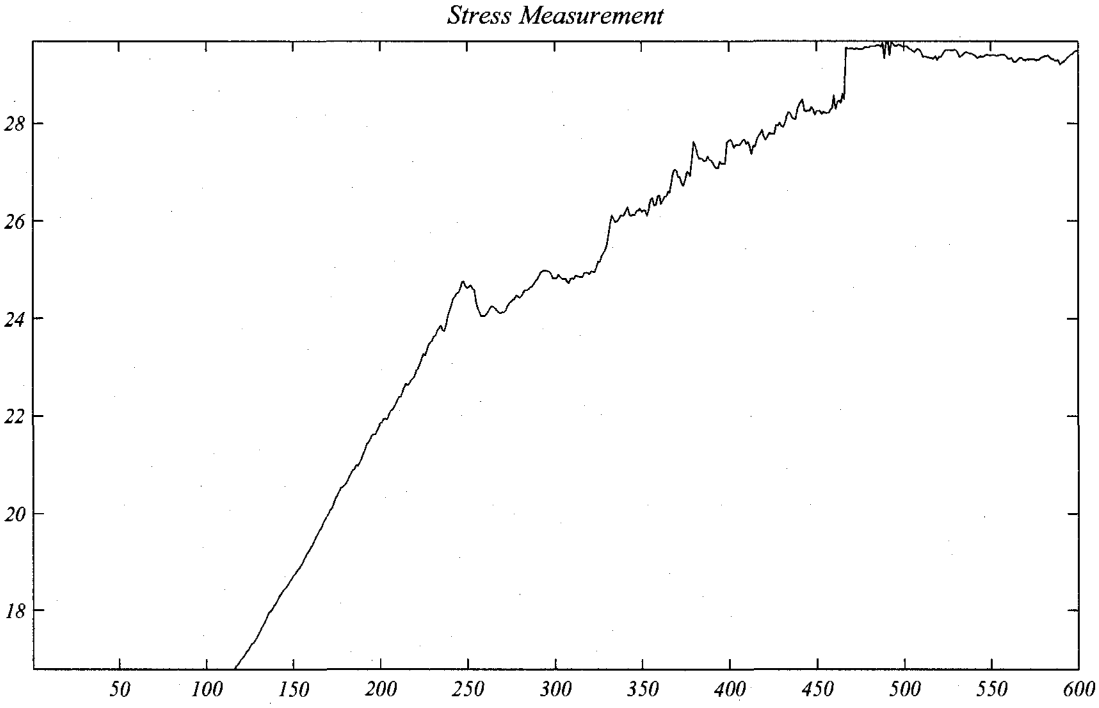
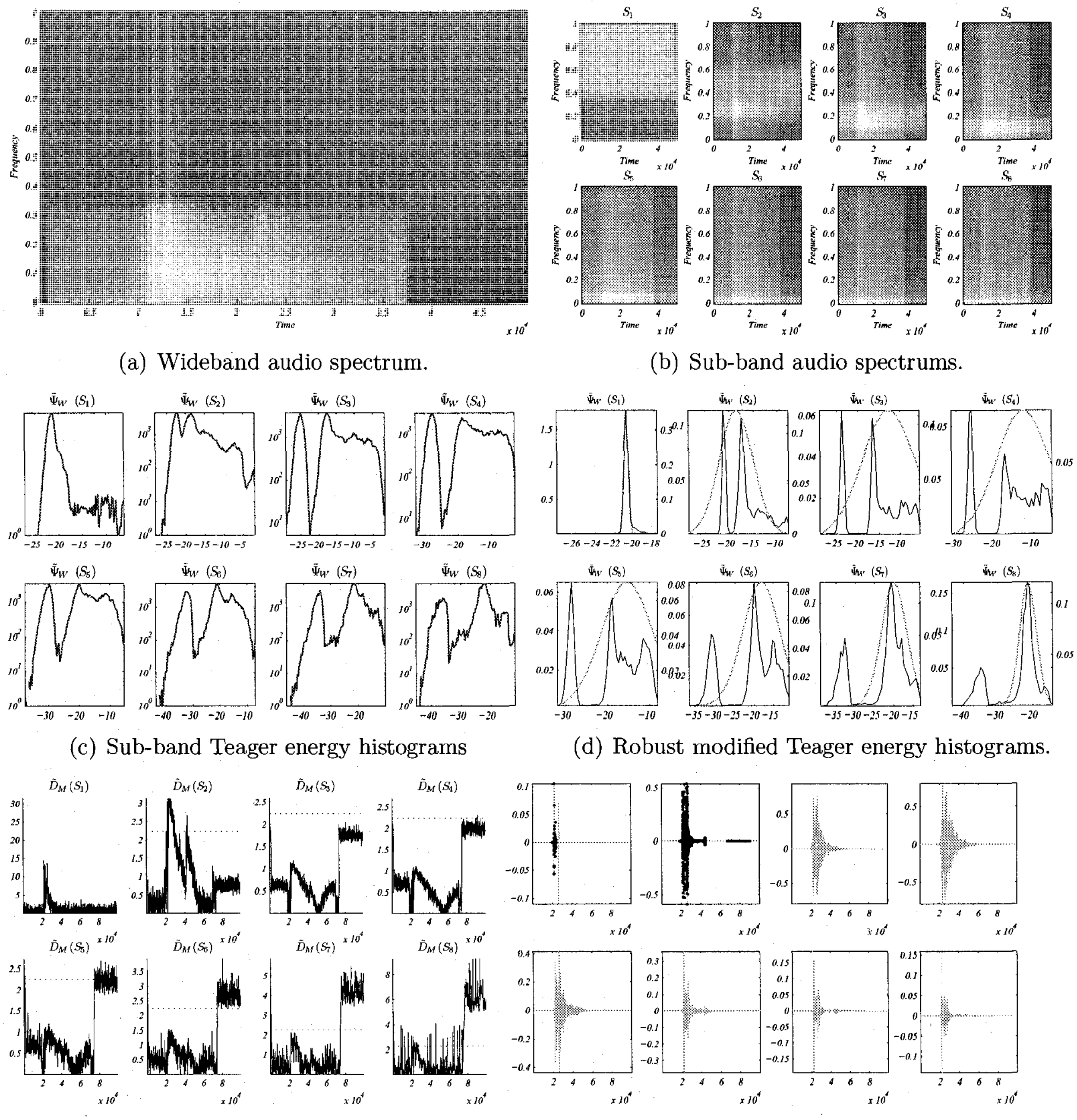


Figure 4.29: System Stress: Rocks hitting each other very hard.

quite far from the robust mean (S_{2-8}). The robust distances in Figure 4.30(e) confirm this and show that they occur when the striking event actually occurs. The dotted line reflects the confidence that 97.5% of the data will have robust Mahalanobis distances below this threshold. Figure 4.30(f) shows the waveforms of each sub-band with those samples detected as minor anomalies in black. The most significant continuous anomaly appears in S_2 and is identified clearly from the silence.



(a) Wideband audio spectrum.

(b) Sub-band audio spectrums.

(c) Sub-band Teager energy histograms

(d) Robust modified Teager energy histograms.

(e) Robust Mahalanobis distances.

(f) Minor anomaly detection

Figure 4.30: Minor Anomaly Detection: Rocks hitting each other very hard, first 100000.

USAT Bomb Blast

This is a digitized tape recording of an actual explosion in the Vale of Belvoir, Leicestershire (United Kingdom, 1988). After the explosion, there is some reverberation that can be heard after which there is a very long silence (see Figure 4.31). *248925 samples were acquired at a sampling rate of 44.1kHz, in 16 bits.*

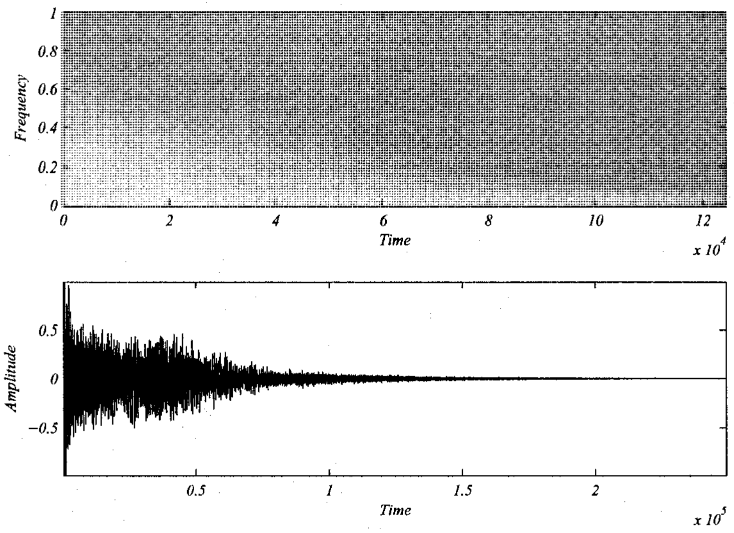


Figure 4.31: The spectrogram and amplitude plot for the USAT bomb blast.

Signal stress measurement was performed on the entire 248925 sample signal. The system stress measure in Figure 4.32 highlights the changing sub-band trends over time. This signal data is similar to what was obtained for the rocks hitting each other and demonstrates the same behavior. System stress is increased as reverberation from the blast continues and tends toward silence.

Figure 4.33 shows the analysis results after minor anomaly detection is applied to the first 100000 samples of the data. This 2.2676 second sub-sample, contains only the explosion as well as the reverberation with a short silence afterward. Figure 4.33(d) shows the modified Teager energy histograms for each of 8 sub-bands, with the robustly estimated Gaussian distribution shown in gray. Sub-band S_1 shows clearly that there are energy anomalies present. To a lesser degree, this also occurs in other sub-bands as well as can be seen in the robust distances in Figure 4.33(e). Inspection

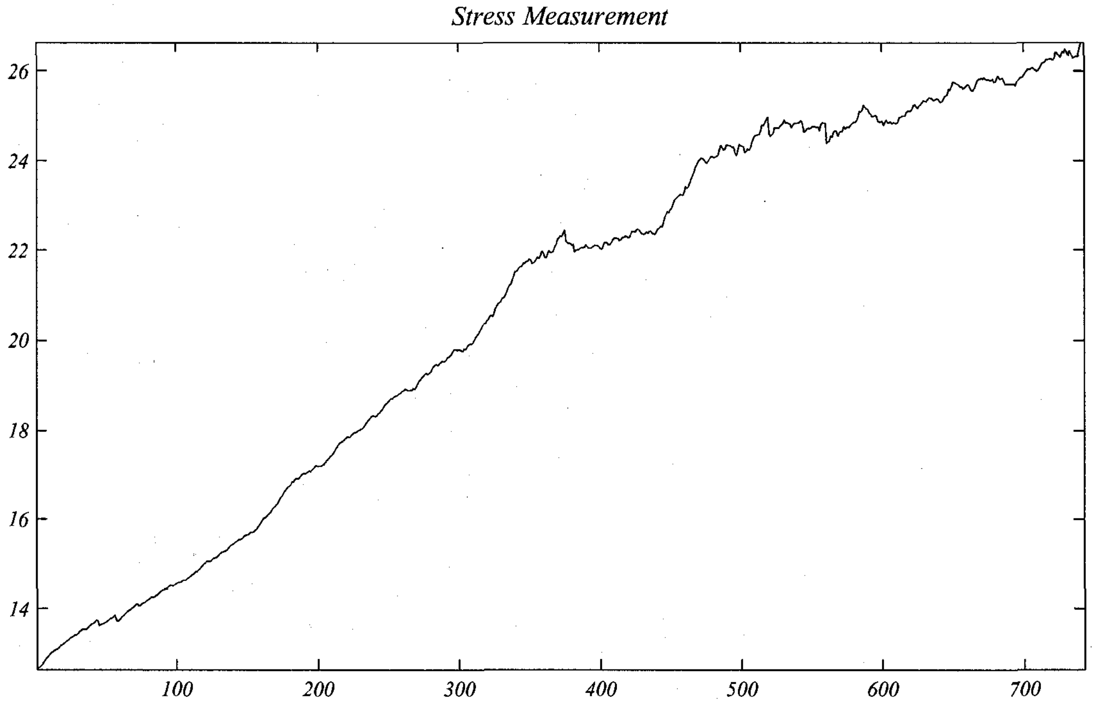
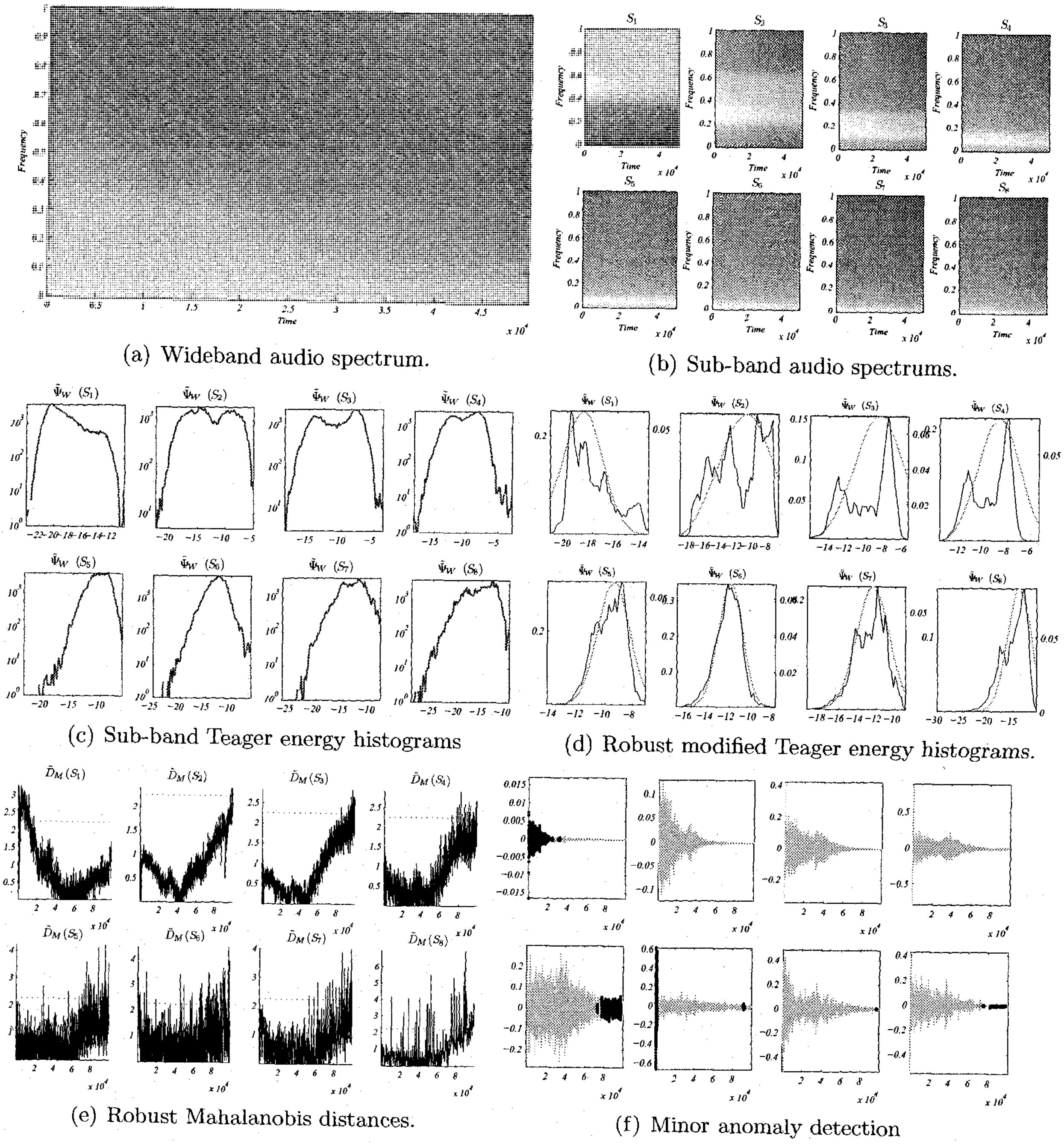


Figure 4.32: System Stress: USAT bomb blast.

of robust distances in all sub-bands show energy deviance at both the beginning and end of the signal. The dotted line reflects the confidence that 97.5% of the data will have robust Mahalanobis distances below this threshold. Figure 4.33(f) shows the waveforms of each sub-band with those samples detected as minor anomalies in black. The anomalies detected correspond to both the initial explosion as well as the silence afterwards.



(a) Wideband audio spectrum.

(b) Sub-band audio spectrums.

(c) Sub-band Teager energy histograms

(d) Robust modified Teager energy histograms.

(e) Robust Mahalanobis distances.

(f) Minor anomaly detection

Figure 4.33: Minor Anomaly Detection: USAT bomb blast, first 100000 samples.

Random Laplacian Audio

This is an audio sample of computer generated random audio with a Laplacian distribution (see Figure 4.34). The random data covers the entire sample without silences or interruptions. *44100 samples were acquired at a sampling rate of 44.1kHz, in 16 bits.*

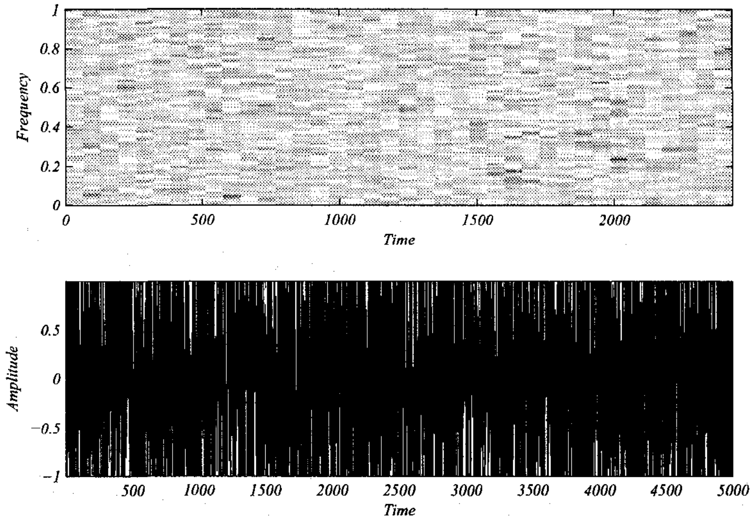


Figure 4.34: The spectrogram and amplitude plot for random Laplacian audio (only 5000 samples shown).

Signal stress measurement was performed on the entire 441000 sample signal. The system stress measure in Figure 4.35 highlights the changing sub-band trends over time. From the scale, it is apparent that the system stress does not change much compared to other signals because all sub-bands demonstrate strong trends.

Figure 4.36 shows the analysis results after minor anomaly detection is applied to the first 100000 samples of the data. This 2.2676 second sub-sample, contains only the random Laplacian audio. Figure 4.36(d) shows the modified Teager energy histograms for each of 8 sub-bands, with the robustly estimated Gaussian distribution shown in gray. There are no significant energies far from the robust mean, and the robust Gaussian fits rather well. The robust distances in Figure 4.36(e) do show some anomalous energies although there are few in number. This is expected with random

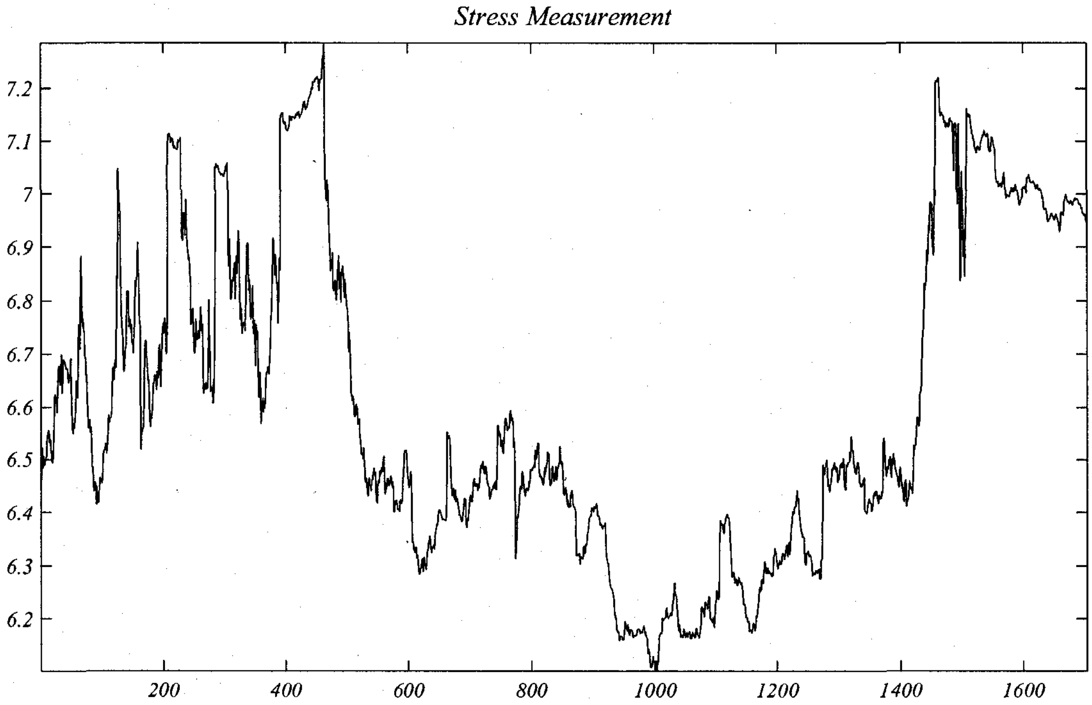
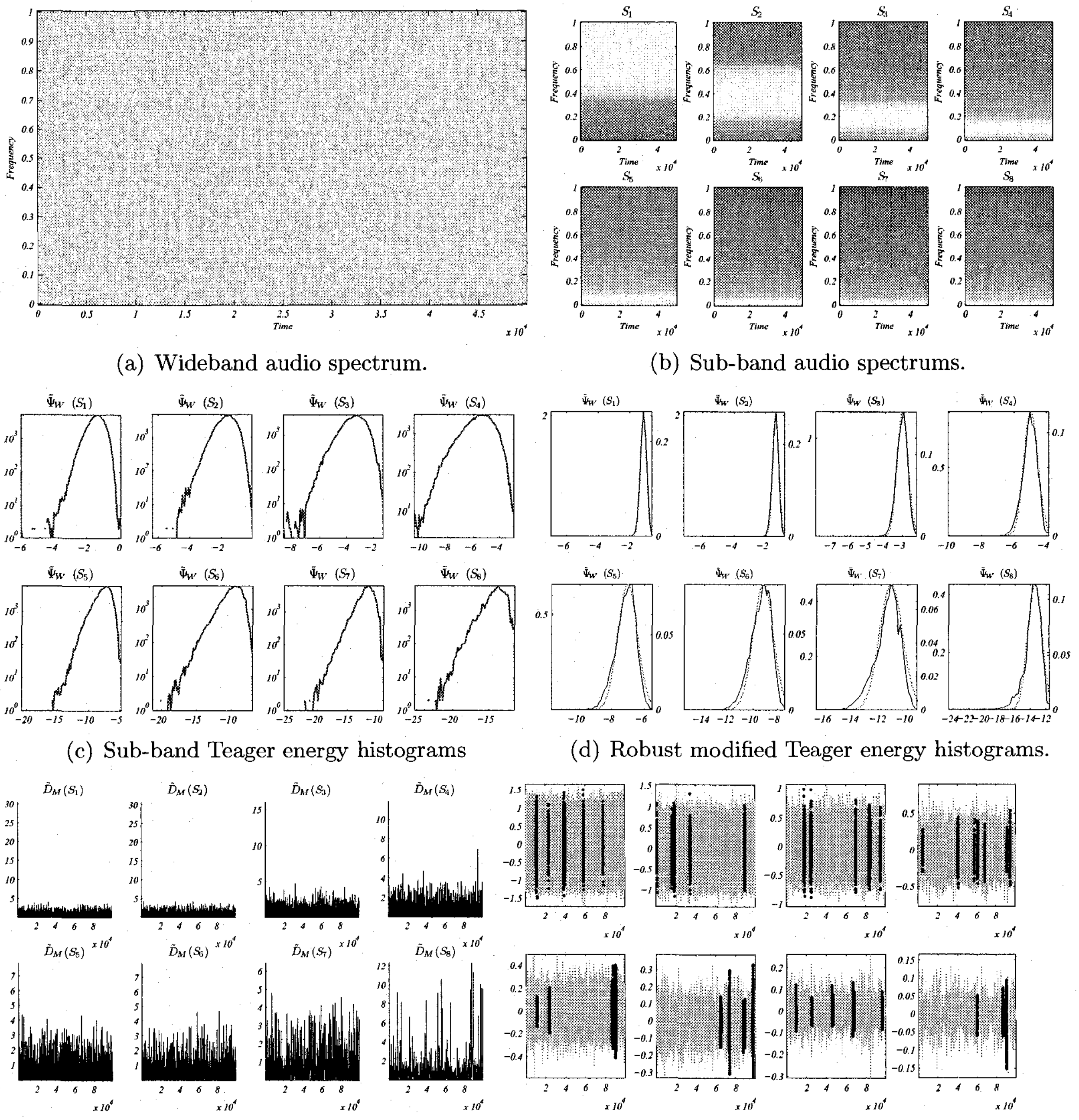


Figure 4.35: System Stress: Random Laplacian Audio

data. The dotted line reflects the confidence that 97.5% of the data will have robust Mahalanobis distances below this threshold. Figure 4.36(f) shows the waveforms of each sub-band with those samples detected as minor anomalies in black. The anomalies detected do not corresponed to any known artefacts in the random data.



(a) Wideband audio spectrum.

(b) Sub-band audio spectrums.

(c) Sub-band Teager energy histograms

(d) Robust modified Teager energy histograms.

(e) Robust Mahalanobis distances.

(f) Minor anomaly detection

Figure 4.36: Minor Anomaly Detection: Random Laplacian audio noise: L(0, 1), first 100000 samples.

4.4 Discussion

The results from these selected experiments highlight some important strengths and weaknesses of the system described in this work. The results of Section 4.2 & 4.3 and 4.2 are now discussed in the context of their specific experimental contribution.

4.4.1 Feasibility for Real-Time Operation

Working with live data was very difficult and highlighted the system's time complexity. Very computationally expensive, the robot first sampled audio and then processed it, repeating this operational cycle again and again. The interpreted software would run in the Matlab environment and was therefore not hardware optimized. The software code itself was partly optimized for Matlab kernel offering some speed improvement. After obtaining a timing profile for the software, it was found that the most serious system bottleneck was the fast-MCD implementation algorithm implementation. While an improvement over its predecessor the latency of this sub-system would require a great deal of hardware optimization. This could be done by implementing kernel or compiled machine-level computer code. Alternatively, a hardware implementation could also provide a great increase in speed. Since minor anomaly detection and consequently system stress measurement depend on robust estimates provided by this sub-system, it is a worthwhile endeavor. Also providing system latency was wavelet band-pass filtering, although this posed less of a problem than the fast-MCD implementation.

4.4.2 Frequency Selectivity

As mentioned in Section 4.3, the robot was placed in various environments with both high and low SNR with respect to the anomaly taken as a signal. In high SNR environments, the system was excellent at detecting minor anomalies, especially when acoustic energy sources were introduced or removed. Major anomaly detection in high SNR environments was also observed to perform well. As experimentation with the robot progressed toward environments with lower SNR, it became apparent that the system had better accuracy with certain types of anomalous signals over others. In low SNR environments, acoustic anomalies with lower frequencies were easier to detect than those with higher frequencies. The uneven frequency weighting of the wavelet band-pass filters are responsible for this. When higher frequency anomalies are passed through the large bandwidth of the first levels of wavelet filters, narrow band anomalies do not experience any improvement in SNR in that sub-band, especially when the ambient noise has much energy in this sub-band as well. Even if the higher sub-bands do not contain energy, the filter's bandwidth is so wide that the cumulative effect of the normally insignificant high frequency low-energy sources will hinder the attempt at improving the SNR of the anomaly so it can be detected. The wavelet filters have a decreasing bandwidth as their center frequencies decrease. Therefore, anomalies with predominantly low frequencies have a better chance of being spectrally isolated and may have an increased SNR in that sub-band provided the ambient noise does not mask the anomaly in *all* bands.

The system's insensitivity for high frequency anomalies, can be mitigated by increasing the number of wavelet band-pass filters by increasing the size of the minor anomaly detection buffer. Increasing the size of the signal sample however, will result in a greater number of samples for the fast-MCD algorithm which the minor anomaly detection algorithm depends on. Because of the complexity of the fast-MCD algorithm, the system latency will increase dramatically. In this sense, improving the SNR of narrow-band high-frequency anomalies will come at a serious performance cost. Another solution could be the redistribution of band-pass filters so that all frequencies are covered with a small, but equal bandwidth.

4.4.3 Minor Anomaly Detection Specificity

Some of the pre-recorded sub-samples selected for minor anomaly detection contained events that were uncharacteristic with respect to the acoustic scene as observed in the sample. Most of these events were labeled by a human observer as being an anomaly. This is a good reference since humans have an excellent ability to detect anomalies, better than any known device or mechanism. The 3 phase blower motor data had a very clear snapping noise as the motor started up which did not appear in any other part of the data. This was detected very specifically as an anomaly, as was the sound of the transient speed of the motor as it ramped up before reaching a steady speed. These had very different frequency and amplitude characteristics when compared to other parts of the signal giving them a very different Teager energy than in the rest of the signal. This was also the case with the Minolta camera attempting to focus. This anomaly was very clearly identified because of its very high SNR in almost all sub-bands. This high SNR was primarily due more to the fact that the event occurred amidst a predominantly silent acoustic background than due to the spectral decomposition. This was seen in other data as well where temporal isolation of the anomaly was excellent, such as where rocks were hitting each other very hard. In contrast, there were some data sets such as those acquired from the gear reduced motor running at low RPM and the USAT bomb blast which had uncharacteristic silences in the acoustic scene. These low energy outliers were identified very accurately.

The samples with low SNR confirmed that minor anomaly detection would still perform reasonably well in identifying anomalies. Almost all of the fireworks data had events that were embedded in ambient noise. When the anomaly had a differing spectral energy than the rest of the background, it was clearly identified. In one case, the sub-sample of the fireworks with human screams, the firecracker made a high pitched sound while in flight amidst other sounds which was identified as uncharacteristic. In another case where acoustic data was obtained from the 1980 Mount St.

Helens's volcano the ambient noise was of very low amplitude, however - the seismic activity from the eruption 140 miles away was barely audible. The natural sounds of the acoustic scene consisted predominantly of higher frequency bird calls. Because of the finer spectral discrimination at lower frequencies offered by the filtering strategy, the seismic event in the sub-sample used was identified, although not as clearly as with high SNR anomalies.

One sample contained no events at all: the random Laplacian audio. In this case, anomalies were still detected. This data suggested that the system would find anomalies where there was none. The anomalous samples accounted for approximately 3% of the testing sample. Because the system has confidence on 97.5% of the robust distances (and therefore samples corresponding to each), the rest are labeled as an anomalies. In this sense, we are observing a *false detection rate* of appoximately 3% for this particular sample. This is entirely expected, but should be taken into account when the ratio of anomalies detected approaches 2.5%.

4.4.4 System Stress and Context Change

Humans are exceptional at identifying not only the context of an acoustic scene, but when it has changed. Stress measurements for the selected data sets were compared with the subjective opinion of human observers. While this provided some subjective insight into the dynamics of the acoustic scene, it did not detract from the fact that in general, when sub-band energies would change - system stress would increase, and when they would settle, system stress would decrease. This in itself is a good indicator that activity in the acoustic scene has changed, which implies that it's context has changed. Minor anomaly detection was quite selective and in some cases quite accurate, but this may pose a problem. System hypersensitivity can result given the right parameter settings, possibly rendering minor anomaly detection if used alone. Stress measurement for the purpose of detecting context change acts as a kind of filter for minor anomalies. In this sense, major anomalies therefore have the property that they reflect an energy deviation locally in signal sub-bands, while taking into account a larger view of the signal across sub-bands, and over a longer period of time.

In practice, minor anomaly detection (or localization) can be used separately depending on the level of information required after processing. Typically, once an anomaly is detected (or localized), a system would expend energy to handle the situation further. If this post-processing is computationally intense, requires resources, or is just very costly in some sense, then major anomaly detection (or localization) should be used.

Chapter 5

Discussion & Conclusion

5.1 Discussion

In this work, signal characterization plays a key role the detection of anomalies. The Teager energy operator is successfully used with the *modulating source assumption* to characterize a source by its total energy measured in sub-bands. This demodulating operator's sensitivity to amplitude and frequency make it far superior to classical energy measures which tend to be very insensitive to signal shape.

A narrow-band source whose energy is undetectable when immersed in the wide-band signal can be given spectral emphasis with the use of a band-pass filter before Teager energy characterization. The modulating source assumption is extended to sub-bands. This considers a spectrally decomposed signal as having composite energies in each band characterized by the Teager source model. A scaled wavelet function acting as a band-pass filter offers very sharp transition bands halving its bandwidth and center frequency for each decomposition level. This non-linear spacing provides finer discrimination in lower spectral bands, where most audio energy tends to reside, and coarser spectral discrimination in higher spectral bands where there is less energy from a typical audio source.

Detecting deviations in sub-band energy requires some pre-processing in order to detect statistical outliers. Laplacian distributed audio yields Teager energy that does not conform to any known standard distribution. A random variable transformation was designed that yields a Gaussian audio sample from a Laplacian audio sample. The window-averaged Teager energy of the transformed Gaussian audio is approximately log-Gaussian distributed. This transformed Teager energy can be re-distributed into a Gaussian distribution using a trivial variable transformation. On the understanding that the random variable transformation functions are all one-to-one, and that the Teager energy operator yields instantaneous energy, statistical outliers in the transformed Teager energy correspond to the audio samples that generated them.

Measuring energy dissimilarity is done with the Mahalanobis distance measure. Its sensitivity to energy location and scatter is mitigated by using a high-breakdown estimator known as the Fast-MCD. Through successive resampling, this method is very robust to outliers and provides estimates that describe the majority a given sample set. This robust estimator is used to obtain the mean and covariance for the transformed Teager energy. Because it is Gaussian distributed, the Mahalanobis distance is Chi distributed. Given a confidence level the inverse-Chi distribution will give a cut-off value for the Mahalanobis distance. Energies with a Mahalanobis distance greater than this value are are considered as energy outliers. The confidence level therefore represents the degree of expectation held that energies in the sample belong to the distribution whose parameters were determined with Fast-MCD.

Samples that generate outlier energies are labeled as one of two types: *minor anomalies* and *major anomalies*. A minor anomaly is a set of samples that produced outlier Teager energies in a sub-band. Typically, this implies (but is not restricted to) signal samples that are high-amplitude, high-frequency as well as those that are low-amplitude, low-frequency when compared to other samples in a buffer. Minor anomalies can occur in some, all, or no sub-bands.

Minor anomalies are re-labeled as major anomalies when the observing system experiences strong deviations in its stress levels. These stress levels are determined by the robust signal trends in each of the sub-bands. These trends, or *contexts* are tracked over time and the presence of context outliers signal that any minor anomalies that appear have done so during a fundamental change in the signal across all of its sub-bands. This sign of signal volatility provides additional meaning to a minor anomaly in the context of the whole signal.

For each sub-band, only major anomalies are localized using the cross-correlation technique. With a location estimate obtained for each sub-band, the final position of the anomaly is the median estimate. This final anomaly localization is only provided if major anomalies are detected. In other words, if a major anomaly is detected, its location is returned by the system, otherwise the system remains dormant.

5.2 Future Work

In the literature, anomaly detection does not receive mainstream attention primarily because of the non-specificity of the problem and the general need to set strict operational bounds in the statistical sense. It was the underlying attempt of this work to provide a framework for structured advancement in this area of research. By reducing the problem to one in robust random statistics, a formal technical language can be adopted for furthering development of other types of anomaly detectors. What follows is an outline of some of the more interesting areas of research that could stem from this work in future endeavors.

5.2.1 Performance Metrics for Anomaly Detectors

Anomalies, by definition are unexpected events that violate an observer's expectations. Reduced to a problem in robust statistics, anomaly detectors center their operation on the establishment of a norm from which deviations are measured. Signal characterization, deviation measure, outlier detection, filtering are all sub-systems that are subject to variation amongst other anomaly detectors. Since this work is based in statistical measure, a statistical measure of performance is required not only for the anomaly detector of this work but of other works as well. Establishment of performance metrics are required to not only to compare anomaly detectors, but to move toward the full parametrization of the anomaly detection problem in general. In this fashion, targeted optimizations can be performed on methods that can be standardized based on need.

5.2.2 Tracking Context Movement in Sub-Band Space

In this work, *context* served to provide some sense of volatility of the source. *Stress*, the context deviation measure, is established from a system expectation that the signal was not going to change in its trends across all sub-bands. Implicitly, it is assumed that the contexts conform to a single multivariate Gaussian in sub-band frequency space. The implication here is that contexts will have a single expected value. For complex signal environments, the mean may develop several expected values over time. A Gaussian mixture could be estimated for contexts that would provide a concise system memory for contexts. Context identification by measuring stresses to each of the cluster centers. Identification by minimum stress however, would not capture the nature of an evolving signal scene. By tracking mean context movement, the anomaly detector can concisely store the nature of a highly dynamic signal scene. Also, as a simple memory, expected context sequences can be established using minimum stress context identification. Context *sequence deviations* can be detected in this fashion.

5.2.3 Improving Blind Source Separation

Recent advances in psychology suggest that the method in which the human mind can decompose convolved sources involves some sort of anomaly detection. Deconvolution, or blind-source separation is a well explored problem in machine learning and statistics. For signal scenes where sources are added (or removed) over time, anomaly detection can be used to identify the number of sources that have appeared, and give some indication of which spectral bands they appear. For wide-sense stationary signals, the anomaly's autocorrelation function could give a hint that would improve the source-separation process. The assumption here, is that by identifying the onset of novel signal components and isolating them in time and frequency, their statistical properties could improve any attempts at signal separation.

5.2.4 Anomaly Detection in Graphs

Graphs appear in many branches of science describing systems and processes. For applications where many graphs are analyzed with small variations, an anomaly detector could be designed to identify graphs that are deviant with respect to what is expected. For example, mapping specific complex metabolic pathways for large populations of micro-organisms should yield similar graphs. Mutant organisms my have alternate pathways that would normally be undetected. Anomaly detection could provide a researcher a means of identifying mutant populations by their metabolic-pathways. By establishing dissimilarity measures for complex graphs as well as a manner in which to statistically characterize them, anomaly detection can be very useful for identifying not only that a graph violates an observer's expectations about it, but also to identify the part of the graph that causes the violation.

5.2.5 Anomaly Detection in Complex Polyhedra

Structures such molecular compounds, proteins, mechanical structures, or manufactured surfaces, can be described using complex polyhedra. By attempting to characterize the variations in such structures, expectations can be formed. Detection of abnormal structures could prove useful for many purposes including process refinement in the case of manufacturing, or identification of novel proteins for use in automated drug or disease discovery agents. Since descriptors using polyhedra are widely used, novel variations from what is expected can be studied for their relevant properties.

5.2.6 Probability Distribution of Teager Energy

Attempted for this work, it was found that the probability distribution function for the Teager energy operator was found to be non-trivial. It could be useful to determine the Teager energy probability distribution for various types of random variables. This could lead to an interesting class of maximum-likelihood estimators that are implicitly frequency and amplitude sensitive.

5.3 Conclusion

The problem of anomaly detection and localization has been reduced to a problem in robust statistics. An automated observer was designed to detect when high energy sources are introduced into an acoustic scene. The modulating source assumption offered a means for measuring total energy in a source using the Teager energy operator. Accounting for potential energy from signal amplitude, and kinetic energy from signal frequency in wavelet-filtered sub-bands a robust statistical characterization scheme was developed. With an expectation of energy content in sub-bands, a detection scheme was designed to detect signal energies that violated that expectation. These minor anomalies provide some sense that a fundamental change in energy has occurred in the sub-band. By examining how the signal is changing across all sub-bands, a detector was designed that was able to determine when a fundamental change occurs in the sub-band signal trends. Minor anomalies occurring during such changes were labeled as major anomalies. Using established localization methods, position estimates are obtained for the major anomalies in each sub-band. Accounting for the possibility of a source with spatio-temporal properties, the median of sub-band position estimates provides the final spatial information about the source.

The hypothesis declared in Section 3.1.2 appears to hold true highlighting the success of this work. The problem of anomaly detection has been successfully treated as a problem in robust statistics. The modulating source assumption applied to each band of a spectrally decomposing a signal allowed for total instantaneous energy to be measured which, over time allowed for joint amplitude and frequency features to be exposed using the demodulation properties of the Teager energy operator. Sensitivity to narrow-band sources in the lower audio bands was increased due to increased spectral discrimination where there was the majority of acoustic energy.

Over short observation periods, statistical deviations in sub-band Teager energy samples provide some indication that there was some significant event in that sub-band. Collectively, if the signal trends in each of the filtered sub-bands changes significantly over a larger observation period, then the significance of events in the sub-bands are given more weight and are used for localization in the far acoustic field.

This work contains several contributions that have been published in a paper at a joint conference of the IEEE International Midwest Symposium on Circuits and Systems (MWSCAS 2007) and the IEEE International North East Workshop on Circuits and Systems NEWCAS (IEEE-NEWCAS 2007) in a paper entitled: "Sub-Band Anomaly Detection and Spatial Localization" (See Appendix A). The major contribution was the use of random variable transformation so that energy outliers can be detected in sub-bands of an acoustic signal. This stems from observing that a moving average of the Teager energy of a Gaussian signal is approximately log-Gaussian. Use of a high-breakdown estimator to characterize a transformed Teager energy distribution is also a contribution. Finally, a minor contribution was the method of resolving spatial information about an anomalous source based on short the duration events detected in in sub-bands.

References

[1] Brent C. Kirkwood. Acoustic source localization using time-delay estimation. Master's thesis, Technical University of Denmark, 2003.

[2] Ehud Ben-Reuven and Yoram Singer. Discriminative binaural sound localization. In *Advances in Neural Information Processing Systems.* MIT Press, 2002.

[3] E. Grassi and S. A. Shamma. A biologically inspired, learning, sound localization algorithm. In *2001 Conference on Information Sciences and Systems*, pages 344–348, University of Maryland, College Park, MD 20742, USA, March 21-23 2001. The Johns Hopkins University.

[4] *Finding Events automatically in Continuously Sampled Data Streams via Anomaly Detection.* IEEE National Aerospace & Electronics Conference (NAECON), October 2000.

[5] D. Modha and E. Masry. Memory universal prediction of stationary random processes. *IEEE Transactions on Information Theory*, 44(1), January 1998.

[6] Andrew K. Chan Jaideva C. Goswami. *Fundamentals of Wavelets, Theory, Algorithms and Applications.* John Wiley & Sons, 1999.

[7] C. Penna J. W. Resende, M. L. R. Chaves. Identifiication of power quality disturbances using the matlab wavelet transform toolbox. Technical report, Universidade de Uberlandia, 2001.

[8] John Potter Paul Seekings. Classification of marine acoustic signals using wavelets & neural networks. In *Proceeding of 8^{th} Western Pacific Acoustics Conference*, 2003.

[9] Firas Jabloun. Large vocabulary speech recognition in noisy environments. Master's thesis, Bilkent University, 1998.

[10] John F. N. Salik. Sub-band anomaly detection and spatial localization. In *2007 IEEE Midwest Symposium on Circuits and Systems and IEEE North Eastern Workshop on Circuits and Systems Joint Conference on Circuits and Systems.* École Polytechnique de Montréal, 2007.

[11] Eivind Kvedalen. Signal processing using the teager energy operator and other nonlinear operators. Master's thesis, University of Oslo, May 2003.

[12] Wei Lin, Chris Hamilton, , and Prabhakar Chitrapu. A generalization to the teager-kaiser energy function & application to resolving two closely spaced tones. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 1637–1640, May 1995.

[13] Samuel Copt and Maria-Pia Victoria-Feser. Variable fast algorithms for computing high breakdown covariance matrices with missing data. Cahiers du Département d'Econométrie 2003.04, Département d'Econométrie, Université de Genève, August 2003.

[14] D. L. Donoho. *Breakdown Properties of Multivariate Location Estimators.* PhD thesis, Harvard University Department of Statistics, 1982.

[15] Peter J. Rousseeuw and Katrien Van Driessen. A fast algorithm for the minimum covariance determinant estimator. In *Technometrics*, volume 41, pages 212–223. American Society for Quality Control and American Statistical Association, August 1999.

[16] Athanasios Papoulis. *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill, 1991.

[17] F. W. M. Stentiford. An evolutionary programming approach to the simulation of visual attention. In *Proceedings of the 2001 IEEE Congress on Evolutionary Computation Seoul, Korea*, Adastral Park, Martlesham Heath, Ipswich, UK, May 27-30 2001. BTexaCT Research.

[18] Pedro Julián, Andreas G. Andreou, Larry Riddle, Shihab Shamma, David H. Goldberg, and Gert Cauwenberghs. A comparative study of sound localization algorithms for energy aware sensor network nodes. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 51(4):640–648, April 2004.

[19] Greg L. Reid and Evangelos Milios. Active stereo sound localization. Technical report, York University, Department of Computer Science, 4700 Keele Street North York, Ontario M3J 1P3 Canada, December 1999.

[20] Dwight F. Mix. *Random Signal Processing.* Prentice Hall, 1995.

[21] Christian Blater. *Wavelets: a primer.* A.K. Peters Ltd., 1998.

[22] George N. Saridis Kimon P. Valavanis. *Intelligent Robotic Systems: Theory, Design and Applications.*, volume Volume 182 of *Kluwer International Series in Engineering and Computer Science.* Kluwer Academic Press, 1992.

[23] Daniel Barbará and Ping Chen. Tracking clusters in evolving data sets. In *Proceedings of the Fourteenth International Florida Artificial Intelligence Research Society Conference*, pages 239–243. AAAI Press, 2001.

[24] Anthony J. Weiss and Ehud Weinstein. Fundamental limitations in passive time delay estimation - part 1: Narrow band systems. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-31(2):472–486, April 1983.

[25] Ehud Weinstein and Anthony J. Weiss. Fundamental limitations in passive time-delay estimation - part 2: Wide-band systems. *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-32(5):1064–1078, October 1984.

[26] John C. Murray, Harry Erwin, and Stefan Wermter. Robotic sound-source localization and tracking using interaural time difference and cross-correlation. In *Proceedings of NeuroBotics Workshop*, pages 89–97, September 2004.

Appendix A

Publication: MWSCAS 2007 / NEWCAS 2007

This work contains several contributions that have been published in a paper at a joint conference of the IEEE International Midwest Symposium on Circuits and Systems (MWSCAS 2007) and the IEEE International North East Workshop on Circuits and Systems NEWCAS (IEEE-NEWCAS 2007) in a paper entitled: "'Sub-Band Anomaly Detection and Spatial Localization"' (See Appendix A). The major contribution was the use of random variable transformation so that energy outliers can be detected in sub-bands of an acoustic signal. This stems from observing that a moving average of the Teager energy of a Gaussian signal is approximately log-Gaussian. Use of a high-breakdown estimator to characterize a transformed Teager energy distribution is also a contribution. Finally the last, but minor contribution, was the method of resolving spatial information about an anomalous source based on short duration events detected in in sub-bands.

The conference was held in Quebec, Canada on August 5-8 at the Marriott Chateau Champlain Hotel in downtown Montreal. The year 2007 will mark the 50th anniversary of MWSCAS and the 5th of NEWCAS, both sponsored by the IEEE.

Sub-Band Anomaly Detection and Spatial Localization

John F. N. Salik
Department of Electrical and Computer Engineering
Concordia University
Montréal, Québec
Canada, H3G 2W1
Email:

I. Introduction

An *anomaly* is the specific event that causes the violation of a process observer's expectations about that process. When a context has changed significantly within a qualified scent, an intelligent system identifies this event as an anomaly. While scene and context features can vary across observers, the existance of a detection mechanism for significant context change is a salient feature of intelligent observers. By detecting an anomaly, an intelligent system can apply a fitting control law to accomodate the new context or initiate learning to adapt or discover a new control law that is appropriate to maintain stability the presence of the altered context without compromising previously established control laws.

The method described here for anomaly detection and localization first attempts to statistically characterize wavelet filtered sub-bands which is especially important when the narrow band power of an anomaly is insignificant when compared to that of the wide band signal. By distinguishing between extreme and outlier Teager energy values that have appeared in the sub-bands of array sensor data. The outlier data in the time-frequency window can then be used to estimate array phase data required for computing acoustic wavefront direction of arrival in the far-field.

II. Total Energy of a Source

A. Demodulation Properties of the Teager Energy Operator

Designed in an attempt to obtain the *total energy* of a signal source, the Teager energy operator assumes the source model to be analogous to a simple spring-mass system. Newtonian physics describes the total energy of the spring-mass system in motion as the sum of both the spring's potential energy and the mass's kinetic energy. For a natural excitation we have:

$$E_T = \frac{1}{2} m \omega^2 A^2 \qquad (1)$$

The total energy E_T of this system is clearly a function of both the amplitude of the oscillation A and the frequency of oscillation ω which are scaled by the constant mass m. Consequently, Teager energy is sensitive to amplitude for a fixed frequency, sensitive to frequency where the amplitude is fixed and is simultaneously sensitive to both. The Teager energy operator, which measures total energy in this system, can therefore be used for the demodulation of AM, FM, or AM-FM signals. Both the continuous and discrete forms of the Teager energy operator share this property.

B. Modulating Source Assumption

Let us assume that all signals of interest have been *modulated* somehow. Demodulation then, would expose features of the signal that could help characterize it among other signals. For example, AM demodulation would expose changes in the signal's envelope for relatively constant frequencies. FM demodulation would expose changes in frequency for a relatively fixed signal envelope. Finally AM-FM demodulation would expose these two features simultaneously. The Teager energy operator can be used successfully in each of these three demodulation modes without any modification or additional computational cost. If we assume the force on the spring-mass system has been modulated, we can make use of the demodulation properties of the Teager energy operator for joint amplitude and frequency feature extraction. We will call this the *modulating source assumption.*

C. Difficulties with Teager Energy

Where we adopt the modulating source assumption, the Teager energy operator is not used for signal characterization without drawbacks. The discrete Teager energy operator has the following definition:

$$\Psi\left[x_n\right] = x_n^2 - x_{n-1}x_{n+1} \qquad (2)$$

Clearly non-causal, there are more serious properties may pose a problem for signal characterization. Notably, the problem of *negative Teager energy,* and *noise sensitivity.*

Teager energy yields negative energy for certain types of signals which is a strange behavior for *any* energy operator. In acoustic signals, this typically occurs for very few samples at a time, but it does occur. The *average Teager energy* for a discretely sampled signal in an arbitrary sub-band is taken as:

$$\hat{\Psi}_s[x_{n,s}] = \frac{1}{W}\left|\sum_{n=1}^{W} x_{n,s}^2 - x_{n-1,s}x_{n+1,s}\right| \qquad (3)$$

Where W is the number of samples to be averaged, s is the index of a particular sub-band of interest, and $x_{n,s}$ is a particular sub-band sample. This moving average, which acts as a low pass filter, mitigates the problem of rare negative energy as well as variations that are due to differentiation noise. Its averaging period W effectively defines its sensitivity in these respects. A detailed explanation of how to guarantee positivity of the energy measurement is not appropriate here, but we should recall that the Teager energy model tries to model the energy of the *source* and not the signal although we speak colloquially to the contrary. The author of [1] suggests that if we consider the observed signal generating negative Teager energy was generated by *two* sources, each generating a sinusoid with one farther away and with higher frequency, then Teager energy measurement will be based on an incorrect assumption of a single source system. This is a very reasonable explanation.

D. Wavelet Band-Pass Filtering

While typically used for time-frequency analysis of signals, the discrete wavelet transform (DWT) has another use: band-pass filtering. A scaled wavelet function will act as a band-pass filter halving its bandwidth for each decomposition level. A direct result of the dyadic sampling scheme used in the DWT, the center frequency of the band-pass filters are nonlinearly spaced across the signal spectrum. Audio data normally has most of its power in the lower spectral bands therefore we would expect that most of the characterization information would also reside there. In this sense, we would choose to have a finer spectral discrimination in lower spectral bands, and coarser spectral discrimination in higher bands. Discrete wavelet decomposition is a very good choice since the band-pass filtering has many of the filters centered in the lower frequencies and fewer in the higher frequencies. In addition to the sharp transition bands offered by wavelets, the natural nonlinear spacing of the band-pass filters is a good choice for audio analysis and comes at minimal cost. With appropriate band-pass filtering, energy measurement can be done in the sub-bands of a signal. Spectral characterization in this fashion can highlight important energies in sub-bands that would otherwise appear as insignificant over the full spectrum of the signal. For highly tonal sources, band-pass filtering effectively increases the SNR of the source's signal in the sub-band containing the tones.

III. Robust Energy Processing

A. Outlier Detection with the MCD

Estimates for statistical model parameters are typically the result of some cost function minimization over all sample points. Consequently, this best estimate may be inaccurate for the model chosen where there exists samples from another distribution that have strong leverage over the cost function and therefore the estimation process as a whole. Not to be confused with extreme values along the asymptotic tails of some distributions, these *outliers* are those samples that belong to a distribution other than the one of the majority of the samples.

Given a fully-specified multivariate Gaussian distribution one can measure the degree of membership of a new sample vector by its Mahalanobis distance. Unlike Euclidean distance which does not take scale into account, Mahalanobis distance is measured in units of the standard deviation: $\mathrm{D}(\vec{x}) = \sqrt{(\vec{x}-\vec{\mu})\Sigma^{-1}(\vec{x}-\vec{\mu})^T}$. If the components of the vector are independent, and normally distributed then $\mathrm{D}(\vec{x})$ is Chi (χ_k) distributed with its k degrees of freedom equal to the number of dimensions of $\vec{x}$. With a confidence interval specified, a minimum distance can be obtained for inclusion into the distribution. Unfortunately, Mahalanobis distance is *very* sensitive to the scatter matrix Σ. This poses a problem where the covariance matrix was estimated with outliers present. Therefore, to obtain good, robust Mahalanobis distance measures, we require robust estimation of the covariance matrix.

For a multivariate Gaussian distribution, the FAST-MCD algorithm [2] has been used successfully for covariance matrix estimation in the presence of outliers using the minimum covariance determinant (MCD). By successive selection of sample subsets, only those samples that have the lowest scatter characteristics in the greatest numbers are used to estimate Σ. With an outlier robust covariance matrix, *robust distance estimates* can be made. The robust χ_k distributed distance measures can then be thresholded given a confidence interval to identify outliers. This method is very good for determining which samples have been introduced into the robustly estimated Gaussian distribution from an arbitrary, unknown distribution. What is important here is that outlier detection using the FAST-MCD method requires that the majority of the samples are *Gaussian distributed*, with all other samples designated as outliers.

B. Random Variable Transformation

Randomly sampled audio data is typically Laplace distributed. A double-sided exponential, this distribution has much higher kurtosis than the Gaussian distribution although they are both symmetrical and asymptotic. The FAST-MCD algorithm can be used on Laplace distributions, but the confidence interval can not be used accurately for outlier detection. With this in mind, consider the following: Given the known cumulative distribution $\mathrm{F}_x(\mathbf{x})$ of the random variable $\mathbf{x}$, we can find the function $\mathbf{y} = \mathrm{g}(\mathbf{x})$ for a specified $\mathrm{F}_y(\mathbf{y})$. In the general case [3], we find that if $\mathbf{y} = \mathrm{F}_y^{-1}(\mathrm{F}_x(\mathbf{x}))$ then $\mathrm{P}(\mathbf{y} \leq y) = \mathrm{F}_y(\mathbf{y})$. Therefore, if $\mathbf{x}$ is Laplacian distributed audio data then the output of the following random variable transformation is Gaussian distributed:

$$\mathrm{g}(\mathbf{x}) = 2b\,\mathrm{erf}^{-1}\left\{\mathrm{sgn}(\mathbf{x}-\hat{\mu})\left(1-e^{-\frac{|\mathbf{x}-\hat{\mu}|}{\hat{b}}}\right)\right\}+\hat{\mu} \quad (4)$$

Where $\hat{\mu}$ is the *sample median* of $\mathbf{x}$ and the estimator for $\hat{b}$ is:

$$\hat{b} = \frac{1}{N}\sum_{i}^{N}|x_i - \hat{\mu}| \quad (5)$$

By changing the distribution of the input data in this fashion, we gain accuracy in the detection of outliers for *signal amplitude data*, however this is not the primary justification for using $\mathrm{g}(\mathbf{x})$.

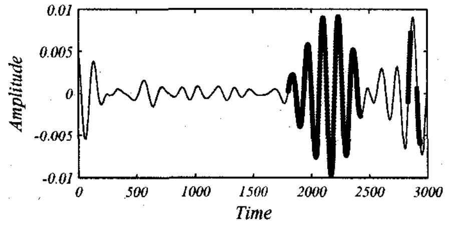


Fig. 1. Anomalies detected in a low frequency audio sub-band.

C. Detection of Teager Energy Outliers in Audio Samples

If we consider the Teager energy operator as a random variable transformation, its distribution is non-trivial for input samples that have a Laplacian distribution. Therefore the decision criteria for outliers will also be non-trivial. If the input samples are Gaussian distributed, then we find empirically that Teager energy is approximately log-Gaussian distributed. Converting a log-Gaussian random variable $\mathbf{v}$ to a Gaussian one is simply done using the following transformation of random variable:

$$\mathrm{h}(\mathbf{v}) = \ln \mathbf{v} \qquad (6)$$

Therefore through successive transformation of variable, robust distance measures designed for Gaussian distributions can be used to identify outlier Teager energies in Laplace distributed sub-bands.

IV. Anomaly Detection & Localization

A. Minor Anomalies

In the context of this work, a *minor anomaly* is considered to be an uncharacteristic increase in sub-band amplitude, frequency, or both. Treated as an outlier detection problem, for time-series samples, we know intuitively that anomalous energies that appear in a signal are likely to be from a continuous source. Therefore, we reject single, non-adjacent samples that were likely a result of the noisy differentiation in the discrete Teager Energy definition (2). Figure 1 shows an example where continuous outliers in a signal form anomalies in a wavelet-filtered band-passed audio signal. Given Laplace distributed audio data, the outlier detection process is summarized in Figure 2. Alone, this method itself is very useful for analysis of complex sound as it can detect those events that are *out of place* such as the onset of speech in a noisy environment or mechanical malfunction in the acoustic signature of machinery. This method is also a useful first step in attempting to *remove* anomalies. *Anomaly filtering* is out of the scope of this paper but it is certainly clear that if anomalous samples in specified time-frequency windows have been identified, then they can also be removed if proper reconstruction techniques are observed.

We define the *attention span* δ describes the length of the sample window required for the described anomaly detection method. For the attention span, we note that the robust estimates in that period describe the distribution parameters of the *majority* of samples. Therefore, for large δ we note an insensitivity to local outliers. Small δ results in a hypersensitivity to outliers and is not useful. In other words, greater attention occurs in short periods, while less attention occurs over long periods.

B. System Stress

Invariably over time, the robust statistical properties of an audio signal will change for differing acoustic scenes implying that the scene *context* has changed. Tracking these variations over time can give a qualitative measure about the degree of difficulty an observer will have in detecting an anomaly in any sub-band. We call this measure of difficulty the system stress. On the assumption that a change in the current context implies that an observer will have difficulty to adapt, we a attempt to follow changes in the current context. We expect that for a for a buffer Δ of past contexts, there is a mean context $\vec{\Xi}_\Delta$ that is subject to some variation Θ_Δ. We can then compute the *system stress* as:

$$\mathrm{S}(\vec{C}) = \sqrt{(\vec{C} - \vec{\Xi}_\Delta)\Theta_\Delta^{-1}(\vec{C} - \vec{\Xi}_\Delta)^T} \qquad (7)$$

Where $\vec{C}$ is the current context containing the robust mean in units of the robust standard deviation for each of the N_s sub-bands:

$$\vec{C} = \left[\frac{\tilde{\mu}_1}{\tilde{\sigma}_1}\ \frac{\tilde{\mu}_2}{\tilde{\sigma}_2}\ \cdots\ \frac{\tilde{\mu}_{N_s}}{\tilde{\sigma}_{N_s}}\right]^T \qquad (8)$$

Also, for the past M context observations in the buffer Δ, we define $\vec{\Xi}_\Delta$ and Θ_Δ as the *unbiased Gaussian maximum-likelihood estimates* for the mean and covariance respectively. The decision criteria for context change is resolved by first determining an acceptable threshold for $\mathrm{S}(\vec{C})$.

C. Major Anomalies

While minor anomalies can reveal interesting features of a signal in its local scope, its importance as an uncharacteristic event over a brief history is determined by the system stress in that period. If the system is sufficiently stressed, then we know that the minor anomalies are related to context change. A minor anomaly that has appeared during a context change is considered as a *major anomaly*. Since in the context of this work, only major anomalies are localized, the threshold criteria for context change, is the same criteria for localization.

The successive transformation of random variable in each audio sub-band resulted in the target Gaussian distribution. The wavelet decomposition into multi-resolution space resulted in robust estimates that form the components of $\vec{C}$. We assert that the stress $\mathrm{S}(\vec{C})$ is approximately Chi distributed. As such, given a probability p that a context change has occurred in the buffer Δ, we compute the threshold $T_c = \chi_{1-p}^{-1}(N_s)$. Minor anomalies that occur while $\mathrm{S}(\vec{C}) \geq T_c$, are considered as major anomalies and should be spatially localized.

Figure 3 shows an example where the system stress was monitored for an audio sample of a fireworks event. Over fifteen explosions of varying intensities are heard in the presence

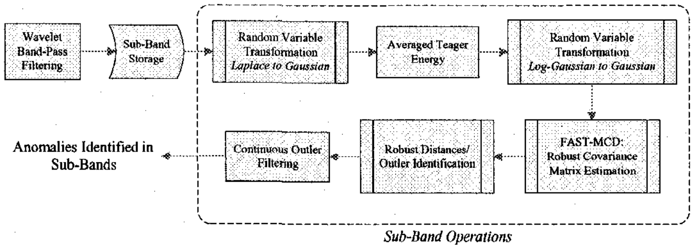

Wavelet Band-Pass Filtering
Sub-Band Storage
Random Variable Transformation *Laplace to Gaussian*
Averaged Teager Energy
Random Variable Transformation *Log-Gaussian to Gaussian*
FAST-MCD: Robust Covariance Matrix Estimation
Robust Distances/ Outlier Identification
Continuous Outler Filtering
Anomalies Identified in Sub-Bands
Sub-Band Operations

Fig. 2. The minor anomaly detection process.

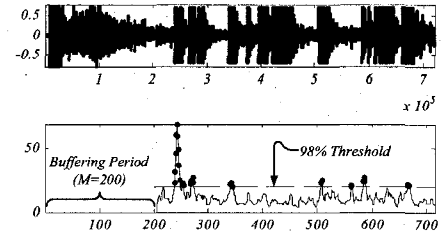


Fig. 3. The upper plot contains the normalized audio intensity of a festive fireworks event. The markers in the lower plot of $\mathrm{S}(\vec{C})$ denote only where major anomalies have occurred: onset of multiple explosions, spectators suddenly cheering, or launching of fireworks.

of other activities including various sounds of the festive event. The markers denote where $\mathrm{S}(\vec{C}) \geq T_c$, which subjectively coincides with the onset of firework blasts, the addition of new spectator vocalizations or sounds from fireworks launches that have not yet resulted in a blast. In this particular case, all of the markers indicate when major anomalies have occurred.

D. Anomaly Localization

The trivial extraction of spatial information from a two sensor array takes advantage of the fact that an energy wavefront emanating from a far-field source will not pass through each of the sensors at exactly the same time because of the medium in which it travels. This delay can be estimated from the cross-spectrum phase of the sensor data whose time domain counterpart is the the cross-correlation function. A major anomaly can be localized using each of the two channels a and b over the attention span δ where the relative number of anomalous samples in each sub-band has exceeded some threshold (say 50%). We estimate the cross-correlation function $\widehat{R}_i(\tau)$ using only these sub-bands from each channel $s_{i,a}$ and $s_{i,b}$ for a time window of width $T = \delta$, centered at time t:

$$\widehat{R}_i(\tau) = \frac{1}{T} \int_{t-\frac{T}{2}}^{t+\frac{T}{2}} s_{i,a}(u) s_{i,b}(u+\tau) du \tag{9}$$

The location τ_d of the single peak in $\widehat{R}_i(\tau)$ allows us to estimate the physical azimuth α_i of the anomaly in each sub-band i: $\cos \alpha_i = \frac{c\tau_d}{d_{mic}}$, where c is the speed of sound and d_{mic} is the physical distance between a stereo microphone pair [4]. The *median* of the α_i estimates is used to finally localize the anomaly.

V. Conclusion

The modulating source assumption allows us to specify the problem of anomaly detection as a problem in robust statistics. The demodulation property of the Teager energy operator is used to jointly expose amplitude and frequency features of audio sub-bands. Spectral decomposition is achieved using a scaled wavelet function acting as a band-pass filter which halves its bandwidth for each decomposition level. Through successive transformation of variable, Teager energy of Laplace distributed sub-bands are redistributed into a Gaussian distribution for which parameters are estimated using the FAST-MCD algorithm. The robust Mahalanobis distances are used in the identification of minor anomalies. Tracked over a brief period, if an uncharacteristic change is detected in the context vector containing the robust sub-band means in units of the robust standard deviations, then the minor anomalies are relabeled as major anomalies and are used in a cross-correlation estimate from both channels of a two sensor array. The median of the resulting delay estimates in each sub-band provide *the localization* for the major *anomaly*.

References

[1] E. Kvedalen, "Signal processing using the teager energy operator and other nonlinear operators," Master's thesis, University of Oslo, May 2003.

[2] P. J. Rousseeuw and K. V. Driessen, "A fast algorithm for the minimum covariance determinant estimator," in *Technometrics*. American Society for Quality Control and American Statistical Association, 1999, pp. 212 – 223.

[3] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 1991.

[4] B. C. Kirkwood, "Acoustic source localization using time-delay estimation," Master's thesis, Technical University of Denmark, 2003.

[5] E. Grassi and S. A. Shamma, "A biologically inspired, learning, sound localization algorithm," in *2001 Conference on Information Sciences and Systems*. University of Maryland, College Park, MD 20742, USA: The Johns Hopkins University, March 21-23 2001, pp. 344–348.

[6] *Finding Events automatically in Continuously Sampled Data Streams via Anomaly Detection*. IEEE National Aerospace & Electronics Conference (NAECON), October 2000.

Appendix B

Matlab Source Code

The Matlab mathematical programming environment was used to help develop the methodology described in this work. The object-oriented methodology was used for major components with their private and public methods listed here. Examining the class constructor reveals all properties which are *private* unless an accessor function is provided.

B.1 Matlab Object: @Recorder

B.1.1 Public Methods

display.m

```
function display(c)
% This function will display the critical parts of the Recorder object.

fprintf(1,'RECORDER Object (16 bit stereo)\n');
fprintf(1,'Sampling Rate        = %6d Hz\n',c.Sampling_Rate);
fprintf(1,'Sample Length        = %6d s\n',c.Sample_Length);
if c.Azimuth_Beam_Angle == 0 & c.Microphone_Spacing ==0
    fprintf(1,'Azimuth Beamforming =    OFF\n');
else
    fprintf(1,'Azimuth Beamforming = %6d degrees, %2.1f cm separation.\n',c.Azimuth_Beam_Angle,c.Microphone_Spacing);
end;
```

Get_Azimuth_Beam_Parameters.m

```
function [Angle, Separation] = Get_Azimuth_Beam_Parameters(obj)

Angle = obj.Azimuth_Beam_Angle;
Separation = obj.Microphone_Separation;
```

Get_Normalization_Mode.m

```
function Mode = Get_Normalization_Mode(obj)
Mode = obj.Normalization_Mode;
```

Get_Sample_Length.m

```
function Length = Get_Sample_Length(obj)
Length = obj.Sample_Length;
```

Get_Sampling_Rate.m

```
function Rate = Get_Sampling_Rate(obj)
Rate = obj.Sampling_Rate;
```

Get_Sound_Data.m

```
function Data = Get_Sound_Data(obj)

if obj.Azimuth_Beam_Angle==0 && obj.Microphone_Spacing == 0
    Data =obj.Sound_Data;
else
    Data =Beamform_Data(obj);
end;
```

Inject_File.m

```
function Inject_File(obj, Filename)

[obj.Sound_Data,obj.Sampling_Rate,obj.Bit_Resolution]=wavread(Filename);
obj.Sample_Length = round(max(size(obj.Sound_Data))/obj.Sampling_Rate);
assignin('caller', inputname(1), obj);
```

kill.m

```
function kill(c)
clear c;
```

Play.m

```
function Play(obj)

if obj.Azimuth_Beam_Angle==0 && obj.Microphone_Spacing == 0
    disp('Original buffer playing...');
    wavplay(obj.Sound_Data,obj.Sampling_Rate);
else
    disp('Beamformed buffer playing...');
    wavplay(Beamform_Data(obj),obj.Sampling_Rate);
end;
```

Recorder.m

```
function x = Recorder(c)
% This is the constructor for the Recorder object.
if nargin == 0
    x.Sampling_Rate = 44100;
    x.Sample_Length = 1;
    x.Bit_Resolution = 16;
    x.Azimuth_Beam_Angle = 0;
    x.Microphone_Spacing = 0;
    x.Microphone = audiorecorder(x.Sampling_Rate,x.Bit_Resolution,2);
    x.Sound_Data = [];
    x = class(x,'Recorder');
elseif isa(c,'Recorder')
    x=c;
end
```

Set_Azimuth_Beam_Parameters.m

```
function Set_Azimuth_Beam_Parameters(obj,Angle,Separation)

if Angle>90 | Angle <-90
    error('RECORDER: Angle must be between -90 and 90 degrees.');
else
    obj.Azimuth_Beam_Angle = Angle;
end;
if Separation<0 | Separation > 100
    error('RECORDER: Microphone separation must be between 0cm and 100cm.');
else
    obj.Microphone_Spacing = Separation;
end;

assignin('caller', inputname(1), obj);
```

Set_Normalization_Mode.m

```
function Set_Normalization_Mode(obj,Mode)

if strcmpi(Mode,'ON') | Mode == 1
    obj.Normalization_Mode = 1;
    assignin('caller',inputname(1),obj);
elseif strcmpi(Mode,'OFF') | Mode == 0
    obj.Normalization_Mode = 0';
    assignin('caller',inputname(1),obj);
else
    error('Verbose mode can either be ON(1) or OFF(0).');
end;
```

Set_Sample_Length.m

```
function Set_Sample_Length(obj,Length)
```

```
obj.Sample_Length = Length;
obj.Microphone = audiorecorder(obj.Sampling_Rate,obj.Bit_Resolution,2);
assignin('caller', inputname(1), obj);
```

Set_Sampling_Rate.m

```
function Set_Sampling_Rate(obj,Rate)
obj.Sampling_Rate = Rate;
obj.Microphone = audiorecorder(obj.Sampling_Rate,obj.Bit_Resolution,2);
assignin('caller', inputname(1), obj);
```

Start_Recording.m

```
function Start_Recording(c)
recordblocking(c.Microphone,c.Sample_Length);
c.Sound_Data = getaudiodata(c.Microphone,'double');
assignin('caller',inputname(1),c);
```

B.1.2 Private Methods

Azimuth_Beam_Angle_To_Delay.m

```
function Delay = Azimuth_Beam_Angle_To_Delay(Angle,Microphone_Spacing)

% The angle in degrees and microphone separation in centimeters is
% converted to a delay in seconds.

% Speed of sound is 34000cm/s
c = 34000;
Delay = Microphone_Spacing*sind(Angle)/c;
```

Beamform_Data.m

```
function Data = Beamform_Data(obj)

Sample_Delay = floor(obj.Sampling_Rate*Azimuth_Beam_Angle_To_Delay(obj.Azimuth_Beam_Angle,obj.Microphone_Spacing));
size(obj.Sound_Data);
if obj.Azimuth_Beam_Angle>0
    Data = (obj.Sound_Data(1:end-Sample_Delay,1)+obj.Sound_Data(1+Sample_Delay:end,2))/2;
else
    Data = (obj.Sound_Data(1:end-Sample_Delay,2)+obj.Sound_Data(1+Sample_Delay:end,1))/2;
end;
```

B.2 Matlab Object: @Robot

B.2.1 Public Methods

Autodetect.m

```
function [Port,Controller_ID] = Autodetect(Robot,Hardware_Controller_Version_String)
% This function will autodetect which COM port the robot is on.

% Controller identifier.
% Hardware_Controller_Version_String = 'SV203 V1.2';
Hardware_Controller_Version_String_Querry = '';

% Find all available COM ports.
Hardware_Scan = instrhwinfo('serial');
Number_Of_Available_COM_Ports = size(Hardware_Scan.AvailableSerialPorts,1);
Port_Found = 0; % No port was found yet.

% Perform scan.
for COM = 1:Number_Of_Available_COM_Ports
    Robot.Serial_Port = serial(...
        Hardware_Scan.AvailableSerialPorts(COM),...
        'BaudRate',9600,...
        'DataBits',8,...
        'Parity','none',...
        'StopBits',1,...
        'Terminator',13,...
        'Timeout',2);

    % Open the serial port.
    fopen(Robot.Serial_Port);
    fprintf(Robot.Serial_Port,'V?');
    [Hardware_Controller_Version_String_Querry,Byte_Count,Message] = fgetl(Robot.Serial_Port);

    % Check to see if the controller's power is off.  In this case, the
    % read will fail with zero bytes read.
    if(Byte_Count==0)
        Port = char(Hardware_Scan.AvailableSerialPorts(COM));
        Controller_ID = 'Controller power may be off.';
        fclose(Robot.Serial_Port);
        assignin('caller',inputname(1),Robot);
        return;
    end;

    % Check to see if the controller is present.
    if(strcmp(Hardware_Controller_Version_String,Hardware_Controller_Version_String_Querry))
        Port_Found = 1;
        Port = char(Hardware_Scan.AvailableSerialPorts(COM));
        Controller_ID = Hardware_Controller_Version_String_Querry;
    end;
    fclose(Robot.Serial_Port);
end;
assignin('caller',inputname(1),Robot);
if ¬Port_Found
```

```
    Port = 'None found';
    Controller_ID = 'Unassigned.';
end;
```

Destroy.m

```
function Destroy(Robot)
% This function is responsible for destroying the Robot object.

fclose(Robot.Serial_Port);
```

DISPLAY.M

```
function display(Robot)
% This function will display the critical parts of the Recorder object.

fprintf(1,'ROBOT Object (2 DOF)\n--------------------\n');

fprintf(1,'Communication Port:                  %s (%s)\n',Robot.COM_Port_ID,Robot.Serial_Port.status);
fprintf(1,'Controller ID:                       %s\n\n',Robot.Controller_ID);

fprintf(1,'Microphone Separation (Angle):       %2.1f cm (%3.1f)\n',...
    Robot.Physical_Microphone_Spacing,...
    Robot.Physical_Arm_Angle);

fprintf(1,'Array Direction:                    %3.1f\n',...
    Robot.Physical_Azimuth_Angle);
```

Relax.m

```
function Relax(Robot)
% This function is called to remove power from both motors.

Send_Command(Robot,'SV1 M0 D100 SV2 M0 D100');
```

Robot.m

```
function x = Robot(Arg1)
% This is the constructor for the Robot object.

% This property is to identify the hardware.
x.Controller_ID = 0;
x.COM_Port_ID = '';

% These properties are for the angular control of the microphone spacing.
% The two values are used for calibration.
x.Physical_Arm_Angle = 90; %Degrees
x.Physical_Arm_Length = 27.0; %Centimeters
x.Physical_Arm_Offset = 3.7; %Centimeters
x.Physical_Microphone_Spacing = x.Physical_Arm_Offset+...
    2*x.Physical_Arm_Length*sin(deg2rad(x.Physical_Arm_Angle)); %Centimeters
x.MIN_Logical_Arm_Angle = 23; %22
x.MAX_Logical_Arm_Angle = 138;

```

```
% These properties are for the azimuth control of the microphone
% assembly. The two values are used for calibration.
x.Physical_Azimuth_Angle = 180;
x.MIN_Logical_Azimuth = 2;
x.MAX_Logical_Azimuth = 228;

switch nargin
            % The serial port is required in order to work with the robot's
            % controller. Since no arguements are specified, use COM1 as the
            % default serial port.
    case 0
            x.Serial_Port = serial('COM1');
            x = class(x,'Robot');
            [x.COM_Port_ID, x.Controller_ID] = Autodetect(x,'SV203 V1.2');
            x.Serial_Port = serial(x.COM_Port_ID,'BaudRate',9600,...
                'DataBits',8,'Parity','none','StopBits',1,'Terminator',13);
            % Open the communication port.
            fopen(x.Serial_Port);

            % Initialize the robot.
            Set_Microphone_Separation(x,0);
            pause(1);
            Set_Array_Direction(x,0);

    case 1
            if(isa(Arg1,'Robot'))
                disp('Case 1: Robot Class');
                x=Arg1;
            elseif isa(Arg1,'char')
                disp('Case 1: Character Class');
                x.Serial_Port = serial(Arg1,'BaudRate',9600,'DataBits',...
                    8,'Parity','none','StopBits',1,'Terminator',13);
                fopen(x.Serial_Port);
                x = class(x,'Robot');
            end;
end;

%if nargin == 0
%
%elseif isa(c,'Robot')
%    x=r;
%end
```

Send_Command.m

```
function Send_Command(Robot,Command)
% This function will send a command to the motor controller.

% The range of the Futuba motor is from 2 to 228.
% The range of the Hitec motor is from 22 to 138.

% Create the command string with line termination.
Command_String = sprintf('%s%c',Command,13);

% Send the command string to the controller.
```

```
fprintf(Robot.Serial_Port,'%s',Command_String);
```

Set_Array_Direction.m

```
function Set_Array_Direction(Robot,Angle)
% This function will take in a value for the array direction in degrees
% and adjust the robot to match. (-90 to +90 degrees)

Angle = Angle + 90;

% Map the normalized angle to the logical range.
Motor_Control_Steps = 180/(Robot.MAX_Logical_Azimuth - Robot.MIN_Logical_Azimuth);
Motor_Position = round(Robot.MIN_Logical_Azimuth + Angle/Motor_Control_Steps);

% Compute the physical position of the arm (corrected to the resolution of
% the controller.
Robot.Physical_Azimuth_Angle = 180*(Motor_Position-Robot.MIN_Logical_Azimuth)/...
    (Robot.MAX_Logical_Azimuth - Robot.MIN_Logical_Azimuth);

% Send the command to the controller.  Motor #1 is connected to the array.
Command_String = sprintf('SV1 M%d D800 M0',Motor_Position);
Send_Command(Robot,Command_String);

% Save all changed values to the class.
assignin('caller',inputname(1),Robot);
```

Set_Microphone_Angle.m

```
function Set_Microphone_Angle(Robot,Angle)
% This function will take in a value for the microphone separation in
% centimeters and adjust the robot to match.

% Map the normalized angle to the logical range.
Motor_Control_Steps = 90/(Robot.MAX_Logical_Arm_Angle - Robot.MIN_Logical_Arm_Angle);
Motor_Position = round(Robot.MIN_Logical_Arm_Angle + Angle/Motor_Control_Steps);

% Compute the physical position of the arm (corrected to the resolution of
% the controller.
Robot.Physical_Arm_Angle = 90*(Motor_Position-Robot.MIN_Logical_Arm_Angle)/...
    (Robot.MAX_Logical_Arm_Angle - Robot.MIN_Logical_Arm_Angle);
Robot.Physical_Microphone_Spacing = Robot.Physical_Arm_Offset + ...
    2*Robot.Physical_Arm_Length*sin(deg2rad(Robot.Physical_Arm_Angle)); %Centimeters

% Send the command to the controller.  Motor #2 is connected to the arm.
Command_String = sprintf('SV2 M%d D2000 M0',Motor_Position);
Send_Command(Robot,Command_String);

% Save all changed values to the class.
assignin('caller',inputname(1),Robot);
```

Set_Microphone_Separation.m

```
function Set_Microphone_Separation(Robot,Spacing)
% This function will take in a value for the microphone separation in
% centimeters and adjust the robot to match.

Robot.Physical_Microphone_Spacing = Spacing - Robot.Physical_Arm_Offset;
Robot.Physical_Arm_Angle = rad2deg(asin(Robot.Physical_Microphone_Spacing/(2*Robot.Physical_Arm_Length)));
Set_Microphone_Angle(Robot,Robot.Physical_Arm_Angle);

assignin('caller',inputname(1),Robot);
```

B.3 Matlab Object: @Sub_Band_Anomaly_Detector

B.3.1 Public Methods

Analyze.m

```
function Analysis_Results = Analyze(obj)

obj.Sub_Band_Information = Load_Data(obj,obj.Sample_Data);
Analysis_Results = obj.Sub_Band_Information;

% Save the data to the class.
assignin('caller',inputname(1),obj);
```

display.m

```
function display(obj)
% This function is responsible for displaying the object correctly.

fprintf(1,'\n* Sub-Band Anomaly Detector Object\n ----------------------------------\n');
% Display the appropriate output for the sample size
[m,n]=size(obj.Sample_Data);
if(m==0 || n==0)
    fprintf(1,'  - Sample Size:\t\t\t\t-\n');
else
    fprintf(1,'  - Sample Size:\t\t\t\t%d\n',n);
end;

fprintf(1,'  - Wavelet Basis:\t\t\t\t%s\n',obj.Wavelet_Basis);

if obj.Maximum_Decomposition
    fprintf(1,'  - Analysis Level:\t\t\t\tMaximum\n');
else
    fprintf(1,'  - Analysis Level:\t\t\t\t%d\n',obj.Analysis_Decomposition_Level);
end;
fprintf(1,'  - Averaging Window Size:\t\t%d\n',obj.Averaging_Window_Size);
if obj.Reduce_Edge_Effects
    fprintf(1,'  - Edge Effect Reduction:\t\ttrue\n');
else
    fprintf(1,'  - Edge Effect Reduction:\t\tfalse\n');
end;

```

```
if obj.Laplacian_Sample_Filter
    fprintf(1,'  — Laplacian Sample Filter:\ton\n');
else
    fprintf(1,'  — Laplacian Sample Filter:\toff\n');
end;
fprintf(1,'  — Minimum Anomaly Length:\t\t%d\n',obj.Minimum_Anomaly_Length);
```

Extract_Feature.m

```
function Feature = Extract_Feature(obj)
% This function will extract the feature vector from the data set.

% If the requested decomposition level is greater than the maximum, return
% an error (don't execute this function).
if obj.Analysis_Decomposition_Level > obj.Maximum_Decomposition_Level;
    obj
    error('The requested level of decomposition is greater than the maximum.\n');
end;


Feature = obj.Extraction_Feature;
switch(obj.Extraction_Feature)
    % Features from the approximation only.
    case 'Approximation Mean'
        Feature = Extract_Feature__Approximation_Mean(obj);
    case 'Approximation Variance'
        Feature = Extract_Feature__Approximation_Variance(obj);
    case 'Approximation Spread'
        Feature = Extract_Feature__Approximation_Mean(obj)./Extract_Feature__Approximation_Variance(obj);
    case 'Approximation Energy'
        Feature = Extract_Feature__Approximation_Energy(obj);
    case 'Approximation Entropy'
        Feature = Extract_Feature__Approximation_Entropy(obj);
    case 'Approximation Trend'
        Feature = Extract_Feature__Approximation_Trend(obj);
    case 'Teager Energy'
        Feature = Extract_Feature__Teager_Energy(obj);
    case 'Entropy'
    case 'Information'
    case 'LPC1'
    otherwise
end;
```

get.m

```
function val = get(obj,Property_Name)
% This function is the function returns the value of the specified property name.

switch(Property_Name)
    case 'Sub_Band_Information'
        val = obj.Sub_Band_Information;
    otherwise
        error([Property_Name,' is not a valid Feature_Extractor property.']);
end;
```

Load_Data.m

```
function Analysis_Result = Load_Data(obj,Input_Data)
% This function will load the input data into the object.

% Ensure that the data is a simple row vector.
[m,n]=size(Input_Data);
Sample_Length = n;
if (m≠1 && n≠1)
    fprintf(1,'ERROR: Single dimension vectors only.  Sample not loaded.\n');
    return;
else
    if(n==1)
        % Convert the column vector to a row vector.
        Input_Data=Input_Data';
        Sample_Length = m;
    end;
    obj.Sample_Data=Input_Data;
end;

% Compute the maximum decompostion level for this sample.
if obj.Maximum_Decomposition
    obj.Analysis_Decomposition_Level=wmaxlev(Sample_Length,obj.Wavelet_Basis);
end

% Call the sub-band outlier detector.
Analysis_Result = Outlier_Detector(obj);
obj.Sub_Band_Information = Analysis_Result;

% Save the data to the class.
assignin('caller',inputname(1),obj);
```

set.m

```
function set(obj,varargin)
% This function is the function sets the value of the specified property name.

Property_List = varargin;
while length(Property_List) >= 2,
    Property = Property_List{1};
    Value = Property_List{2};
    Property_List = Property_List(3:end);
    switch(Property)
        case 'Wavelet_Basis'
            obj.Wavelet_Basis = Value;
%          case 'Sampling_Method'
%              obj.Sampling = Value;
        case 'Analysis_Level'
            obj.Analysis_Decomposition_Level = Value;
            if size(obj.Sample_Data,2)>0
                Max_Decomp=wmaxlev(size(obj.Sample_Data,2),obj.Wavelet_Basis);
                if Value<Max_Decomp
                    obj.Maximum_Decomposition = false;
                else
                    obj.Maximum_Decomposition = true;
                    obj.Analysis_Decomposition_Level = Max_Decomp;
                end;
            end;
        case 'Averaging_Window_Size'
            obj.Averaging_Window_Size = Value;
        case 'Maximum_Decomposition'
            if strcmpi(Value,'true')|strcmpi(Value,'on')|Value==1
                obj.Maximum_Decomposition = true;
            else
                if strcmpi(Value,'false')|strcmpi(Value,'off')|Value==1Value==0
                    obj.Maximum_Decomposition = false;
                else
                    if islogical(Value)
                        obj.Maximum_Decomposition=Value;
                    else
                        error('Maximum_Decomposition can only take on values: {on|off}, {true|false} or {0|1}');
                    end;
                end;
            end;
        case 'Edge_Effect_Reduction'
            if strcmpi(Value,'true')|strcmpi(Value,'on')|Value==1
                obj.Reduce_Edge_Effects = true;
            else
                if strcmpi(Value,'false')|strcmpi(Value,'off')|Value==0
                    obj.Reduce_Edge_Effects = false;
                else
                     if islogical(Value)
                        obj.obj.Reduce_Edge_Effects=Value;
                     else
                        error('Edge_Effect_Reduction can only take on values: {on|off}, {true|false} or {0|1}');
                     end;
                end;
            end;
        case 'Laplacian_Sample_Filter'
            if strcmpi(Value,'true')|strcmpi(Value,'on')|Value==1
                obj.Laplacian_Sample_Filter = true;
            else
                if strcmpi(Value,'false')|strcmpi(Value,'off')|Value==0
                    obj.Laplacian_Sample_Filter = false;
                else
                    error('Laplacian_Sample_Filter can only take on values: {on|off}, {true|false} or {0|1}');
                end;
            end;
        case 'Averaging_Window_Size'
            obj.Averaging_Window_Size = Value;
        case 'Minimum_Anomaly_Length'
            if Value<1
                error('The minimum anomaly length is one sample');
            else
                obj.Minimum_Anomaly_Length = Value;
            end;
        otherwise
            error('Unrecognized property.');
    end;
end;

% Save the data to the class.
```

```
assignin('caller',inputname(1),obj);
```

subsref.m

```
function value = subsref(obj,index)

% Check the depth of the references requested.
Number_Of_References=size(index,2);
if Number_Of_References>2
    error('??? Too many references.  Not implemented.');
end;

% Check the first reference and subsequent references if requested and
% allowed.
switch index(1).type
    case '{}'
        % Sub-Band Cell-Structure References
        % ------------------------------------------------
        % When referenced in this mode, all of the gathered sub-band
        % information is returned for the specified decomposition level.
        if index(1).subs{1}==0
            value=size(obj.Sub_Band_Information,2);
        else
            if (Number_Of_References==2) & index(2).type=='.'
                Object=obj.Sub_Band_Information{index(1).subs{1}};
                value=eval(['Object.' index(2).subs]);
            else
                value=obj.Sub_Band_Information{index(1).subs{1}};
            end;
        end;
    case '()'
        % Sub-Band Array References
        % ------------------------------------
        % When referenced in this mode, only the band-pass signals are
        % returned at the specified decompostion levels.
        if index(1).subs{1}==0
            % All of the samples are the same size, use the first for
            % measurement of the array.
            value=size(obj.Sub_Band_Information{1}.Data,2);
        else
            value=obj.Sub_Band_Information{index(1).subs{1}}.Data;
        end;
    case '.'
        % Object Property References
        % ------------------------------------
        % When referenced in this mode, the value of the object's specified
        % reference is returned.
        switch index(1).subs
            case 'Sample_Size'
                value=size(obj.Sample_Data,2);
            case 'Wavelet_Basis'
                value=obj.Wavelet_Basis;
            case 'Analysis_Level'
                value=obj.Analysis_Decomposition_Level;
            case 'Averaging_Window_Size'
```

```
                value=obj.Averaging_Window_Size;
            case 'Edge_Effect_Reduction'
                value=obj.Reduce_Edge_Effects;
            case 'Laplacian_Sample_Filter'
                value=obj.Laplacian_Sample_Filter;
            otherwise
                error('??? Attempt to reference a non-existant field.');
        end;
    otherwise
        % Unknown Property References
        % ---------------------------
        % This is an error condition only.
        error('??? Unknown reference type.')
end;
```

Sub_Band_Anomaly_Detector.m

```
function Obj = Sub_Band_Anomaly_Detector(Obj)
% This function is the constructor for the Sub_Band_Anomaly_Detector object.

if nargin==0
    % Main object property initialization
    Obj.Wavelet_Basis = 'dmey';
    Obj.Analysis_Decomposition_Level = 0;
    Obj.Maximum_Decomposition = true;
    Obj.Sample_Data = [];

    % Properties specific for the outlier detector.
    Obj.Averaging_Window_Size = 20;
    Obj.Reduce_Edge_Effects = true;
    Obj.Laplacian_Sample_Filter = true;
    Obj.Minimum_Anomaly_Length = 1;

    % Analysis results.
    Obj.Sub_Band_Information = {};
    Obj = class(Obj,'Sub_Band_Anomaly_Detector');
else
    if isa(x,'Sub_Band_Anomaly_Detector')
        Obj = x;
    end;
end;
```

B.3.2 Private Methods

Lap2Gauss.m

```
function  y = Lap2Gauss(x)
% y = Lap2Gaus(x)
% ---------------
%
% This function will convert a data set that follows a Laplacian
% distribution to a normal, Gaussian distribution.
```

```
%


%% Estimate the Laplace distribution parameters for the sample.
N=max(size(x));
mu=median(x);
b=1/N*sum(abs(x—mu));
sigma_2=2*b^2;
sigma=sqrt(sigma_2);
sqrt_2=sqrt(2);

%% Convert the data set to a Gaussian distribution by passing it through a
%% non—linearity.
y = erfinv(   sign(x—mu).* (1—exp(—abs(x—mu)/b))  )*sigma*sqrt_2+mu;
```

modifiedfastmcd2.m

```
function [res,raw]=modifiedfastmcd2(data,options);

% version 22/12/2000, revised 19/01/2001,
% new reweighted correction factors and old cutoff 9/07/2001
% last revision 20/04/2006
%
% FASTMCD computes the MCD estimator of a multivariate data set.  This
% estimator is given by the subset of h observations with smallest covariance
% determinant.  The MCD location estimate is then the mean of those h points,
% and the MCD scatter estimate is their covariance matrix.  The default value
% of h is roughly 0.75n (where n is the total number of observations), but the
% user may choose each value between n/2 and n.
%
% The MCD method is intended for continuous variables, and assumes that
% the number of observations n is at least 5 times the number of variables p.
% If p is too large relative to n, it would be better to first reduce
% p by variable selection or principal components.
%
% The MCD method was introduced in:
%
%   Rousseeuw, P.J. (1984), "Least Median of Squares Regression,"
%   Journal of the American Statistical Association, Vol. 79, pp. 871—881.
%
% The MCD is a robust method in the sense that the estimates are not unduly
% influenced by outliers in the data, even if there are many outliers.
% Due to the MCD's robustness, we can detect outliers by their large
% robust distances. The latter are defined like the usual Mahalanobis
% distance, but based on the MCD location estimate and scatter matrix
% (instead of the nonrobust sample mean and covariance matrix).
%
% The FASTMCD algorithm uses several time—saving techniques which
% make it available as a routine tool to analyze data sets with large n,
% and to detect deviating substructures in them. A full description of the
% algorithm can be found in:
%
%   Rousseeuw, P.J. and Van Driessen, K. (1999), "A Fast Algorithm for the
%   Minimum Covariance Determinant Estimator," Technometrics, 41, pp. 212—223.
%
% An important feature of the FASTMCD algorithm is that it allows for exact
% fit situations, i.e. when more than h observations lie on a (hyper)plane.
% Then the program still yields the MCD location and scatter matrix, the latter
% being singular (as it should be), as well as the equation of the hyperplane.
%
% Usage:
%   [res,raw]=fastmcd(data,options)
%
% If only one output argument is listed, only the final result ('res') is returned.
% The first input argument 'data' is a vector or matrix.  Columns represent
% variables, and rows represent observations.  Missing values (NaN's) and
% infinite values (Inf's) are allowed, since observations (rows) with missing
% or infinite values will automatically be excluded from the computations.
%
% The second input argument 'options' is a structure.  It specifies certain
% parameters of the algorithm:
%
%  options.cor: If non-zero, the robust correlation matrix will be
%               returned. The default value is 0.
%  options.alpha: The percentage of observations whose covariance determinant will
%                 be minimized.  Any value between 0.5 and 1 may be specified.
%                 The default value is 0.75.
%  options.ntrial: The number of random trial subsamples that are drawn for
%                  large datasets. The default is 500.
%
% The output structure 'raw' contains intermediate results, with the following
% fields :
%
%  raw.center: The raw MCD location of the data.
%  raw.cov: The raw MCD covariance matrix (multiplied by a finite sample
%           correction factor and an asymptotic consistency factor).
%  raw.cor: The raw MCD correlation matrix, if options.cor was non-zero.
%  raw.objective: The determinant of the raw MCD covariance matrix.
%  raw.robdist: The distance of each observation from the raw MCD location
%               of the data, relative to the raw MCD scatter matrix 'raw.cov'
%  raw.wt: Weights based on the estimated raw covariance matrix 'raw.cov' and
%          the estimated raw location of the data. These weights determine
%          which observations are used to compute the final MCD estimates.
%
% The output structure 'res' contains the final results, namely:
%
%  res.n_obs: The number of data observations (without missing values).
%  res.quan: The number of observations that have determined the MCD estimator,
%            i.e. the value of h.
%  res.mahalanobis:  The distance of each observation from the classical
%                    center of the data, relative to the classical shape
%                    of the data. Often, outlying points fail to have a
%                    large Mahalanobis distance because of the masking
%                    effect.
%  res.center: The robust location of the data, obtained after reweighting, if
%              the raw MCD is not singular.  Otherwise the raw MCD center is
%              given here.
%  res.cov: The robust covariance matrix, obtained after reweighting and
%           multiplying with a finite sample correction factor and an asymptotic
%           consistency factor, if the raw MCD is not singular.  Otherwise the
%           raw MCD covariance matrix is given here.
%  res.cor: The robust correlation matrix, obtained after reweighting, if
%          options.cor was non-zero.
%  res.method: A character string containing information about the method and
%              about singular subsamples (if any).
%  res.robdist:  The distance of each observation to the final,
%                reweighted MCD center of the data, relative to the
%                reweighted MCD scatter of the data.  These distances allow
%                us to easily identify the outliers. If the reweighted MCD
%                is singular, raw.robdist is given here.
%  res.flag: Flags based on the reweighted covariance matrix and the
%            reweighted location of the data.  These flags determine which
%            observations can be considered as outliers. If the reweighted
%            MCD is singular, raw.wt is given here.
%  res.plane:  In case of an exact fit, res.plane contains the coefficients
%              of a (hyper)plane a_1(x_i1-m_1)+...+a_p(x_ip-m_p)=0
%              containing at least h observations, where (m_1,...,m_p)
%              is the MCD location of these observations.
%  res.X: The data matrix. Rows containing missing or infinite values are
%         ommitted.
%
% FASTMCD also automatically calls the function PLOTMCD which creates plots for
% visualizing outliers detected by FASTMCD. The plots that can be produced are:
%
% 1. Plot of Mahalanobis distances versus case number.
% 2. Plot of robust distances versus case number.
% 3. QQ plot: shows robust distances versus chi-squared quantiles.
% 4. Robust distances versus Mahalanobis distances (i.e. the D-D plot).
% 5. The 97.5% robust tolerance ellipse (if the dataset is bivariate).
%
% Usage:
%     plotmcd(mcdres,options)
%
% The first input argument 'mcdres' is the output argument of the function
% FASTMCD. The second input argument 'options' is a structure containing:
%
%  options.ask: A logical flag: if set to 0, all plots are produced sequentially;
%               if set to 1, PLOTMCD displays a menu listing all the plots that
%               can be produced. The default value is 1.
%  options.nid: Number of points (must be less than n) to be highlighted in the
%               appropriate plots. These will be the 'nid' most extreme points,
%               i.e. those with largest robust distance.
%  options.xlab: Label of the X-axis in the MCD tolerance ellipse plot.
%  options.ylab: Label of the Y-axis in the MCD tolerance ellipse plot.


% The fastmcd algorithm works as follows:
%
%       The dataset contains n cases and p variables.
%       When n < 2*nmini (see below), the algorithm analyzes the dataset as a whole.
%       When n >= 2*nmini (see below), the algorithm uses several subdatasets.
%
%       When the dataset is analyzed as a whole, a trial subsample of p+1 cases
%       is taken, of which the mean and covariance matrix are calculated.
%       The h cases with smallest relative distances are used to calculate
%       the next mean and covariance matrix, and this cycle is repeated csteps1
%       times. For small n we consider all subsets of p+1 out of n, otherwise
%       the algorithm draws 500 random subsets by default.
%       Afterwards, the 10 best solutions (means and corresponding covariance
%       matrices) are used as starting values for the final iterations.
%       These iterations stop when two subsequent determinants become equal.
%       (At most csteps3 iteration steps are taken.) The solution with smallest
%       determinant is retained.
%
%       When the dataset contains more than 2*nmini cases, the algorithm does part
%       of the calculations on (at most) maxgroup nonoverlapping subdatasets, of
%       (roughly) maxobs cases.
%
%       Stage 1: For each trial subsample in each subdataset, csteps1 (see below) iterations are
%       carried out in that subdataset. For each subdataset, the 10 best solutions are
%       stored.
%
%       Stage 2 considers the union of the subdatasets, called the merged set.
%       (If n is large, the merged set is a proper subset of the entire dataset.)
%       In this merged set, each of the 'best solutions' of stage 1 are used as starting
%       values for csteps2 (sse below) iterations. Also here, the 10 best solutions are stored.
%
%       Stage 3 depends on n, the total number of cases in the dataset.
%       If n <= 5000, all 10 preliminary solutions are iterated.
%       If n > 5000, only the best preliminary solution is iterated.
%       The number of iterations decreases to 1 according to n*p (If n*p <= 100,000 we
%       iterate csteps3 (sse below) times, whereas for n*p > 1,000,000 we take only one iteration step).


% The maximum value for n (= number of observations) is:
%nmax=50000;
nmax=100000;

% The maximum value for p (= number of variables) is:
pmax=50;

% To change the number of subdatasets and their size, the values of
% maxgroup and nmini can be changed.
maxgroup=5;
nmini=300;

% The number of iteration steps in stages 1,2 and 3 can be changed
% by adapting the parameters csteps1, csteps2, and csteps3.
csteps1=2;
csteps2=2;
csteps3=100;

% dtrial : number of subsamples if not all (p+1)-subsets will be considered.
dtrial=500;

% The 0.975 quantile of the chi-squared distribution.
chi2q=[5.02389,7.37776,9.34840,11.1433,12.8325,...
       14.4494,16.0128,17.5346,19.0228,20.4831,21.920,23.337,...
       24.736,26.119,27.488,28.845,30.191,31.526,32.852,34.170,...
       35.479,36.781,38.076,39.364,40.646,41.923,43.194,44.461,...
       45.722,46.979,48.232,49.481,50.725,51.966,53.203,54.437,...
       55.668,56.896,58.120,59.342,60.561,61.777,62.990,64.201,...
       65.410,66.617,67.821,69.022,70.222,71.420];

```

```
% Median of the chi-squared distribution.
chimed=[0.454937,1.38629,2.36597,3.35670,4.35146,...
       5.34812,6.34581,7.34412,8.34283,9.34182,10.34,11.34,12.34,...
       13.34,14.34,15.34,16.34,17.34,18.34,19.34,20.34,21.34,22.34,...
       23.34,24.34,25.34,26.34,27.34,28.34,29.34,30.34,31.34,32.34,...
       33.34,34.34,35.34,36.34,37.34,38.34,39.34,40.34,41.34,42.34,...
       43.34,44.34,45.34,46.34,47.33,48.33,49.33];


seed=0;
quan=0;
alpha=0.75;
file=0;

% The value of the fields of the input argument OPTIONS are now determined.
% If the user hasn't given a value to one of the fields, the default value
% is assigned to it.
if nargin==1
   cor=0;
   ntrial=dtrial;
   lts=0;
elseif isstruct(options)
   names=fieldnames(options);

    if strmatch('cor',names,'exact')
      cor=options.cor;
   else
      cor=0;
    end

    if strmatch('alpha',names,'exact')
      quan=1;
      alpha=options.alpha;
    end

    if strmatch('ntrial',names,'exact')
    ntrial=options.ntrial;
    else
    ntrial=dtrial;
   end

   if strmatch('lts',names,'exact')
      lts=options.lts;
   else
      lts=0;
   end

else
   error('The second input argument is not a structure.')
end

if size(data,1)==1
   data=data';
end

% Observations with missing or infinite values are ommitted.
```

```
ok=all(isfinite(data),2);
data=data(ok,:);
n=size(data,1);
p=size(data,2);

% Some checks are now performed.
if n==0
   error('All observations have missing or infinite values.')
end

if n > nmax
   error(['The program allows for at most ' int2str(nmax) ' observations.'])
end

if p > pmax
   error(['The program allows for at most ' int2str(pmax) ' variables.'])
end

if n < 2*p
   error('Need at least 2*(number of variables) observations.')
end

% hmin is the minimum number of observations whose covariance determinant
% will be minimized.
hmin=quanf(0.5,n,p);

if ¬quan
   h=quanf(0.75,n,p);
else
   h=quanf(alpha,n,p);
   if h < hmin
      error(['The MCD must cover at least ' int2str(hmin) ' observations.'])
   elseif h > n
      error('quan is greater than the number of non-missings and non-infinites.')
   end
end

fid=NaN;

% The value of some fields of the output argument are already known.
res.n_obs=n;
res.quan=h;
res.X=data;

% Some initializations.
res.flag=repmat(NaN,1,length(ok));
raw.wt=repmat(NaN,1,length(ok));
raw.robdist=repmat(NaN,1,length(ok));
res.robdist=repmat(NaN,1,length(ok));
res.mahalanobis=repmat(NaN,1,length(ok));
if ¬lts
   res.method=sprintf('\nMinimum Covariance Determinant Estimator.');
else
   res.method=sprintf('\nThe function fastmcd.m is called to compute robust distances.');
end
correl=NaN;
```

```

%    z    : if at least h observations lie on a hyperplane, then z contains the
%           coefficients of that plane.
% weights : weights of the observations that are not excluded from the computations.
%           These are the observations that don't contain missing or infinite values.
% bestobj : best objective value found.
z(1:p)=0;
weights=zeros(1,n);
bestobj=inf;

% The breakdown point of the MCD estimator is computed.
if h==hmin
   %the breakdown point is maximal.
   breakdown=(h-p)*100/n;
else
   breakdown=(n-h+1)*100/n;
end

% The classical estimates are computed.
clasmean=mean(data);
clascov=cov(data);

if p < 5
   eps=1e-12;
elseif p <= 8
   eps=1e-14;
else
   eps=1e-16;
end

% The standardization of the data will now be performed.
med=median(data);
mad=sort(abs(data-repmat(med,n,1)));
mad=mad(h,:);
ii=min(find(mad < eps));
if length(ii)
   % The h-th order statistic is zero for the ii-th variable. The array plane contains
   % all the observations which have the same value for the ii-th variable.
   plane=find(abs(data(:,ii)-med(ii)) < eps)';
   meanplane=mean(data(plane,:));
   weights(plane)=1;
   if p==1
      res.flag=weights;
      raw.wt=weights;
      [raw.center,res.center]=deal(meanplane);
      [raw.cov,res.cov,raw.objective]=deal(0);
      if ¬lts
         res.method=sprintf('\nUnivariate location and scale estimation.');
         res.method=strvcat(res.method,sprintf('%g of the %g observations are identical.',length(plane),n));
%           disp(res.method);
      end
   else
      z(ii)=1;
      res.plane=z;
      covplane=cov(data(plane,:));
      [raw.center,raw.cov,res.center,res.cov,raw.objective,raw.wt,res.flag,...
      res.method]=displ(3,length(plane),weights,n,p,meanplane,covplane,res.method,z,ok,...
                        raw.wt,res.flag,file,fid,0,NaN,h,ii);
   end
   return
end
data=(data-repmat(med,n,1))./repmat(mad,n,1);

% The standardized classical estimates are now computed.
clmean=mean(data);
clcov=cov(data);

% The univariate non-classical case is now handled.
if p==1 & h≠n
   if ¬lts
      res.method=sprintf('\nUnivariate location and scale estimation.');
   end
   [raw.center,raw.cov]=mcduni(data,n,h,n-h+1,alpha);
   scale=raw.cov./sqrt(rawconsfactor(h,n,p)*rawcorfactor(p,n,alpha));
   sor=sort((data-raw.center).^2);
   raw.objective=1/(h-1)*sum(sor(1:h))*prod(mad)^2;
   %m=2*n/asvardiag(h,n,p);
   %quantile=qf(0.975,p,m-p+1);
   quantile=chi2q(p);
   %weights=((data-raw.center)/scale).^2*(m-p+1)/(m*p)<quantile;
   weights=((data-raw.center)/scale).^2<quantile;
   raw.wt=weights;
   [res.center,res.cov]=weightmecov(data,weights,n,p);
   factor=rewconsfactor(weights,n,p);
   factor=factor*rewcorfactor(p,n,alpha);
   res.cov=factor*res.cov;
   mah=(data-res.center).^2/res.cov;
   mah_raw=(data-raw.center).^2/raw.cov;
   res.flag= mah <= chi2q(1);
   [raw.cov,raw.center]=trafo(raw.cov,raw.center,med,mad,p);
   [res.cov,res.center]=trafo(res.cov,res.center,med,mad,p);
   res.mahalanobis=abs(data'-clmean)/sqrt(clcov);
   raw.robdist=sqrt(mah_raw');
   res.robdist=sqrt(mah');
%    if ¬lts
%       disp(res.method);
%    end

   spec.ask=1;
   if ¬lts
      plotmcd(res,spec);
   end

   return
end

if det(clascov) < exp(-50*p)
   % all observations lie on a hyperplane.
   [z, eigvl]=eigs(clcov,1,0,struct('disp',0));
   res.plane=z;
   weights(1:n)=1;
   if cor
```

```
      correl=clcov./(sqrt(diag(clcov))*sqrt(diag(clcov))');
   end
   [clcov,clmean]=trafo(clcov,clmean,med,mad,p);
   [raw.center,raw.cov,res.center,res.cov,raw.objective,raw.wt,res.flag,...
   res.method]=displ(1,n,weights,n,p,clmean,clcov,res.method,z./mad',ok,...
                  raw.wt,res.flag,file,fid,cor,correl);
   if cor
      [res.cor,raw.cor]=deal(correl);
   end
   return
end

% The classical case is now handled.
if h==n
   if ¬lts
      msg=sprintf('The MCD estimates based on %g observations are equal to the classical estimates.\n',h);
      res.method=strvcat(res.method,msg);
   end
   raw.center=clmean;
   raw.cov=clcov;
   raw.objective=det(clcov);
   mah=mahalanobis(data,clmean,clcov,n,p);
   res.mahalanobis=sqrt(mah);
   raw.robdist=res.mahalanobis;
   weights=mah <= chi2q(p);
   raw.wt=weights;
   [res.center,res.cov]=weightmecov(data,weights,n,p)
   if cor
      raw.cor=raw.cov./(sqrt(diag(raw.cov))*sqrt(diag(raw.cov))');
      res.cor=res.cov./(sqrt(diag(res.cov))*sqrt(diag(res.cov))');
   end
   if det(res.cov) < exp(−50*p)
      [center,covar,z,correl,plane,count]=fit(data,NaN,med,mad,p,z,cor,res.center,res.cov,n);
      res.plane=z;
      if cor
         correl=covar./(sqrt(diag(covar))*sqrt(diag(covar))');
      end
      res.method=displrw(count,n,p,center,covar,res.method,file,z,fid,cor,correl);
      [raw.cov,raw.center]=trafo(raw.cov,raw.center,med,mad,p);
      [res.cov,res.center]=trafo(res.cov,res.center,med,mad,p);
      res.robdist=raw.robdist;
   else
      mah=mahalanobis(data,res.center,res.cov,n,p);
      weights=mah <= chi2q(p);
      [raw.cov,raw.center]=trafo(raw.cov,raw.center,med,mad,p);
      [res.cov,res.center]=trafo(res.cov,res.center,med,mad,p);
      res.robdist=sqrt(mah);
   end
   raw.objective=raw.objective*prod(mad)^2;
   res.flag=weights;
%    if ¬lts
%       disp(res.method)
%    end

   spec.ask=1;
   if ¬lts
```

```
      plotmcd(res,spec);
   end

   return
end

percent=h/n;

%  If n >= 2*nmini the dataset will be divided into subdatasets.  For n < 2*nmini the set
%  will be treated as a whole.

if n >= 2*nmini

   maxobs=maxgroup*nmini;
   if n >= maxobs
      ngroup=maxgroup;
      group(1:maxgroup)=nmini;
   else
      ngroup=floor(n/nmini);
      minquan=floor(n/ngroup);
      group(1)=minquan;
      for s=2:ngroup
         group(s)=minquan+double(rem(n,ngroup)>=s-1);
      end
   end
   part=1;
   adjh=floor(group(1)*percent);
   nsamp=floor(ntrial/ngroup);
   minigr=sum(group);
   obsingroup=fillgroup(n,group,ngroup,seed,fid);
   % obsingroup : i-th row contains the observations of the i-th group.
   % The last row (ngroup+1-th) contains the observations for the 2nd stage
   % of the algorithm.

else

   [part,group,ngroup,adjh,minigr,obsingroup]=deal(0,n,1,h,n,n);
   replow=[50,22,17,15,14,zeros(1,45)];
   if n < replow(p)
      % All (p+1)-subsets will be considered.
      al=1;
      perm=[1:p,p];
      nsamp=nchoosek(n,p+1);
   else
      al=0;
      nsamp=ntrial;
   end

end

% some further initialisations.

csteps=csteps1;
inplane=NaN;
% tottimes : the total number of iteration steps.
% fine     : becomes 1 when the subdatasets are merged.
% final    : becomes 1 for the final stage of the algorithm.
[tottimes,fine,final,prevdet]=deal(0);

if part
   % bmean1 : contains, for the first stage of the algorithm, the means of the ngroup*10
   %          best estimates.
   % bcov1  : analogous to bmean1, but now for the covariance matrices.
   % bobj1  : analogous to bmean1, but now for the objective values.
   % coeff1 : if in the k-th subdataset there are at least adjh observations that lie on
   %          a hyperplane then the coefficients of this plane will be stored in the
   %          k-th column of coeff1.
   coeff1=repmat(NaN,p,ngroup);
   bobj1=repmat(inf,ngroup,10);
   bmean1=cell(ngroup,10);
    bcov1=cell(ngroup,10);
    [bmean1{:}]=deal(NaN);
    [bcov1{:}]=deal(NaN);
end

% bmean : contains the means of the ten best estimates obtained in the second stage of the
%         algorithm.
% bcov  : analogous to bmean, but now for the covariance matrices.
% bobj  : analogous to bmean, but now for the objective values.
% coeff : analogous to coeff1, but now for the merged subdataset.
% If the data is not split up, the 10 best estimates obtained after csteps1 iterations
% will be stored in bmean, bcov and bobj.
coeff=repmat(NaN,p,1);
bobj=repmat(inf,1,10);
bmean=cell(1,10);
bcov=cell(1,10);
[bmean{:}]=deal(NaN);
[bcov{:}]=deal(NaN);

seed=0;



while final≠2


   if fine | (¬part & final)

      nsamp=10;

        if final

         adjh=h;
         ngroup=1;

         if n*p <= 1e+5
            csteps=csteps3;
         elseif n*p <=1e+6
            csteps=10-(ceil(n*p/1e+5)-2);
        else
            csteps=1;
        end

      if n > 5000
         nsamp=1;
      end

   else

      adjh=floor(minigr*percent);
      csteps=csteps2;

   end

end

% found : becomes 1 if we have a singular intermediate MCD estimate.
found=0;

for k=1:ngroup

   if ¬fine
      found=0;
   end

   for i=1:nsamp

      tottimes=tottimes+1;

      % ns becomes 1 if we have a singular trial subsample and if there are at
      % least adjh observations in the subdataset that lie on the concerning hyperplane.
      % In that case we don't have to take C-steps. The determinant is zero which is
      % already the lowest possible value. If ns=1, no C-steps will be taken and we
      % start with the next sample. If we, for the considered subdataset, haven't
      % already found a singular MCD estimate, then the results must be first stored in
      % bmean, bcov, bobj or in bmean1, bcov1 and bobj1.  If we, however, already found
      % a singular result for that subdataset, then the results won't be stored
      % (the hyperplane we just found is probably the same as the one we found earlier.
      % We then let adj be zero. This will guarantee us that the results won't be
      % stored) and we start immediately with the next sample.
      adj=1;
      ns=0;

      % For the second and final stage of the algorithm the array sortdist(1:adjh)
      % contains the indices of the observations corresponding to the adjh observations
      % with minimal relative distances with respect to the best estimates of the
      % previous stage. An exception to this, is when the estimate of the previous
      % stage is singular.  For the second stage we then distinguish two cases :
      %
      % 1. There aren't adjh observations in the merged set that lie on the hyperplane.
      %    The observations on the hyperplane are then extended to adjh observations by
      %    adding the observations of the merged set with smallest orthogonal distances
      %    to that hyperplane.
      % 2. There are adjh or more observations in the merged set that lie on the
      %    hyperplane. We distinguish two cases. We haven't or have already found such
      %    a hyperplane. In the first case we start with a new sample.  But first, we
      %    store the results in bmean1, bcov1 and bobj1. In the second case we
      %    immediately start with a new sample.
```

```
%
% For the final stage we do the same as 1. above (if we had h or more observations
% on the hyperplane we would already have found it).

if final
   if ¬isinf(bobj(i))
       meanvct=bmean{i};
       covmat=bcov{i};
       if bobj(i)==0
         [dis,sortdist]=sort(abs(sum((data-repmat(meanvct,n,1))'.*repmat(coeff,1,n))));
      else
         sortdist=mahal(data,meanvct,covmat,part,fine,final,k,obsingroup,group,...
                        minigr,n,p);
      end
   else
      break;
   end
elseif fine
   if ¬isinf(bobjl(k,i))
      meanvct=bmeanl{k,i};
       covmat=bcovl{k,i};
       if bobjl(k,i)==0
       [dis,ind]=sort(abs(sum((data(obsingroup{end},:)-repmat(meanvct,minigr,1))'.*...
           repmat(coeffl(:,k),1,minigr))));
         sortdist=obsingroup{end}(ind);
         if dis(adjh) < 1e-8
            if found==0
               obj=0;
               coeff=coeffl(:,k);
               found=1;
           else
               adj=0;
           end
           ns=1;
       end
      else
         sortdist=mahal(data,meanvct,covmat,part,fine,final,k,obsingroup,group,...
                        minigr,n,p);
      end
   else
      break;
   end
else
   % The first stage of the algorithm.
   % index : contains trial subsample.
   if ¬part
      if al
           k=p+1;
           perm(k)=perm(k)+1;
           while ¬(k==1 |perm(k) <=(n-(p+1-k)))
               k=k-1;
               perm(k)=perm(k)+1;
                   for j=(k+1):p+1
                   perm(j)=perm(j-1)+1;
               end
           end
      index=perm;
   else
      [index,seed]=randomset(n,p+1,seed);

   end
else
   [index,seed]=randomset(group(k),p+1,seed);
   index=obsingroup{k}(index);
end

meanvct=mean(data(index,:));
covmat=cov(data(index,:));

if det(covmat) < exp(-50*p)

   % The trial subsample is singular.
   % We distinguish two cases :
   %
   % 1. There are adjh or more observations in the subdataset that lie
   %    on the hyperplane. If the data is not split up, we have adjh=h and thus
   %    an exact fit. If the data is split up we distinguish two cases.
   %    We haven't or have already found such a hyperplane.  In the first case
   %    we check if there are more than h observations in the entire set
   %    that lie on the hyperplane. If so, we have an exact fit situation.
   %    If not, we start with a new trial subsample.  But first, the
   %    results must be stored bmean1, bcov1 and bobj1.  In the second case
   %    we immediately start with a new trial subsample.
   %
   % 2. There aren't adjh observations in the subdataset that lie on the
   %    hyperplane. We then extend the trial subsample until it isn't singular
   %    anymore.


   % eigvct : contains the coefficients of the hyperplane.
   [eigvct, eigvl]=eigs(covmat,1,0,struct('disp',0));

   if ¬part
      dist=abs(sum((data-repmat(meanvct,n,1))'.*repmat(eigvct,1,n)));
   else
      dist=abs(sum((data(obsingroup{k},:)-repmat(meanvct,group(k),1))'.*repmat(eigvct,1,group(k))));
   end

   obsinplane=find(dist < 1e-8);
   % count : number of observations that lie on the hyperplane.
   count=length(obsinplane);

   if count >= adjh

      if ¬part
         [center,covar,eigvct,correl]=fit(data,obsinplane,med,mad,p,eigvct,cor);
         res.plane=eigvct;
         weights(obsinplane)=1;
         [raw.center,raw.cov,res.center,res.cov,raw.objective,...
         raw.wt,res.flag,res.method]=displ(2,count,weights,n,p,center,covar,...
         res.method,eigvct,ok,raw.wt,res.flag,file,fid,cor,correl);
         if cor
```

```
                  [res.cor,raw.cor]=deal(correl);
               end
               return
            elseif found==0
               dist=abs(sum((data-repmat(meanvct,n,1))'.*repmat(eigvct,1,n)));
               obsinplane=find(dist < 1e-8);
               count2=length(obsinplane);
               if count2>=h
                  [center,covar,eigvct,correl]=fit(data,obsinplane,med,mad,p,eigvct,cor);
                  res.plane=eigvct;
                  weights(obsinplane)=1;
                  [raw.center,raw.cov,res.center,res.cov,raw.objective,...
                  raw.wt,res.flag,res.method,varargout]=displ(2,count2,weights,n,p,center,covar,...
                  res.method,eigvct,ok,raw.wt,res.flag,file,fid,cor,correl);
                  if cor
                     [res.cor,raw.cor]=deal(correl);
                  end
                 return
               end
               obj=0;
               inplane(k)=count;
               coeff1(:,k)=eigvct;
               found=1;
               ns=1;
            else
               ns=1;
               adj=0;
            end

         else

            while det(covmat) < exp(-50*p)
               [index,seed]=addobs(index,n,seed);
               covmat=cov(data(index,:));
            end
            meanvct=mean(data(index,:));

         end
      end

      if ¬ns
         sortdist=mahal(data,meanvct,covmat,part,fine,final,k,obsingroup,group,...
                           minigr,n,p);
      end

   end

   if ¬ns

      for j=1:csteps

         tottimes=tottimes+1;

         if j > 1
            % The observations correponding to the adjh smallest mahalanobis
            % distances determine the subset for the next iteration.
   sortdist=mahal(data,meanvct,covmat,part,fine,final,k,obsingroup,group,...
                     minigr,n,p);
 end

 obs_in_set=sort(sortdist(1:adjh));
meanvct=mean(data(obs_in_set,:));
covmat=cov(data(obs_in_set,:));
obj=det(covmat);

 if obj < exp(-50*p)

   % The adjh-subset is singular. If adjh=h we have an exact fit situation.
   % If adjh < h we distinguish two cases :
   %
   % 1. We haven't found earlier a singular adjh-subset. We first check if
   %    in the entire set there are h observations that lie on the hyperplane.
   %    If so, we have an exact fit situation. If not, we stop taking C-steps
   %    (the determinant is zero which is the lowest possible value) and
   %    store the results in the appropriate arrays.  We then begin with
   %    the next trial subsample.
   %
   % 2. We have, for the concerning subdataset, already found a singular
   %    adjh-subset. We then immediately begin with the next trial subsample.

   if ¬part | final | (fine & n==minigr)
      [center,covar,z,correl,obsinplane,count]=fit(data,NaN,med,mad,p,NaN,...
                                              cor,meanvct,covmat,n);
      res.plane=z;
      weights(obsinplane)=1;
      [raw.center,raw.cov,res.center,res.cov,raw.objective,...
      raw.wt,res.flag,res.method]=displ(2,count,weights,n,p,center,covar,...
      res.method,z,ok,raw.wt,res.flag,file,fid,cor,correl);
      if cor
          [res.cor,raw.cor]=deal(correl);
      end
      return
  elseif found==0
      [eigvct, eigvl]=eigs(covmat,1,0,struct('disp',0));
      dist=abs(sum((data-repmat(meanvct,n,1))'.*repmat(eigvct,1,n)));
      obsinplane=find(dist<1e-8);
      count=length(obsinplane);
      if count >= h
          [center,covar,eigvct,correl]=fit(data,obsinplane,med,mad,p,eigvct,cor);
          res.plane=eigvct;
          weights(obsinplane)=1;
          [raw.center,raw.cov,res.center,res.cov,raw.objective,...
          raw.wt,res.flag,res.method]=displ(2,count,weights,n,p,center,covar,...
          res.method,eigvct,ok,raw.wt,res.flag,file,fid,cor,correl);
          if cor
             [res.cor,raw.cor]=deal(correl);
          end
         return
      end
      obj=0;
       found=1;
       if ¬fine
```

```
               coeff1(:,k)=eigvct;
               dist=abs(sum((data(obsingroup{k},:)-...
                   repmat(meanvct,group(k),1))'.*repmat(eigvct,1,group(k))));
               inplane(k)=length(dist(dist<1e-8));
            else
               coeff=eigvct;
               dist=abs(sum((data(obsingroup{end},:)-repmat(meanvct,minigr,1))'.*repmat(eigvct,1,minigr)));
               inplane=length(dist(dist<1e-8));
            end
            break;
       else
           adj=0;
           break;
       end

      end

      % We stop taking G-steps when two subsequent determinants become equal.
      % We have then reached convergence.
      if j >= 2 & obj == prevdet
       break;
       end
      prevdet=obj;

   end % C-steps

end


% After each iteration, it has to be checked whether the new solution
% is better than some previous one.  A distinction is made between the
% different stages of the algorithm:
%
%  - Let us first consider the first (second) stage of the algorithm.
%    We distinguish two cases if the objective value is lower than the largest
%    value in bobj1 (bobj) :
%
%      1. The new objective value did not yet occur in bobj1 (bobj).  We then store
%         this value, the corresponding mean and covariance matrix at the right
%         place in resp. bobj1 (bobj), bmean1 (bmean) and bcov1 (bcov).
%         The objective value is inserted by shifting the greater determinants
%         upwards. We perform the same shifting in bmean1 (bmean) and bcov1 (bcov).
%
%      2. The new objective value already occurs in bobj1 (bobj). A comparison is
%         made between the new mean vector and covariance matrix and those
%         estimates with the same determinant. When for an equal determinant,
%         the mean vector or covariance matrix do not correspond, the new results
%         will be stored in bobj1 (bobj), bmean1 (bmean) and bcov1 (bcov).
%
%    If the objective value is not lower than the largest value in bobj1 (bobj),
%    nothing happens.
%
%  - For the final stage of the algorithm, only the best solution has to be kept.
%    We then check if the objective value is lower than the till then lowest value.
%    If so, we have a new best solution. If not, nothing happens.

```

```

         if ¬final & adj
            if fine | ¬part
               if obj < max(bobj)
                  [bmean,bcov,bobj]=insertion(bmean,bcov,bobj,meanvct,covmat,obj,1,eps);
               end
            else
               if obj < max(bobj1(k,:))
                  [bmean1,bcov1,bobj1]=insertion(bmean1,bcov1,bobj1,meanvct,covmat,obj,k,eps);
               end
            end
         end

          if final & obj< bestobj
             % bestset           : the best subset for the whole data.
             % bestobj           : objective value for this set.
             % initmean, initcov : resp. the mean and covariance matrix of this set.
             bestset=obs_in_set;
             bestobj=obj;
             initmean=meanvct;
             initcov=covmat;
         end

      end % nsamp
   end % ngroup


   if part & ¬fine
      fine=1;
   elseif (part & fine & ¬final) | (¬part & ¬final)
      final=1;
   else
      final=2;
   end

end % while loop

% factor : if we multiply the raw MCD covariance matrix with factor, we obtain consistency
%          when the data come from a multivariate normal distribution.
factor=rawconsfactor(h,n,p);
factor=factor*rawcorfactor(p,n,alpha);
% initcov=factor*initcov;
%%NIEUW
raw.cov=factor*initcov;
raw.objective=bestobj*prod(mad)^2;
[raw.cov,raw.center]=trafo(raw.cov,initmean,med,mad,p);

if cor
   raw.cor=initcov./(sqrt(diag(initcov))*sqrt(diag(initcov))');
end

% We express the results in the original units.
%[raw.cov,raw.center]=trafo(initcov,initmean,med,mad,p);
%raw.cov=factor*raw.cov;
%raw.objective=bestobj*prod(mad)^2;

```

```
%The reweighted robust estimates are now computed.
%mah=mahalanobis(data,initmean,initcov,n,p);
%%%NIEUW
mah=mahalanobis(data,initmean,initcov*factor,n,p);
raw.robdist=sqrt(mah);
%m=2*n/asvardiag(h,n,p);
%quantile=qf(0.975,p,m-p+1);
quantile=chi2q(p);
%weights=mah*(m-p+1)/(m*p)<quantile;
weights=mah<quantile;
raw.wt=weights;
[res.center,res.cov]=weightmecov(data,weights,n,p);
factor=rewconsfactor(weights,n,p);
factor=factor*rewcorfactor(p,n,alpha);
res.cov=factor*res.cov;

[trcov,trcenter]=trafo(res.cov,res.center,med,mad,p);

if cor
   res.cor=res.cov./(sqrt(diag(res.cov))*sqrt(diag(res.cov))');
end

if det(trcov) < exp(-50*p)
   [center,covar,z,correl,plane,count]=fit(data,NaN,med,mad,p,z,cor,res.center,res.cov,n);
   res.plane=z;
   if cor
      correl=covar./(sqrt(diag(covar))*sqrt(diag(covar))');
   end
   res.method=displrw(count,n,p,center,covar,res.method,file,z,fid,cor,correl);
   res.flag=weights;
   res.robdist=raw.robdist;
else
   mah=mahalanobis(data,res.center,res.cov,n,p);
   res.flag=(mah <= chi2q(p));
   res.robdist=sqrt(mah);
end

res.mahalanobis=sqrt(mahalanobis(data,clmean,clcov,n,p));
res.cov=trcov;
res.center=trcenter;

% if ¬lts
%    disp(res.method)
% end
spec.ask=1;
if ¬lts
   plotmcd(res,spec);
end

%--------------------------------------------------------------------------------------------
function [raw_center,raw_cov,center,covar,raw_objective,raw_wt,mcd_wt,method]=displ(exactfit,...
          count,weights,n,p,center,covar,method,z,ok,raw_wt,mcd_wt,file,fid,cor,correl,varargin)

% Determines some fields of the output argument RES for the exact fit situation.  It also
% displays and writes the messages concerning the exact fit situation.  If the raw MCD
% covariance matrix is not singular but the reweighted is, then the function displrw is
% called instead of this function.

[raw_center,center]=deal(center);
[raw_cov,cov]=deal(covar);
raw_objective=0;
mcd_wt=weights;
raw_wt=weights;

switch exactfit
case 1
   msg='The covariance matrix of the data is singular.';
case 2
   msg='The covariance matrix has become singular during the iterations of the MCD algorithm.';
case 3
   msg=sprintf('The %g-th order statistic of the absolute deviation of variable %g is zero. ',...
       varargin{1},varargin{2});
end

msg=sprintf([msg '\nThere are %g observations in the entire dataset of %g observations that lie on the \n'],count,n);
switch p
case 2
   msg=sprintf([msg 'line with equation %g(x_i1-m_1)%+g(x_i2-m_2)=0 \n'],z);
   msg=sprintf([msg 'where the mean (m_1,m_2) of these observations is the MCD location']);
case 3
   msg=sprintf([msg 'plane with equation %g(x_i1-m_1)%+g(x_i2-m_2)%+g(x_i3-m_3)=0 \n'],z);
   msg=sprintf([msg 'where the mean (m_1,m_2,m_3) of these observations is the MCD location']);
otherwise
   msg=sprintf([msg 'hyperplane with equation a_1 (x_i1-m_1) + ... + a_p (x_ip-m_p) = 0 \n']);
   msg=sprintf([msg 'with coefficients a_i equal to : \n\n']);
   msg=sprintf([msg sprintf('%g  ',z)]);
   msg=sprintf([msg '\n\nand where the mean (m_1,...,m_p) of these observations is the MCD location']);
end

method=strvcat(method,[msg '.']);
% disp(method)


%------------------------------------------------------------------------------------------
function method=displrw(count,n,p,center,covar,method,file,z,fid,cor,correl)

% Displays and writes messages in the case the reweighted robust covariance matrix
% is singular.

msg=sprintf('The reweighted MCD scatter matrix is singular. \n');
msg=sprintf([msg 'There are %g observations in the entire dataset of %g observations that lie on the\n'],count,n);

switch p
case 2
   msg=sprintf([msg 'line with equation %g(x_i1-m_1)%+g(x_i2-m_2)=0 \n\n'],z);
   msg=sprintf([msg 'where the mean (m_1,m_2) of these observations is : \n\n']);
case 3
   msg=sprintf([msg 'plane with equation %g(x_i1-m_1)%+g(x_i2-m_2)%+g(x_i3-m_3)=0 \n\n'],z);
   msg=sprintf([msg 'where the mean (m_1,m_2,m_3) of these observations is : \n\n']);
otherwise
   msg=sprintf([msg 'hyperplane with equation a_1 (x_i1-m_1) + ... + a_p (x_ip-m_p) = 0 \n']);
   msg=sprintf([msg 'with coefficients a_i equal to : \n\n']);
   msg=sprintf([msg sprintf('%g  ',z)]);
   msg=sprintf([msg '\n\nand where the mean (m_1,...,m_p) of these observations is : \n\n']);
end

msg=sprintf([msg sprintf('%g  ',center)]);
msg=sprintf([msg '\n\nTheir covariance matrix equals : \n\n']);
msg=sprintf([msg sprintf([repmat('% 13.5g ',1,p) '\n'],covar)]);
if cor
   msg=sprintf([msg '\n\nand their correlation matrix equals : \n\n']);
   msg=sprintf([msg sprintf([repmat('% 13.5g ',1,p) '\n'],correl)]);
end

method=strvcat(method,msg);

%------------------------------------------------------------------------------

function [wmean,wcov]=weightmecov(dat,weights,n,nvar)

% Computes the reweighted estimates.

if size(weights,1)==1
   weights=weights';
end
wmean=sum(dat.*repmat(weights,1,nvar))/sum(weights);
wcov=zeros(nvar,nvar);
for obs=1:n
   hlp=dat(obs,:)-wmean;
   wcov=wcov+weights(obs)*hlp'*hlp;
end
wcov=wcov/(sum(weights)-1);

%------------------------------------------------------------------------------
function [initmean,initcov]=mcduni(y,ncas,h,len,alpha)

% The exact MCD algorithm for the univariate case.

y=sort(y);

ay(1)=sum(y(1:h));

for samp=2:len
   ay(samp)=ay(samp-1)-y(samp-1)+y(samp+h-1);
end

ay2=ay.^2/h;

sq(1)=sum(y(1:h).^2)-ay2(1);

for samp=2:len
   sq(samp)=sq(samp-1)-y(samp-1)^2+y(samp+h-1)^2-ay2(samp)+ay2(samp-1);
end

sqmin=min(sq);
ii=find(sq==sqmin);
ndup=length(ii);
slutn(1:ndup)=ay(ii);
```

```
initmean=slutn(floor((ndup+1)/2))/h;
factor=rawcorfactor(1,ncas,alpha);
factor=factor*rawconsfactor(h,ncas,1);
initcov=factor*sqmin/h;

%-----------------------------------------------------------------------------
function [initmean,initcov,z,correl,varargout]=fit(dat,plane,med,mad,p,z,cor,varargin)

% This function is called in the case of an exact fit. It computes the correlation
% matrix and transforms the coefficients of the hyperplane, the mean, the covariance
% and the correlation matrix to the original units.

if isnan(plane)
   [meanvct,covmat,n]=deal(varargin{:});
   [z, eigvl]=eigs(covmat,1,0,struct('disp',0));
   dist=abs(sum((dat-repmat(meanvct,n,1))'.*repmat(z,1,n)));
   plane=find(dist < 1e-8);
   varargout{1}=plane;
   varargout{2}=length(plane);
end

z=z./mad';
[initcov,initmean]=trafo(cov(dat(plane,:)),mean(dat(plane,:)),med,mad,p);
if cor
   correl=initcov./(sqrt(diag(initcov))*sqrt(diag(initcov))');
else
   correl=NaN;
end

%-----------------------------------------------------------------------------
function obsingroup=fillgroup(n,group,ngroup,seed,fid)

% Creates the subdatasets.

obsingroup=cell(1,ngroup+1);

jndex=0;
for k=1:ngroup
   for m=1:group(k)
      [random,seed]=uniran(seed);
      ran=floor(random*(n-jndex)+1);
       jndex=jndex+1;
       if jndex==1
         index(1,jndex)=ran;
         index(2,jndex)=k;
      else
         index(1,jndex)=ran+jndex-1;
         index(2,jndex)=k;
         ii=min(find(index(1,1:jndex-1) > ran-1+[1:jndex-1]));
         if length(ii)
            index(1,jndex:-1:ii+1)=index(1,jndex-1:-1:ii);
            index(2,jndex:-1:ii+1)=index(2,jndex-1:-1:ii);
            index(1,ii)=ran+ii-1;
            index(2,ii)=k;
         end
      end
   end
   obsingroup{k}=index(1,index(2,:)==k);
   obsingroup{ngroup+1}=[obsingroup{ngroup+1},obsingroup{k}];
end

%--------------------------------------------------------------------------------
function [ranset,seed]=randomset(tot,nel,seed)

% This function is called if not all (p+1)−subsets out of n will be considered.
% It randomly draws a subsample of nel cases out of tot.

for j=1:nel
   [random,seed]=uniran(seed);
   num=floor(random*tot)+1;
   if j > 1
      while any(ranset==num)
         [random,seed]=uniran(seed);
         num=floor(random*tot)+1;
      end
   end
   ranset(j)=num;
end

%--------------------------------------------------------------------------------
function [index,seed]=addobs(index,n,seed)

% Extends a trial subsample with one observation.

jndex=length(index);
[random,seed]=uniran(seed);
ran=floor(random*(n−jndex)+1);
jndex=jndex+1;
index(jndex)=ran+jndex−1;
ii=min(find(index(1:jndex−1) > ran−1+[1:jndex−1]));
if length(ii)≠0
   index(jndex:−1:ii+1)=index(jndex−1:−1:ii);
   index(ii)=ran+ii−1;
end

%--------------------------------------------------------------------------------

function mahsort=mahal(dat,meanvct,covmat,part,fine,final,k,obsingroup,group,minigr,n,nvar)

% Orders the observations according to the mahalanobis distances.

if ¬part | final
   [dis,ind]=sort(mahalanobis(dat,meanvct,covmat,n,nvar));
   mahsort=ind;
elseif fine
   [dis,ind]=sort(mahalanobis(dat(obsingroup{end},:),meanvct,covmat,minigr,nvar));
   mahsort=obsingroup{end}(ind);
else
   [dis,ind]=sort(mahalanobis(dat(obsingroup{k},:),meanvct,covmat,group(k),nvar));
   mahsort=obsingroup{k}(ind);
end

```

```
%--------------------------------------------------------------------------------------

function [covmat,meanvct]=trafo(covmat,meanvct,med,mad,nvar)

% Transforms a mean vector and a covariance matrix to the original units.

covmat=covmat.*repmat(mad,nvar,1).*repmat(mad',1,nvar);
meanvct=meanvct.*mad+med;

%--------------------------------------------------------------------------------------
function [bestmean,bestcov,bobj]=insertion(bestmean,bestcov,bobj,meanvct,covmat,obj,row,eps)

% Stores, for the first and second stage of the algorithm, the results in the appropriate
% arrays if it belongs to the 10 best results.

insert=1;

equ=find(obj==bobj(row,:));

for j=equ
   if (meanvct==bestmean{row,j}) & all(covmat==bestcov{row,j})
      insert=0;
   end
end

if insert
   ins=min(find(obj < bobj(row,:)));

   if ins==10
      bestmean{row,ins}=meanvct;
        bestcov{row,ins}=covmat;
      bobj(row,ins)=obj;
   else
      [bestmean{row,ins+1:10}]=deal(bestmean{row,ins:9});
    bestmean{row,ins}=meanvct;
    [bestcov{row,ins+1:10}]=deal(bestcov{row,ins:9});
    bestcov{row,ins}=covmat;
    bobj(row,ins+1:10)=bobj(row,ins:9);
    bobj(row,ins)=obj;
   end

end
%--------------------------------------------------------------------------------------

function mah=mahalanobis(dat,meanvct,covmat,n,p)

% Computes the mahalanobis distances.

for k=1:p
   d=covmat(k,k);
   covmat(k,:)=covmat(k,:)/d;
   rows=setdiff(1:p,k);
   b=covmat(rows,k);
   covmat(rows,:)=covmat(rows,:)-b*covmat(k,:);
   covmat(rows,k)=-b/d;
   covmat(k,k)=1/d;
```

```
end

hlp=dat-repmat(meanvct,n,1);
mah=sum(hlp*covmat.*hlp,2)';

%------------------------------------------------------------------------------

function [random,seed]=uniran(seed)

% The random generator.

seed=floor(seed*5761)+999;
quot=floor(seed/65536);
seed=floor(seed)-floor(quot*65536);
random=seed/65536.D0;

%------------------------------------------------------------------------------

function plotmcd(mcdres,options)
return; % DEBUG: This renders this function inert while maintaining code structure.

% The 0.975 quantile of the chi-squared distribution:
chi2q=[5.02389,7.37776,9.34840,11.1433,12.8325,...
       14.4494,16.0128,17.5346,19.0228,20.4831,21.920,23.337,...
       24.736,26.119,27.488,28.845,30.191,31.526,32.852,34.170,...
       35.479,36.781,38.076,39.364,40.646,41.923,43.194,44.461,...
       45.722,46.979,48.232,49.481,50.725,51.966,53.203,54.437,...
       55.668,56.896,58.120,59.342,60.561,61.777,62.990,64.201,...
       65.410,66.617,67.821,69.022,70.222,71.420];


p=size(mcdres.X,2);

if det(mcdres.cov) < exp(-50*p)
   error('The MCD covariance matrix is singular ')
end

% The value of the fields of the input argument OPTIONS are now determined.
% If the user hasn't given a value to one of the fields, the default value
% is assigned to it.
if nargin==1
   ask=0;
   nid=3;
   xlab='X1';
   ylab='X2';
elseif isstruct(options)
   names=fieldnames(options);

   if strmatch('ask',names,'exact')
      ask=options.ask;
   else
      ask=0;
    end

   if strmatch('nid',names,'exact')
      nid=options.nid;
   else
      nid=3;
   end

   if strmatch('xlab',names,'exact')
      xlab=options.xlab;
   else
      xlab='X1';
   end

   if strmatch('ylab',names,'exact')
      ylab=options.ylab;
   else
      ylab='X2';
   end

else
   error('The second input argument is not a structure')
end

data=mcdres.X;
choice=1;
n=size(data,1);
ellip=[];

if ask
   al=0;
else
   al=1;
end

closeplot=0;

% md and rd contain resp. the classical and robust distances.
md=sqrt(mahalanobis(data,mean(data),cov(data),n,p));
%rd=sqrt(mahalanobis(data,mcdres.center,mcdres.cov,n,p));
rd=sqrt(mahalanobis(data,mcdres.center,mcdres.cov,n,p));
%
% while choice ≠7
%    if ask
%
%       choice=menu('Make a plot selection :','All','Robust Distances',...
%          'Mahalanobis Distances','QQ plot of Robust Distances',...
%          'Robust versus Mahalanobis Distances',...
%          'MCD Tolerance Ellipse','Exit');
%
%       if closeplot==1 & choice≠7 & ¬(choice==6 & p≠2)
%          % Close previous plots.
%          for i=1:5
%             close
%          end
%          closeplot=0;
%       end
%
%       if choice==1
%          al=2;
%      end
%
%   end
%
%   if choice==1
%      choice=2;
%   end
%
%   if al & ¬(choice==6 & p≠2 ¦ choice==2)
%      % Create a new figure window.
%      figure
%   end
%
%   switch choice
%
%   case 2
%      x=1:n;
%      y=rd;
%      ymax=max([max(y),sqrt(chi2q(p)),2.5])*1.05;
% %     beg('Index','Robust Distance',rd,x,y,nid,n,-0.025*n,n*1.05,-0.025*ymax,ymax);
%      beg('Production Sequence','Robust Distance',rd,x,y,nid,n,-0.025*n,n*1.05,-0.025*ymax,ymax);
%      line([-0.025*n,n*1.05],repmat(max([sqrt(chi2q(p)),2.5]),1,2),'Color','r');
%
%   case 3
%      x=1:n;
%      y=md;
%      ymax=max([max(y),sqrt(chi2q(p)),2.5])*1.05;
% %     beg('Index','Mahalanobis Distance',md,x,y,nid,n,-0.025*n,n*1.05,-0.025*ymax,ymax);
%      beg('Production Sequence','Mahalanobis Distance',md,x,y,nid,n,-0.025*n,n*1.05,-0.025*ymax,ymax);
%      line([-0.025*n,n*1.05],repmat(max([sqrt(chi2q(p)),2.5]),1,2),'Color','r');
%
%   case 4
%      chisqquantile=repmat(0,1,n);
%      for i=1:n
%         chisqquantile(i)=qchisq((i-1/3)/(n+1/3),p);
%      end
%      normqqplot(sqrt(chisqquantile),rd);
%      box;
%      xlabel('Square root of the quantiles of the chi-squared distribution');
%      ylabel('Robust distances');
%
%   case 5
%      x=md;
%      y=rd;
%      ymax=max([max(y),sqrt(chi2q(p)),2.5])*1.05;
%      xmax=max([max(x),sqrt(chi2q(p)),2.5])*1.05;
%      beg('Mahalanobis Distance','Robust Distance',rd,x,y,nid,n,-0.01*xmax,xmax,-0.01*ymax,ymax);
%      line(repmat(max([sqrt(chi2q(p)),2.5]),1,2),[-0.01*ymax,ymax],'Color','r');
%      line([-0.01*xmax,xmax],repmat(max([sqrt(chi2q(p)),2.5]),1,2),'Color','r');
%      hold on
%      plot([-0.01*xmax,min([xmax,ymax])],[-0.01*ymax,min([xmax,ymax])],':','Color','g');
%      hold off
%
%   case 6
%      if p≠2
%         disp('MCD Tolerance Ellips is only drawn for two-dimensional datasets')
%       else
%          if isempty(ellip)
%           ellip=ellipse(mcdres.center,mcdres.cov);
%          end
%          xmin=min([data(:,1);ellip(:,1)]);
%          xmax=max([data(:,1);ellip(:,1)]);
%          ymin=min([data(:,2);ellip(:,2)]);
%          ymax=max([data(:,2);ellip(:,2)]);
%          xmarg=0.05*abs(xmax—xmin);
%          ymarg=0.05*abs(ymax—ymin);
%          xmin=xmin—xmarg;
%           xmax=xmax+xmarg;
%           ymin=ymin—ymarg;
%           ymax=ymax+ymarg;
%          beg(xlab,ylab,rd,data(:,1)',data(:,2)',nid,n,xmin,xmax,ymin,ymax);
%          title('Tolerance ellipse ( 97.5 % )');
%          line(ellip(:,1),ellip(:,2));
%       end
%
%    end
%
%    if al & choice < 6
%       ask=0;
%       choice=choice+1;
%    elseif al==1 & choice==6
%       choice=7;
%    elseif al==2 & choice==6
%       al=0;
%       ask=1;
%       closeplot=1;
%    end
%
%  end

%————————————————————————————————————————————————————————————————————————

function beg(xlab,ylab,ord,x,y,nid,n,xmin,xmax,ymin,ymax)

% Creates a scatter plot.

scatter(x,y,3,'k')

xlabel(xlab);
ylabel(ylab);

xlim([xmin,xmax]);
ylim([ymin,ymax]);
box;
if nid
   [ord,ind]=sort(ord);
   ind=ind(n—nid+1:n)';
    text(x(ind),y(ind),int2str(ind));
end

%————————————————————————————————————————————————————————————————————————
```

```
function coord=ellipse(mean,covar)

% Determines the coordinates of some points that lie on the 97.5 % tolerance ellipse.

deter=covar(1,1)*covar(2,2)-covar(1,2)^2;
ylimit=sqrt(7.37776*covar(2,2));
y=-ylimit:0.005*ylimit:ylimit;
sqtdi=sqrt(deter*(ylimit^2-y.^2))/covar(2,2);
sqtdi([1,end])=0;
b=mean(1)+covar(1,2)/covar(2,2)*y;
x1=b-sqtdi;
x2=b+sqtdi;
y=mean(2)+y;
coord=[x1,x2([end:-1:1]);y,y([end:-1:1])]';

%------------------------------------------------------------------------------

function quan=quanf(alpha,n,rk)

quan=floor(2*floor((n+rk+1)/2)-n+2*(n-floor((n+rk+1)/2))*alpha);

%------------------------------------------------------------------------------

function rawconsfac=rawconsfactor(quan,n,p)

qalpha=qchisq(quan/n,p);
calphainvers=pgamma(qalpha/2,p/2+1)/(quan/n);
calpha=1/calphainvers;
rawconsfac=calpha;

%------------------------------------------------------------------------------

function rewconsfac=rewconsfactor(weights,n,p)

if sum(weights)==n
   cΔ.rew=1;
else
   qΔ.rew=qchisq(sum(weights)/n,p);
   cΔinvers.rew=pgamma(qΔ.rew/2,p/2+1)/(sum(weights)/n);
   cΔ.rew=1/cΔinvers.rew;
end
rewconsfac=cΔ.rew;

%------------------------------------------------------------------------------

function rawcorfac=rawcorfactor(p,n,alpha)

if p > 2
   coeffqpkwad875=[-0.455179464070565,1.11192541278794,2;-0.294241208320834,1.09649329149811,3]';
   coeffqpkwad500=[-1.42764571687802,1.26263336932151,2;-1.06141115981725,1.28907991440387,3]';
   y1_500=1+(coeffqpkwad500(1,1)*1)/p^coeffqpkwad500(2,1);
   y2_500=1+(coeffqpkwad500(1,2)*1)/p^coeffqpkwad500(2,2);
   y1_875=1+(coeffqpkwad875(1,1)*1)/p^coeffqpkwad875(2,1);
   y2_875=1+(coeffqpkwad875(1,2)*1)/p^coeffqpkwad875(2,2);
   y1_500=log(1-y1_500);
    y2_500=log(1-y2_500);
    y_500=[y1_500;y2_500];
    A_500=[1,log(1/(coeffqpkwad500(3,1)*p^2));1,log(1/(coeffqpkwad500(3,2)*p^2))];
    coeffic_500=A_500\y_500;
    y1_875=log(1-y1_875);
     y2_875=log(1-y2_875);
    y_875=[y1_875;y2_875];
    A_875=[1,log(1/(coeffqpkwad875(3,1)*p^2));1,log(1/(coeffqpkwad875(3,2)*p^2))];
    coeffic_875=A_875\y_875;
    fp_500_n=1-(exp(coeffic_500(1))*1)/n^coeffic_500(2);
    fp_875_n=1-(exp(coeffic_875(1))*1)/n^coeffic_875(2);
else
    if p == 2
        fp_500_n=1-(exp(0.673292623522027)*1)/n^0.691365864961895;
        fp_875_n=1-(exp(0.446537815635445)*1)/n^1.06690782995919;
    end
    if p == 1
        fp_500_n=1-(exp(0.262024211897096)*1)/n^0.604756680630497;
        fp_875_n=1-(exp(-0.351584646688712)*1)/n^1.01646567502486;
    end
end
if 0.5 <= alpha & alpha <= 0.875
    fp_alpha_n=fp_500_n+(fp_875_n-fp_500_n)/0.375*(alpha-0.5);
end
if 0.875 < alpha & alpha < 1
    fp_alpha_n=fp_875_n+(1-fp_875_n)/0.125*(alpha-0.875);
end
rawcorfac=1/fp_alpha_n;

%------------------------------------------------------------------------------------------

function rewcorfac=rewcorfactor(p,n,alpha)

if p > 2
    coeffrewqpkwad875=[-0.544482443573914,1.25994483222292,2;-0.343791072183285,1.25159004257133,3]';
    coeffrewqpkwad500=[-1.02842572724793,1.67659883081926,2;-0.26800273450853,1.35968562893582,3]';
    y1_500=1+(coeffrewqpkwad500(1,1)*1)/p^coeffrewqpkwad500(2,1);
    y2_500=1+(coeffrewqpkwad500(1,2)*1)/p^coeffrewqpkwad500(2,2);
    y1_875=1+(coeffrewqpkwad875(1,1)*1)/p^coeffrewqpkwad875(2,1);
    y2_875=1+(coeffrewqpkwad875(1,2)*1)/p^coeffrewqpkwad875(2,2);
     y1_500=log(1-y1_500);
     y2_500=log(1-y2_500);
    y_500=[y1_500;y2_500];
    A_500=[1,log(1/(coeffrewqpkwad500(3,1)*p^2));1,log(1/(coeffrewqpkwad500(3,2)*p^2))];
    coeffic_500=A_500\y_500;
    y1_875=log(1-y1_875);
     y2_875=log(1-y2_875);
    y_875=[y1_875;y2_875];
    A_875=[1,log(1/(coeffrewqpkwad875(3,1)*p^2));1,log(1/(coeffrewqpkwad875(3,2)*p^2))];
    coeffic_875=A_875\y_875;
    fp_500_n=1-(exp(coeffic_500(1))*1)/n^coeffic_500(2);
    fp_875_n=1-(exp(coeffic_875(1))*1)/n^coeffic_875(2);
else
    if p == 2
        fp_500_n=1-(exp(3.11101712909049)*1)/n^1.91401056721863;
        fp_875_n=1-(exp(0.79473550581058)*1)/n^1.10081930350091;
    end
```

```
   if p == 1
      fp_500_n=1-(exp(1.11098143415027)*1)/n^1.5182890270453;
      fp_875_n=1-(exp(-0.66046776772861)*1)/n^0.88939595831888;
   end
end
if 0.5 <= alpha & alpha <= 0.875
   fp_alpha_n=fp_500_n+(fp_875_n-fp_500_n)/0.375*(alpha-0.5);
end
if 0.875 < alpha & alpha < 1
   fp_alpha_n=fp_875_n+(1-fp_875_n)/0.125*(alpha-0.875);
end
rewcorfac=1/fp_alpha_n;

%-----------------------------------------------------------------------------------------

function x = qchisq(p,a)
%QCHISQ   The chisquare inverse distribution function
%
%          x = qchisq(p,DegreesOfFreedom)

%         Anders Holtsberg, 18-11-93
%         Copyright (c) Anders Holtsberg

if any(any(abs(2*p-1)>1))
   error('A probability should be 0<=p<=1, please!')
end
if any(any(a<=0))
   error('DegreesOfFreedom is wrong')
end

x = qgamma(p,a*0.5)*2;

%-----------------------------------------------------------------------------------------

function x = qgamma(p,a)
%QGAMMA   The gamma inverse distribution function
%
%          x = qgamma(p,a)

%         Anders Holtsberg, 18-11-93
%         Copyright (c) Anders Holtsberg

if any(any(abs(2*p-1)>1))
   error('A probability should be 0<=p<=1, please!')
end
if any(any(a<=0))
   error('Parameter a is wrong')
end

x = max(a-1,0.1);
dx = 1;
while any(any(abs(dx)>256*eps*max(x,1)))
   dx = (pgamma(x,a) - p) ./ dgamma(x,a);
   x = x - dx;
   x = x + (dx - x) / 2 .* (x<0);
end
```

```

I0 = find(p==0);
x(I0) = zeros(size(I0));
I1 = find(p==1);
x(I1) = zeros(size(I0)) + Inf;

%------------------------------------------------------------------------

function f = dgamma(x,a)
%DGAMMA   The gamma density function
%
%          f = dgamma(x,a)

%        Anders Holtsberg, 18-11-93
%        Copyright (c) Anders Holtsberg

if any(any(a<=0))
   error('Parameter a is wrong')
end

f = x .^ (a-1) .* exp(-x) ./ gamma(a);
I0 = find(x<0);
f(I0) = zeros(size(I0));

%------------------------------------------------------------------------

function F = pgamma(x,a)
%PGAMMA   The gamma distribution function
%
%          F = pgamma(x,a)

%        Anders Holtsberg, 18-11-93
%        Copyright (c) Anders Holtsberg

if any(any(a<=0))
   error('Parameter a is wrong')
end

F = gammainc(x,a);
I0 = find(x<0);
F(I0) = zeros(size(I0));

%------------------------------------------------------------------------

function x = rchisq(n,a)
%RCHISQ   Random numbers from the chisquare distribution
%
%          x = rchisq(n,DegreesOfFreedom)

%        Anders Holtsberg, 18-11-93
%        Copyright (c) Anders Holtsberg

if any(any(a<=0))
   error('DegreesOfFreedom is wrong')
end

```

```
x = rgamma(n,a*0.5);


%--------------------------------------------------------------------------------

function x = rgamma(n,a)
%RGAMMA   Random numbers from the gamma distribution
%
%          x = rgamma(n,a)

%         Anders Holtsberg, 18-11-93
%         Copyright (c) Anders Holtsberg

if any(any(a<=0))
   error('Parameter a is wrong')
end

if size(n)==1
   n = [n 1];
end

x = qgamma(rand(n),a);

%--------------------------------------------------------------------------------

function normqqplot(x,y);

y = sort(y);

scatter(x,y,3,'k')

%--------------------------------------------------------------------------------

%function asvar=asvardiag(quan,n,p)
%
%alfa=quan/n;
%alfa=1-alfa;
%qalfa=qchisq(1-alfa,p);
%calfainvers=pgamma(qalfa/2,p/2+1);
%calfa=(1-alfa)/calfainvers;
%c2=-1/2*pgamma(qalfa/2,p/2+1);
%c3=-1/2*pgamma(qalfa/2,p/2+2);
%c4=3*c3;
%b1=(calfa*(c3-c4))/(1-alfa);
%b2=1/2+(calfa/(1-alfa))*(c3-((qalfa/p)*(c2+(1-alfa)/2)));
%asvar=(1-alfa)*b1^2*(alfa*((calfa*qalfa)/p-1)^2-1);
%asvar=asvar-2*c3*(calfa)^2*(3*(b1-p*b2)^2+(p+2)*b2*(2*b1-p*b2));
%asvar=asvar/(((1-alfa)*b1*(b1-p*b2))^2);
%
%--------------------------------------------------------------------------------

function x = qf(p,a,b)
%QF        The F inverse distribution function
%
%           x = qf(p,df1,df2)

%        Anders Holtsberg, 18-11-93
%        Copyright (c) Anders Holtsberg

x = qbeta(p,a/2,b/2);
x = x.*b./((1-x).*a);

%-----------------------------------------------------------------------

function x = qbeta(p,a,b)
%QBETA    The beta inverse distribution function
%
%         x = qbeta(p,a,b)

%       Anders Holtsberg, 27-07-95
%       Copyright (c) Anders Holtsberg

if any(any((a<=0)|(b<=0)))
   error('Parameter a or b is nonpositive')
end
if any(any(abs(2*p-1)>1))
   error('A probability should be 0<=p<=1, please!')
end
b = min(b,100000);

x = a ./ (a+b);
dx = 1;
while any(any(abs(dx)>256*eps*max(x,1)))
   dx = (betainc(x,a,b) - p) ./ dbeta(x,a,b);
   x = x - dx;
   x = x + (dx - x) / 2 .* (x<0);
   x = x + (1 + (dx - x)) / 2 .* (x>1);
end

%-----------------------------------------------------------------------

function d = dbeta(x,a,b)
%DBETA    The beta density function
%
%         f = dbeta(x,a,b)

%       Anders Holtsberg, 18-11-93
%       Copyright (c) Anders Holtsberg

if any(any((a<=0)|(b<=0)))
   error('Parameter a or b is nonpositive')
end

I = find((x<0)|(x>1));

d = x.^(a-1) .* (1-x).^(b-1) ./ beta(a,b);
d(I) = 0*I;

%-----------------------------------------------------------------------
```

Outlier_Detector.m

```
function Result = Outlier_Detect(obj)

N=size(obj.Sample_Data,2);

%% (1) Decompose the audio signal into subbands.  Convert each of bands
%% from a Laplacian Distribution to a Gaussian Distribution.  Smooth the
%% data using a low pass filter (moving average).
S=Wavelet_BPF(obj.Sample_Data,obj.Wavelet_Basis,obj.Analysis_Decomposition_Level);
N_Subbands=size(S,1);
Moving_Average_Window_Size=obj.Averaging_Window_Size;
Extraneous_Value_Index={};
T=[];
for i=1:N_Subbands
    if obj.Laplacian_Sample_Filter
        Band_Pass_Signal=Lap2Gauss(S(i,:));
    else
        Band_Pass_Signal=S(i,:);
    end;

%     % The signal is windowed in the time domain using a Hamming window to
%     % reduce windowing effects.  This must be investigated further.
%     if obj.Reduce_Edge_Effects
%         Band_Pass_Signal=Band_Pass_Signal.*hamming(size(Band_Pass_Signal,2))';
%     end;

    % The Teager energy is approximately log-normal distributed.  Construct
    % the feature vector using the log of the Teager energy to convert the
    % distribution to an approximately normal one.
    Averaged_T_Energy=filter(ones(1,Moving_Average_Window_Size)./Moving_Average_Window_Size,1,...
        Teager_Energy(Band_Pass_Signal));
    Averaged_T_Energy = Averaged_T_Energy(Moving_Average_Window_Size:end);
    %T_Energy=Averaged_T_Energy;

    % The signal is windowed in the time domain using a Hamming window to
    % reduce windowing effects on the Teager energy sample.  This must be investigated further.
    if obj.Reduce_Edge_Effects
        %Band_Pass_Signal=Band_Pass_Signal.*hamming(size(Band_Pass_Signal,2))';
        T_Energy=Averaged_T_Energy.*hamming(size(Averaged_T_Energy,2))';
    else
        T_Energy=Averaged_T_Energy;
    end;
    if obj.Laplacian_Sample_Filter
        warning off;
        lastwarn('');
        Log_T_Energy=log(T_Energy);
        warning on;
        % If there were zero energies, the logarithm will be negative
        % infinity.  This will cause problems in MCD processing, amongst
        % other things.  Remove all negative infinite energies and replace
        % them with the maximum negative value that Matlab allows.  Note
        % that a floating point minimum yields errors, while an integer
        % minimum does not (therefore it is used here).
        if any(isinf(Log_T_Energy))
            [R_Inf,C_Inf]=find(isinf(Log_T_Energy));
            Log_T_Energy(R_Inf,C_Inf)=-intmax;
            lastwarn('');
        end;
        T=vertcat(T,Log_T_Energy);
    else
        T=vertcat(T,T_Energy);
    end;
    %Log_T_Energy=log(T_Energy);
    %T=vertcat(T,Log_T_Energy);

    % Collect infinite and non-numerical (extraneous) values.
    Extraneous_Value_Index{i}=find(isnan(S(i,:)) | isinf(S(i,:)));
end;

%% Compute the MCD scatter estimator for each subband.  Note that all
%% extraneous values are automatically removed from the data during
%% analysis.

% Options for the Fast-MCD algorithm implementation.
Fast_MCD_Options.cor=1;
Fast_MCD_Options.ntrial=1000;
Fast_MCD_Options.alpha=0.75;

Sub_Band_Stats={};
for j=1:N_Subbands
    [Sub_Band_Stats{j},X]=modifiedfastmcd2(T(j,:),Fast_MCD_Options);
    %fprintf(1,'%6.6f  ------  %6.6f\n',Sub_Band_Stats{j}.center,X.center);
    %Sub_Band_Stats{j}=modifiedfastmcd2(T(j,:));
    % Now that analysis has been done, adjust the flag values so that they
    % take into account the original sub-band sample indices.
    Outlier_Index=Sub_Band_Stats{j}.flag;
    for k=1:size(Extraneous_Value_Index{j},2)
        Lower_List = Outlier_Index(find(Outlier_Index<Extraneous_Value_Index{j}(k)));
        Upper_List = Outlier_Index(find(Outlier_Index>=Extraneous_Value_Index{j}(k)));
        Outlier_Index=[Lower_List; Upper_List+1];
    end;
    Sub_Band_Stats{j}.flag = Outlier_Index;
end;

% Create a container class for each of the sub-bands.
Sub_Band_Analysis_Result={};
for m=1:N_Subbands
    Temp_Result={};
    Temp_Result.Data=S(m,2:end-1);
    Temp_Result.Outlier_Index=Remove_Discontinuous((¬Sub_Band_Stats{m}.flag)',obj.Minimum_Anomaly_Length);
    Temp_Result.Robust_Mean=Sub_Band_Stats{m}.center;
    Temp_Result.Robust_Variance=Sub_Band_Stats{m}.cov;
    Sub_Band_Analysis_Result{m}=Temp_Result;
end;
Result=Sub_Band_Analysis_Result;
obj.Sub_Band_Information = Sub_Band_Analysis_Result;

% Save the data to the class.
assignin('caller',inputname(1),obj);
```

Remove_Discontinuous.m

```
function Result = Remove_Discontinuous(x,N)
% Removes all flags (sets to 0) if they are not at least N samples long.
% This function will take one or two parameters.  If only one is specified,
% the data, then N is assumed to be 5.

%% Check input arguments.
if nargin==1
    N=5;
end;

%% Force row vectors only.
if size(x,1)>size(x,2)
    x=x';
end;
N_Vals=size(x,2);

%% Perform removal operation.
i=1;
Set_Flag_Index=[];
while i<=N_Vals
    % If a non—zero value is encountered, start counting the series waiting
    % for it to end.
    if x(i)≠0
        Set_Flag_Index=[Set_Flag_Index i];
        Set_Flag_Index_Size=size(Set_Flag_Index,2);
    else
        Set_Flag_Index_Size=size(Set_Flag_Index,2);
        if Set_Flag_Index_Size<N
            x(Set_Flag_Index)=zeros(1,Set_Flag_Index_Size);
        end;
        Set_Flag_Index=[];
        Set_Flag_Index_Size=0;
    end;
    % Increment the counter.
    i=i+1;
end;

%% If there was a count running before the analysis was complete, check to
%% make sure values are removed properly.
if Set_Flag_Index_Size<N
    x(Set_Flag_Index)=zeros(1,Set_Flag_Index_Size);
end;
Result=x;
```

Teager_Energy.m

```
function Energy = Teager_Energy(x,Averaging_Window)
% This function will compute the teager energy for the given signal vector.

[Rows,Columns] = size(x);
if Columns>Rows
    x =x;
    [Rows,Columns]=size(x);
end;

```

```
Energy = zeros(1,Columns-2);
n=2:Columns-1;
Energy(n-1)=x(n).^2-x(n-1).*x(n+1);

if nargin>1
    Moving_Average_Energy = filter(ones(1,Averaging_Window)/Averaging_Window,1,Energy);
    Energy = Moving_Average_Energy;
end;
```

Wavelet_BPF.m

```
function Subbands=Wavelet_BPF(x,Wavelet_Name,Levels)
% This function will decompose a signal into sub-bands using the specified
% wavelet.

%% Preprocessing
% Ensure the data is in a row vector.
if(size(x,2)<size(x,1))
    x=x';
end;

%  Get the number of samples and compute the maximum levels of
%  decomposition.
N=size(x,2);
Max_Levels=wmaxlev(N,Wavelet_Name);
if nargin==3
    Max_Levels=Levels;
end;

%% Perform the wavelet decomposition and reconstruction of sub-bands.
[C,L]=wavedec(x,Max_Levels,Wavelet_Name);
Subbands=[];
for i=1:Max_Levels
    Subbands(i,:) = wrcoef('d',C,L,Wavelet_Name,i);
end;

%% Produce a plot if no output is given.
if nargout==0
    Max_Levels=4;
    figure(1);
    [C,L]=wavedec(x,Max_Levels,Wavelet_Name);
    Subbands=[];
    for i=1:Max_Levels
        Subbands(i,:) = wrcoef('d',C,L,Wavelet_Name,i);
    end;
    subplot(Max_Levels+2,1,1);plot(x,'r');title('Original Signal');
    for p=2:(Max_Levels+1)
        subplot(Max_Levels+2,1,p);plot(Subbands(p-1,:));
        title(sprintf('Level %d Detail',p-1));
    end;
    subplot(Max_Levels+2,1,Max_Levels+2);plot(wrcoef('a',C,L,Wavelet_Name,Max_Levels));
    title(sprintf('Level %d Approximation',Max_Levels));

    figure(2);
    subplot(Max_Levels+2,1,1);plot(x,'r');title('Original Signal');
    for p=2:(Max_Levels+1)
        subplot(Max_Levels+2,1,p);plot(Teager_Energy(Subbands(p-1,:)));
        title(sprintf('Level %d Detail \\Psi',p-1));
    end;
    subplot(Max_Levels+2,1,Max_Levels+2);
    plot(Teager_Energy(wrcoef('a',C,L,Wavelet_Name,Max_Levels)));
    title(sprintf('Level %d Approximation',Max_Levels));
end;
```