Landmine Detection with a Standoff Acoustic/Laser Technique

by

LT (jg) John Houston Doherty US Navy

B.S., Mathematics (2006)

United States Naval Academy

Submitted to the Department of Mechanical Engineering
In Partial Fulfillment of the Requirements for the Degree of
Master of Science in Oceanographic Engineering

at the

Massachusetts Institute of Technology / Woods Hole Oceanographic Institute

August 2008

Signature of Author .....
Department of Mechanical Engineering
August 27, 2008

Certified by .....
.....
Robert Haupt
Technical Staff Member MIT Lincoln Laboratory
Thesis Supervisor

Certified by .....
Arthur Baggeroer
Professor of Mechanical Engineering
Advisor

Accepted by .....
Chairman, Department Committee on Graduate Students
LALLIT ANAND

Abstract

Landmines and mine-like traps are effective weapons that are difficult to detect and discriminate from a safe distance. The ability to detect landmines in their host environment at a distance and to discriminate them from other objects would be valuable for countering the landmine threat. This paper explores a standoff acoustic/laser technique to discriminate landmines from other forms of man-made objects (clutter) in an urban environment.

A novel approach currently under investigation by MIT Lincoln Labs, University of Mississippi, and other groups employs a non-contact acoustic/laser technique to detect landmines from a safe standoff range. This technique uses a sound source to excite vibrations in targets with an acoustic wave. These vibrations are in turn measured remotely with a Laser Doppler Vibrometer (LDV).

In this thesis, the vibration responses of landmine variants are measured, analyzed, and compared to those of common urban objects likely to be found on a landmine field or roadside. The Fourier Transform of the vibration of the target as measured by the LDV is used to generate a target vibration spectrum. Target vibration spectra in response to a sound source were experimentally measured for 59 trials, 28 of which were of simulated landmine variants and the remaining trials were of urban clutter objects.

Using an algorithm adapted from a methodology for mass spectral analysis, parameters of the target signatures are estimated; then individual target signatures are classified using a Support Vector Machine (SVM) with a training set composed of parameters from the remaining members of the total population. The best results

obtained from this methodology had a 71% probability of detection and a 3% false alarm rate corresponding to 20 of 28 of the simulated landmine variants correctly identified and a single clutter object misidentified as a landmine variant.

Table of Contents

Acknowledgments

This project would not have been possible without the invaluable contributions of a wide array of individuals and institutions. I am especially thankful to the faculty and staff of the Massachusetts Institute of Technology and Lincoln Laboratories. MIT's Admissions Department provided the series of gross clerical errors required for me to enroll in such a prestigious university, and my advisor Art Baggeroer was instrumental in ensuring I graduated. Most of all the man, the myth, the legend Rob Haupt of MIT Lincoln Labs is largely responsible for whatever success this project enjoyed.

The members of the Active Optical Systems group of MIT Lincoln Labs were especially helpful. Jerry Chen provided valuable insight into data processing techniques as well as generously provided access to his optics equipment. Tim Cazazza's expertise in Internet Protocol addresses was indispensable in troubleshooting problems with the Wavebook Data Acquisition Device. Tony Tavilla was always willing to share technical insight on electrical systems. Zahi Karam suggested the applicability of Support Vector Machines to this classification problem. Charles Doll was always willing to share his equipment and know-how. Sumanth Kaushik generously gave his time to tackle the administrative challenges required to complete this research. Dave Shue generously provided access to the laser optics range facility and equipment. Dave Ceddia provided valuable insight into the relative merits of Microsoft Windows and Apple operating systems as well as the installation of printer drivers.

Although the contributions of the above listed individuals contributed towards and were a precondition for this paper's successes, I alone am responsible for its shortcomings.

Introduction

The landmine has proven itself to be a highly effective weapon and area denial device. The primary strength of the landmine is its ability to be emplaced in a concealed location by an unseen enemy and inflict damage at a later time. In order to more effectively destroy or immobilize enemy personnel or vehicles, landmines are deliberately designed and emplaced to be as difficult to detect as possible. The ability to detect landmines would be valuable tool for denying the enemy the use of this capable weapon.

Landmines can be actuated by a timer, pressure plate, electromagnetic influence or command signal. Although the newest landmines in use by the US military include self-neutralizing features to minimize unintended casualties, self-neutralizing or self-destructing features are not typically used worldwide [26]. Furthermore due to the chaotic nature of warfare, records documenting the location of landmines are often lost, destroyed, or never existed. Even with self-neutralizing features and well documented records, landmines are inherently indiscriminate weapons.

These factors create a situation in which civilian casualties are easily inflicted during and beyond the scope of the original military conflict. According to the 2006 Landmine Monitor report, there were 1,743 fatalities and 7,328 total casualties in calendar year 2006 attributable to landmines and Explosive Remnants of War (also referred to as unexploded ordinance or UXO) [27]. Others have estimated there to be sixty to seventy million landmines worldwide that inflict 24,000 civilian casualties yearly [16]. Although casualty figures from landmines are imprecise, it is clear that significant loss of human life occurs due to land mines and unexploded ordinance.

Any method to detect landmines must have a high probability of detection, standoff capability, and a low false alarm rate in order to be useful operationally. Failure to detect a landmine places personnel and equipment in danger. The ability to detect mines minimizes the operator's risk of death or injury. False alarms diminish the utility of the system as time and resources are wasted to investigate false alarms. Urban terrain increases the level of difficulty in maintaining a low false alarm rate due to the difficulty in discriminating targets from both naturally occurring and manmade objects. In an urban environment, there is a wide range of possible disguises for the target itself and of common manmade objects or clutter that share enough characteristics with the target to possibly cause a false alarm. Existing methodologies possess some combination of the above described capabilities with varying degrees of effectiveness. A method with a high probability of detection, standoff capability, and a low false alarm rate would be an effective countermeasure for landmines.

Current Detection Methods

Existing countermeasure systems and techniques include metal detectors, ground penetrating radar, chemical detectors, and acoustic-to-seismic detectors. Electromagnetic detection methods most frequently rely on some combination of magnetometry, electromagnetic induction (EMI) detection, and ground-penetrating radar (GPR) [28]. These electromagnetic detection techniques exploit the fact that the metallic components of the landmine have higher electromagnetic conductivity than surrounding earthen material.

The metal detector uses an electrically energized transmitting coil to produce a magnetic field. This magnetic field creates eddy currents in the metal components of

landmines or other innocuous metallic objects in the vicinity. The eddy currents in turn generate a second magnetic field that is measured by the receiving coil [28 29]. Once the presence of a metallic object has been detected, the operator typically probes the ground with a bayonet or other implement in order to determine if the object is a landmine or false alarm.

The US military uses the AN/PSS-12 Mine Detector. The AN/PSS-12 set is identical to the AN-19/2 system used by other NATO countries and humanitarian organizations. Weighing in at 6 kg, the AN/PSS-12 / AN-19/2 has a detection range of a tenth of a meter for an antipersonnel landmine with 0.15 g of metal components [29].

Soldier using metal detector
www.dod.mil

A range of less than a meter places the operator of a metal detector within the blast range of the landmine exposing the operator to the risk of death or serious injury. Also, landmines are increasingly made of plastic components to counter metal detectors decreasing the probability of detection. Furthermore, metal detectors are likely to encounter metallic objects in urban areas generating a high rate of false alarms. Therefore, the metal detector lacks standoff capability and has a probability of detection and false alarm rate that can vary depending on the target and its environment.

Ground Penetrating Radar (GPR) systems have demonstrated the capability to detect buried mines and can do so at distances greater than a conventional metal detector [28 29]. GPR exploits the radar signature of the casing and internal components of landmines. Unlike metal detectors, GPR can detect plastic landmines; however, the image resolution needed to discriminate plastic mines from background clutter requires the use of short pulses that reduces range [28]. GPR operating just above the ground surface is not limited by attenuation but encounters a high false alarm rate from naturally occurring clutter such as rocks and tree roots [17]. Minefields and areas with unexploded ordinance are also likely to have shrapnel and metal debris, creating false detections from manmade clutter as well [28]. While GPR is a promising emerging technology for landmine detection, the cost, complexity, size, slowness, and limited ability to discriminate mines from clutter limit its current applications [29]. In a DARPA study of the effects of clutter on minefield clearance operations, the DARPA group conducted a site survey using infrared, EMI, and GPR systems. The DARPA group experienced a high false alarm rate and recovered only 14 inert mines out of 203 anomalies that the group detected [18]. This suggests that even when used jointly EMI and GPR are ill-suited to discriminating landmines from background.

Chemical and biological sensors have also been used for mine detection. Regardless of the material composition of the mine casing or its actuation mechanism, all landmines possess some type of explosive charge. The explosive charge is typically limited to three explosives TNT, RDX, and PETN which presents an opportunity for the chemical or biological detection of these specific compounds [20]. Oak Ridge National Research Laboratory created a "bioreporter bacteria" that was genetically modified to

experience bioluminescence when in contact with these explosive compounds [20]. While this method has proved effective in the laboratory, its unconventional nature would make it logistically challenging to employ on the battlefield.

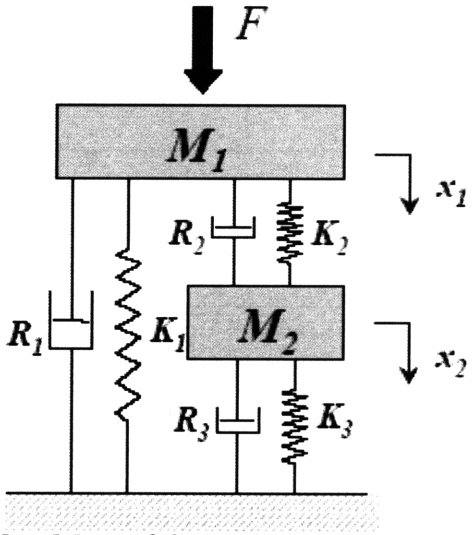
A novel detection methodology with standoff capability is the acoustic/laser technique. The acoustic/laser technique employs uses a sound source to generate an acoustic pressure wave. This acoustic pressure wave induces vibration in a target which is measured by a Laser Doppler Vibrometer (LDV). Landmines are subject to vibration due to resonant responses in their casings and internal components to these acoustic pressure waves [1 2 3 4 5 6 8 9 13 14 25]. The range of operation of the acoustic/laser technique is limited only by the power of the sound source and the sensitivity of the LDV providing standoff capability [1]. Unlike GPR systems, the false alarm rate from rocks, roots, and other soil inhomogeneities is low because solid incompressible materials such as dirt and concrete are not subject to vibration from acoustic pressure waves although they can create a radar signature that could be mistaken for a mine with a GPR system [1 2 3 4 17].

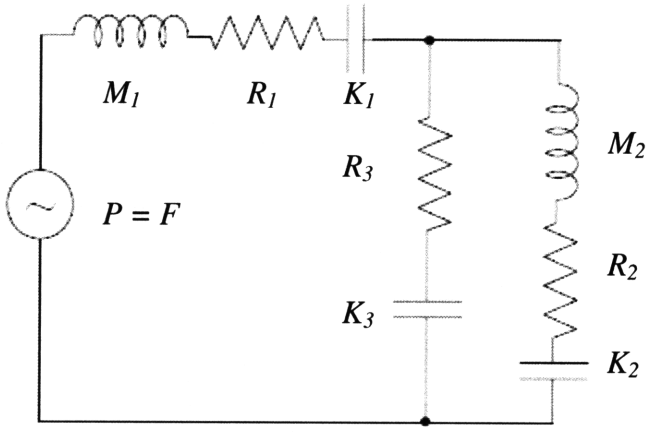
Dr. James Sabatier of the University of Mississippi and Rob Haupt of MIT Lincoln Labs among others have been developing this acoustic/laser technique by exploiting the unique characteristics of landmines in reference to acoustic-to-seismic coupling [1 2 3 4]. This acoustic-to-seismic coupling methodology was used by Dr. Sabatier and Dr. Xiang of the University of Mississippi to measure the vibration of a plastic VS 2.2 anti-tank mine with a Laser Doppler Vibrometer (LDV) [4]. The University of Mississippi team was successful in demonstrating that either metallic or

nonmetallic pressure actuated landmines would exhibit strong vibrational resonances when exposed to a sound source [3 4].

The strong vibrational resonances common to all landmines that this methodology exploits are caused by the dynamically compliant casing of the mine [33]. Because the surrounding soil is stiffer than the landmine, the area above a buried landmine experiences greater amplitude vibrations than the surrounding soil [33].

Yu and Donskoy model the buried landmine as a series of mass, springs and dampeners, or inductors, capacitors, and resistors respectively [32 33]. These models are mathematically equivalent and both have resonant frequencies for an applied force or voltage [32 33]. Furthermore, there is no such resonant response from undisturbed soil without the presence of a landmine or other highly compliant hollow object to provide an impedance contrast. Below are Yu (left) and Donskoy's (right) models of a buried mine.





M_1 – Mass of Ground
M_2 – Mass of Mine Pressure Plate
F – Acoustic Force
K_1 – Spring Constant due to Soil Sheer
K_3 – Spring constant of Mine's upper diaphragm
R_1 – Dampening Coefficient due to Soil Sheer
R_2 – Dampening Coefficient due to Soil Compression
R_3 – Mine's Dampening Coefficient
K_2 – Spring Constant due to Soil Compression

[32] [33]

$$F = -R_2\dot{x}_2 - K_2 x_2$$

$$x_2(t) = ce^{-\xi\omega t}\sin\left(\omega\sqrt{1-\xi^2}t + \theta\right)$$

$$\omega^2 = \frac{K_2 + K_3}{M_2}, \qquad \xi = \frac{R_2 + R_3}{2M_2\omega}$$

[32]

The mine vibration resonance was then successfully measured with a LDV. Dr. Sabatier's team conducted a blind test of a minefield at Fort AP Hill and attained a probability of detection of .95 and a false alarm rate or 0.03 using this methodology [4]. Rob Haupt used a similar methodology with a Parametric Acoustic Array (PAA) to excite vibration of landmines at ranges in excess of ten meters demonstrating a proof of concept of this technique for the standoff detection of landmines [1].

The acoustic/laser detection methodology used by Sabatier's team demonstrated the ability to discriminate antipersonnel mines from undisturbed soil [3] and antitank mines from gravel roads [4]. Furthermore, Haupt demonstrated that the use of a Parametric Acoustic Array as the sound source would provide this technique with standoff capability [1]. The capability to discriminate landmines from naturally occurring as well as manmade clutter would make the acoustic/laser technique an attractive and more effective landmine countermeasure than many other detection methods being researched.

Acoustic-to-Seismic Coupling

The acoustic/laser detection technique measures the vibration of landmines exposed to an acoustic pressure wave. Specifically, the sound source used for the

acoustic/laser technique generates a Rayleigh surface wave, shear wave, fast P-waves, and slow P-waves [1 2 3 4]. Both Sabatier and Haupt exploited the acoustic-to-seismic coupling of a compressional wave called the slow P-wave to excite vibration in the casing of the target [1 2 3 4].

An acoustic pressure wave incident to the ground has most of its energy reflected back into the air; however, some of its energy couples to the air/soil interface creating several seismic waves [1]. These seismic waves include the Rayleigh wave, shear wave, the fast P-wave, and slow P-wave. The Rayleigh wave, shear wave, and fast P-wave all propagate quickly through the solid granules of the soil [1]. The slow P-wave propagates through the pores of the soil which slows its speed of propagation down due to viscous drag [11]. Furthermore, the Rayleigh wave propagates along the surface inhibiting it from inducing resonances in buried objects. The equations governing the speed of the sheer wave, slow P-wave, and fast P-wave are as follows [1].

$$V_S(f) = \left(\frac{G(f)}{\rho_M + \rho_F} \right)^{1/2}$$

$$V_P(f)slow = \left(\frac{\frac{1}{C_F(f)}}{\rho_M + \rho_F} \right)^{1/2}$$

$$V_P(f)fast = \left(\frac{\frac{1}{C_F(f)} + \frac{4G(f)}{3}}{\rho_M + \rho_F} \right)^{1/2}$$

G is the soil rigidity, C_M is the soil grain matrix, C_F is the soil pore fluid compressibility, ρ_M is the soil grain matrix density, and ρ_F is the soil pore fluid density [1].

The acoustic/laser methodology relies upon acoustic-to-seismic coupling in porous material of the slow P-wave [1 2 3 4]. The Rayleigh surface wave, shear wave, and fast P-wave all propagate at higher speed than the slow P-wave [3 4]. This is significant because slower speeds are required to generate the waveforms with wavelengths comparable to the size of a landmine. Wavelengths on the order of the size of the target are necessary to induce vibration [1 2 3 4]. Furthermore, the lower sound speed of the slow P-wave in comparison to sound speed in air causes incident waves to refract downward towards the direction normal to the pressure plate of the mine which is conducive to vibration along the top surface of the mine upon which the LDV is most likely to be directed [4]. The slow P-wave is explained in greater detail by M. A. Biot [11].

Inconveniently, the slow P-wave attenuates rapidly in soil and requires high sound power. Thus the range of a standoff system is limited with commercially available sound sources due to practical size and power constraints. Haupt circumvented the limited range of a commercial speaker by using a Parametric Acoustic Array [1]. For convenience, an Eminence APT-150 commercially available sound source was used for the majority of the data collected later; however, the PAA is the lynchpin of the acoustic/laser detection technique's standoff capability.

Parametric Acoustic Array

The range of the acoustic/laser landmine detection system is a function of the amplitude of the acoustic sound source and the sensitivity of the LDV [1]. The Parametric Acoustic Array (PAA) generates a high amplitude narrow beam acoustic pressure wave that is suitable for landmine detection. P.J. Westervelt named the

Parametric Acoustic Array after the parametric amplifier due to the conceptual similarity of the two systems [10]. In P.J. Westervelt's own words:

> "It has long been known, both theoretically and experimentally, that two plane waves of differing frequencies generate, when traveling in the same direction, two new waves, one of which has a frequency equal to the sum of the original two frequencies and the other equal to the difference frequency. These 'sum' and 'difference' waves have an existence that is, in the following sense, independent of the existence of the primary generating waves: consider a semipermeable screen capable of totally absorbing the generating waves, yet freely transmitting the sum and difference waves; the latter waves will be launched into an independent existence." [10]

Haupt exploited the ability of the PAA to generate a 'difference' wave of frequency lower than the original ultrasonic acoustical signals [1]. This allows the system to be covert and protect the operator's hearing by using acoustical signals outside the audible spectrum as the carrier waves while creating a lower frequency 'difference' wave that attenuates less rapidly in the ground and is advantageous to exciting vibration in the target [1].

Acoustic/Laser Detection and Identification Methodology

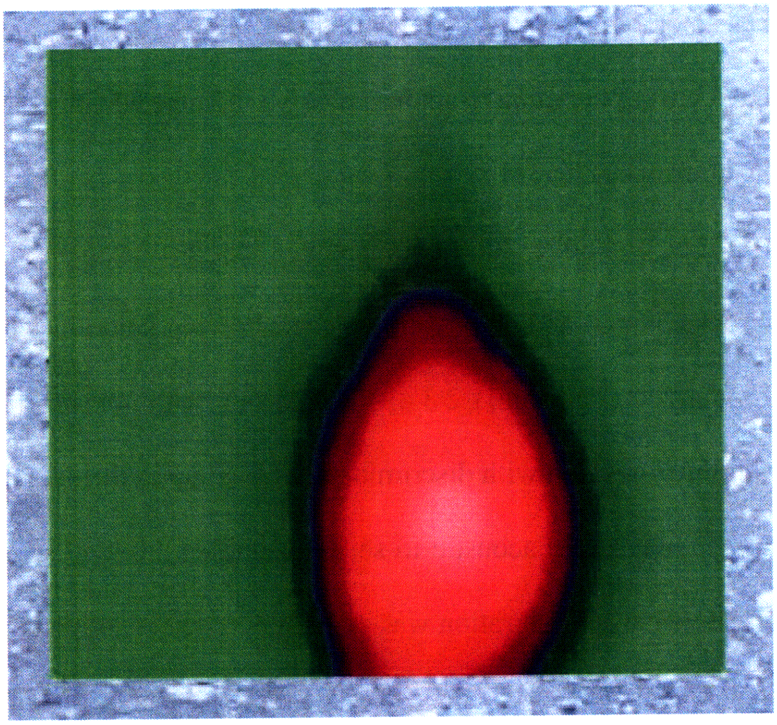
The two critical steps of a sensor based approach to mine countermeasures is detection and identification. The objective of this acoustic landmine detection methodology is to excite and measure resonant vibrations in the target in order to evaluate whether a mine is present rather than to cause its detonation. Since many landmines may not be pressure actuated, such a methodology would be insufficient to ensure for the safety of the operator and for quality assurance of the mine clearance operation. Therefore, the acoustic/laser technique relies upon measurements of the target's response to a pressure wave for both detection and identification.

Sabatier and Haupt use the large vibration signature amplitude taken from a LDV in order to detect the presence of a target. Sabatier, Haupt, Kercel, Korman, Scott, and others have demonstrated that landmines experience vibration when exposed to an incident pressure wave [1 2 3 4 5 8 25]. Mine membranes and plungers are designed to depress and detonate the mine when a person or vehicle encounters the mine [1]. As a consequence, these membranes and plungers are highly compliant and can vibrate in response to sound, causing the mine to act as a passive radiator when excited by the appropriate acoustic frequencies [1].

Sabatier, Haupt, and others have demonstrated that this resonant vibration is not characteristic of rocks, sticks, and other solid objects allowing landmines to be discriminated from undisturbed soil [1 2 3 4]. The porous nature of the ground is conducive to landmine detection with discrimination based upon the anomalous behavior of the ground surface in terms of acoustic-to-seismic coupled motion [3]. Specifically, the amplitude of the velocity of vibration was exploited to detect antipersonnel landmine [3]. A landmine was considered to be present when there was amplification of the magnitude of velocity over a relatively broad frequency band and when a circular shape in the remained intact when through this broad frequency band [3].

As mentioned earlier, Sabatier was able to yield a high probability of detection and a low false alarm rate when searching for landmines in a region of undisturbed soil that may or may not contain a landmine [3 4]. Sabatier was able to effectively use the magnitude of the velocity of vibration and the shape and size of the area with increased vibration in order to discriminate the antitank mines from areas of undisturbed soil with no mines; however, this classification technique was not tested by Sabatier with

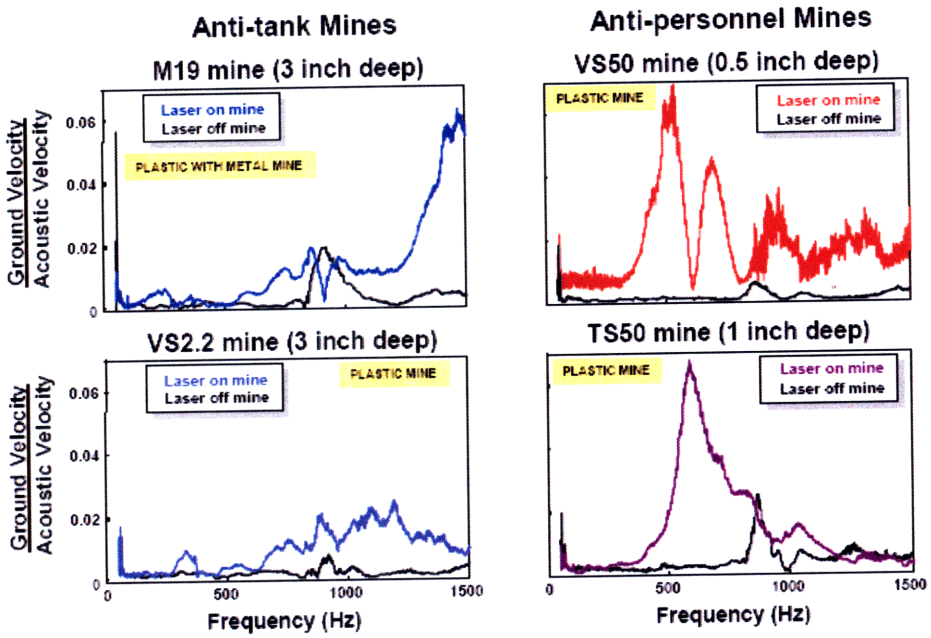
manmade urban clutter objects present [3 4]. While rocks, tree roots, soil, and other solid incompressible objects lack mechanical resonances, a variety of manmade objects are more likely to experience vibration [5]. Sabatier effectively discriminated landmines from background by spatially mapping this vibration response as seen below with the higher amplitude response of the landmine shown in red [3]:



"Scanning results in form of a color map on a PMD 6 antipersonnel mine buried 5 cm deep. Its rectangular shape has a ~top view! length of 20.5 cm and a width of 9 cm. A grid of 32 by 32 points covering an area 30 by 30 cm was defined, resulting in a spatial resolution of 1 cm. Magnitude spectra were integrated within 280–310 Hz [3]."

Haupt was also able to discriminate landmines from undisturbed soil. Haupt created a velocity vibration spectrum of the mine response to a wider band linear sound chirp function measured on a single point instead of the amplitude, size, and shape of areas of increased vibration for a series of narrow band chirps measured over an area. Haupt demonstrated that the velocity vibration spectrum of landmines is distinct from

that of the ground and could be exploited to discriminate mines from undisturbed soil as seen below [1]:



Landmine Velocity Profiles
Haupt Standoff Acoustic-to-Seismic Landmine Detection [1]

An analytical explanation for the frequency response of the landmine in comparison to soil was explored by Ssu-Hsin Yu [33]. Yu modeled the buried landmine as a series of masses, springs, and dampeners, and solved a system of linear equations for the amplitude of vibration of the soil surface in response to a force upon its surface [33]. Yu predicted that the frequency response of a landmine would have a broad peak of varying frequency and a null would exist at approximately the same frequency regardless of the relative values of the spring constants of the soil and landmine itself [33]. Using the same methodology, Yu also predicted that porous soil without the presence of a landmine would lack any such null [33]. Yu exploited the presence of the null for classification purposes with a merit function he defined as follows [33]:

$$Merit = \log(\max_{f_1 \le f \le f_2}(M(f))) - \log(\min_{f_3 \le f \le f_4}(M(f)))$$

$$f_1 \leftrightarrow f_2 - Frequency \quad Range \quad of \quad Pole$$

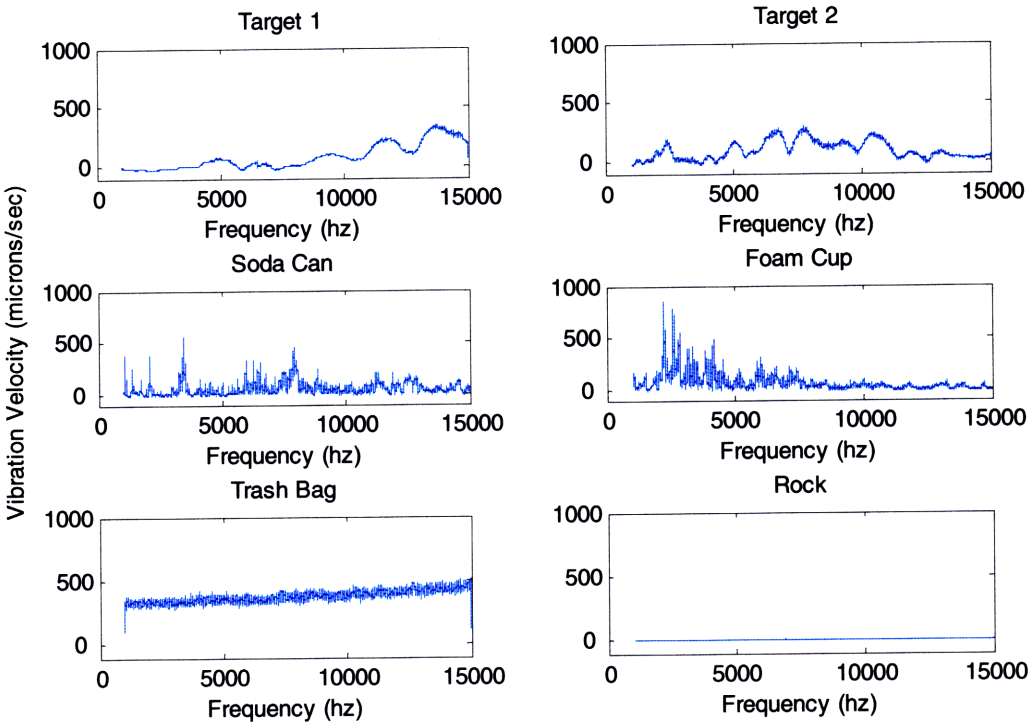
$$f_3 \leftrightarrow f_4 - Frequency \quad Range \quad of \quad Null$$

$$M(f) \quad - Magnitude \quad of \quad Vibration$$

As demonstrated by Sabatier, Haupt, and Yu, these classification approaches are potentially suitable for the detection of landmines in an environment devoid of other manmade objects [1 2 3 4 33]. Discriminating landmines from rock, soil, and other solid incompressible objects is not difficult due to the very low amplitude of vibration experienced by this type of naturally occurring clutter. However manmade objects are more likely to be manufactured from compressible materials or contain hollow spaces that can experience acoustic resonances creating false alarms for these classification techniques. These classification techniques also may experience difficulty detecting mine-like traps due to differences in their internal components and casings compared to landmines. Sabatier's discrimination method is largely dependent on the known approximate size and shape of conventional landmines buried in soil and would likely experience problems with above ground mines due to the variances in their design. Yu's merit function based on peaks and nulls would potentially encounter problems with aboveground landmines based on his own analysis [33].

Haupt's discrimination method that is based on observations of measurements of simulated landmines and clutter objects seems more promising. Haupt conducted

experimental measurements on simulated landmine variants (Target 1 and Target 2) as well as a soda can, foam cup, trash bag, and a rock, shown below:



Simulated Landmine Variants and Clutter Objects Velocity Profiles Collected by Rob Haupt

Due to the compliant nature of the simulated landmine variant casing and its larger dimensions in comparison to the clutter objects, the velocity profiles of the landmines were distinct. Conveniently the landmine velocity profiles have similar characteristics in terms of their range of amplitudes and the wide bandwidth of their features in comparison to the velocity profiles of the buried landmines shown before. Specifically, Haupt relied upon a user-controlled algorithm to select features and estimate their values for Q and modal density with Q and modal density defined as follows:

$$Q = \frac{f_{\max}}{BW}$$

$$Modal \quad Denisty = \frac{\#\, peaks}{BW}$$

$$BW - bandwidth \quad of \quad 3dB \quad 1/2 \quad width$$

$$f_{\max} - frequency \quad of \quad feature's \quad peak$$

This methodology yielded promising results; however, its reliance on user selected features presents the potential for user bias in the results. An automated methodology for the selection of peaks within the velocity profiles would negate this problem. Conveniently, William Wallace, Anthony Kearsley, and Charles Guttman of the National Institute of Standards and Technology's paper "An Operator-Independent Approach for Mass Spectral Peak Identification and Integration" addresses this same problem [19].

Haupt's classification methodology seemed to be the most promising overall in terms of landmine detection, standoff capability and clutter discrimination. Wallace's peak picking method is used to create an algorithm to select features of velocity profiles. Estimated measurements of these velocity profiles were used to classify targets on the basis of measured characteristics in order to discriminate targets from clutter without user input or potential bias.

Experimental Objective

Using Haupt's methodology, a proof of concept of the ability to discriminate landmines from clutter is the main effort of the experimental portion of this project. This was accomplished experimentally by generating target signatures of simulated landmine variants as well as various clutter objects that are likely to appear in an urban

environment. A variant of the Wallace peak picking technique was used to identify features and estimate parameters [19]. Using a Support Vector Machine (SVM) developed by Steve Gunn of the University of Southampton, a regression analysis of these estimated measurements is used to correlate specific parameters to the identity of the target signatures [15]. The SVM uses the results of this regression analysis to generalize about the characteristics of unknown objects and make predictions. A series of MATLAB algorithms were created to process experimental data and estimate the measurements needed for the SVM's regression analysis. The results of this regression analysis are used to classify objects. The results of the classification are then compared to ground truth for performance assessment.

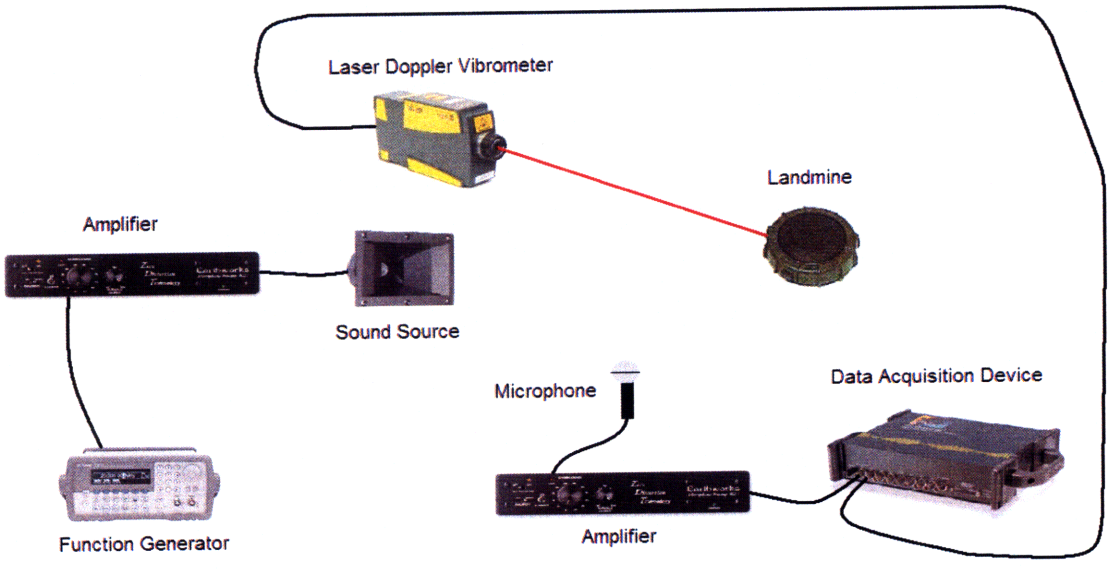
Experimental Nomenclature

- Target signature - The velocity profile of the target's response versus the frequency of the sound source provided the cornerstone of discriminating simulated landmine variant from clutter and henceforth is referred to as the target signature.
- Trial - The set of experimental measurements required to generate an individual target signature are referred to as a trial.
- Feature - The target signatures are divided into a series of individual simple shapes or features.
- Parameters - Estimated measurements of these features are parameters.
- Data set – A series of trials referred to as data sets were collected with the experimental processes and equipment described below.

Processes and Equipment

Using a standoff acoustic-laser methodology, target signatures for simulated landmines and urban clutter objects were generated experimentally. Two mockups of landmines were constructed by the technical support staff of MIT Lincoln Labs. Empty soda cans, water bottles, bags of trash, rocks, solid aluminum cylinders, and foam cups were used to simulate urban clutter. The set of data used later specifically consisted of 15 trials for the first simulated landmine designated Target 1, 13 trials for the second simulated landmine designated Target 2, 4 trials for an intact soda can, 5 trials for a crushed soda can, 5 trials for an empty water bottle, 5 trials for a foam cup, 6 trials for an bag of office trash, and 4 trials for a solid aluminum cylinder (see Appendix B).

For each trial, a sound source swept a given range of frequencies inducing vibration in the object which was measured by a LDV. Voltages from the LDV and a background microphone were measured with a Wavebook data acquisition device which allowed the raw data to be processed and analyzed. The diagram below roughly approximates the experimental configuration of the hardware used for the experiment.

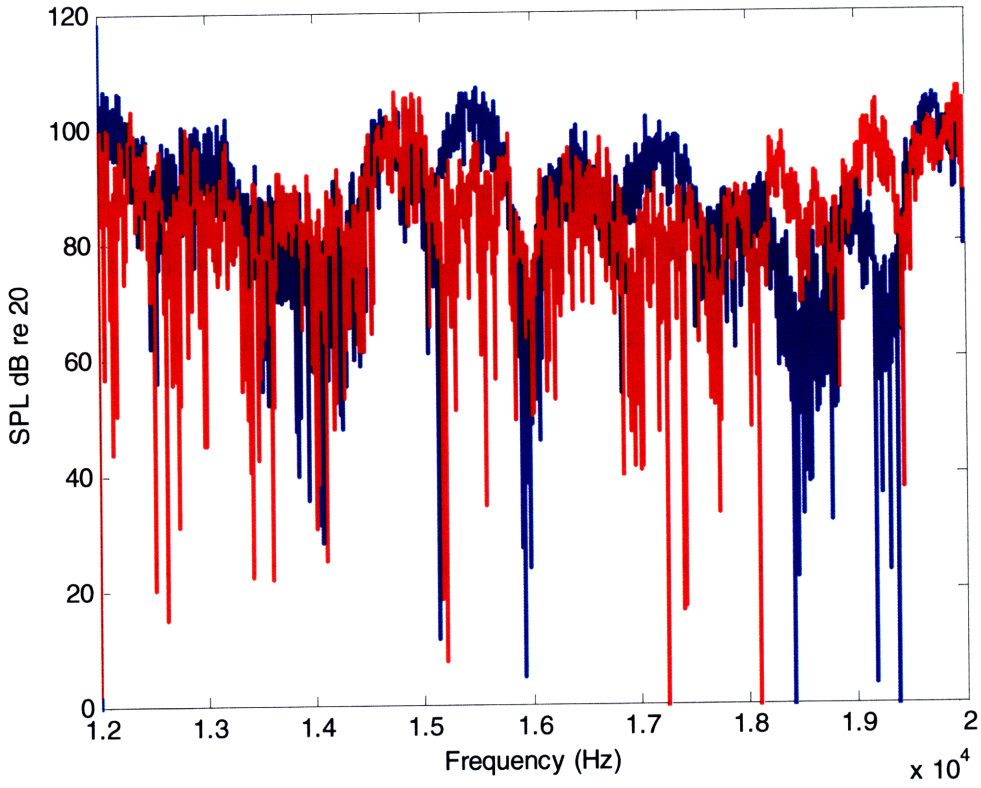


Experimental Setup

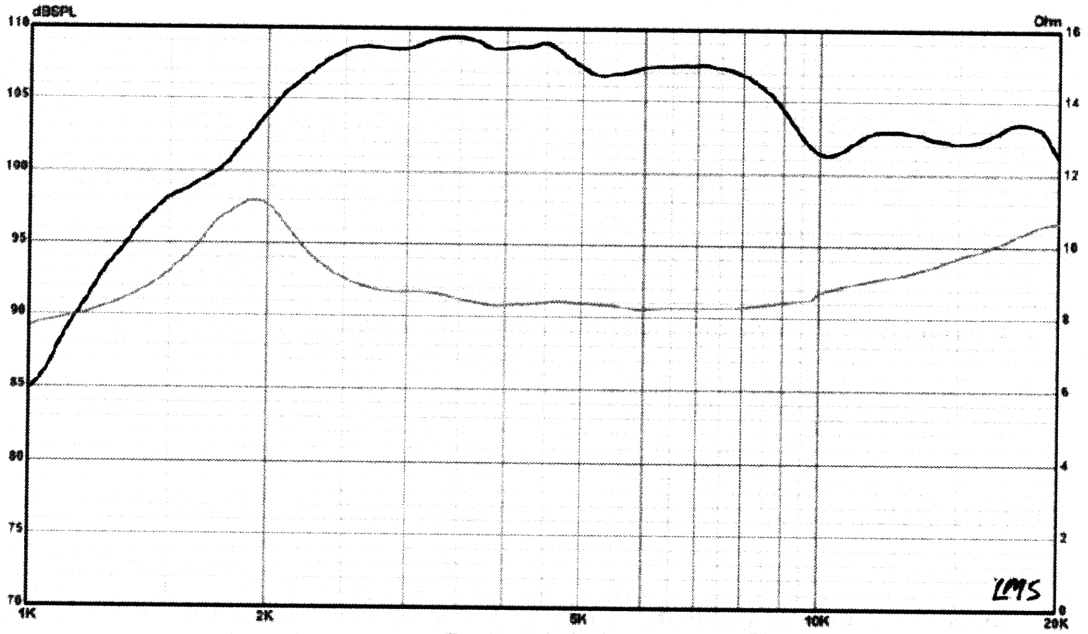
The sound source used was an Eminence APT 150, a parametric acoustic array, or a generic commercial speaker depending on the frequency band and other signal characteristics desired. An Agilent function generator through a commercial amplifier output the speaker's input signal. The speaker's input signal consisted of a sinusoid with linearly increasing frequency from a low frequency to a high frequency and constant amplitude.

Sound Source

For convenience, an Eminence APT-150 commercial speaker was used as the sound source in lieu of the Parametric Acoustic Array described earlier. The APT-150 was not able to generate as high an amplitude as the PAA but was adequate for trials conducted in indoors. In order to ensure the sound source was consistent, an Earthworks microphone located near the target measured the ground truth sound pressure level. The frequency spectrum of the microphone measurements from two nonconsecutive trials is shown below with the separate trials in red and blue respectively.



It is desirable for the sound source to generate constant amplitude levels so that the vibration of the target represents intrinsic properties of the target itself rather than external factors. From the graph above, it is clear that amplitudes of the sound source varied at different frequencies as well as between trials. The speaker's input signal was measured to be constant between trials and with changing frequency, so this variation is attributable to limitations of the speaker itself and the laboratory environment. Conducting experimental measurements in an enclosed lab space may have contributed to some of the variance of the sound pressure level due to backscattering off of walls and other hard surfaces. Additionally, the manufacturer's specifications for the APT-150 show that the sound pressure level generated is not constant with frequency under ideal conditions as seen below.



Frequency (Hz)

Manufacturer's specifications for APT-150 tweeter
www.eminence.com

There was no easy way to compensate for the inherent deficiencies of the sound source shown above. In spite of the limitations of the APT-150, it was adequate for generating target signatures and convenient for use in an indoor test facility. However, the APT-150's fluctuating and comparatively low amplitude would be make it ill suited for the detection of landmines at the long ranges desired of an operational system.

Laser Doppler Vibrometer

The standoff capability of the acoustic/laser methodology is dependent on being able to measure the vibration of a target from a distance. This is accomplished through the use of a Laser Doppler Vibrometer. The LDV emits a laser beam onto the surface of the target. Vibration in the surface of the target produces a Doppler shift in the frequency of light that is reflected back towards the LDV. The LDV uses a photoelectric cell to generate a frequency-modulated signal from the reflected light [3]. The Doppler shifted

frequency is used to determine the velocity of vibration in the direction of the beam, specifically [34]:

$$f_D = \frac{2v}{\lambda}$$

$$f_D - Doppler \quad shift$$

$$v - velocity \quad of \quad vibration \quad in \quad direction \quad of \quad beam$$

$$\lambda - wavelength$$

$$\Rightarrow v = \frac{f_D}{2\lambda}$$

Since this equation yields the velocity of vibration in the direction of the beam, the LDV and target are oriented so that the surface of the target is as close to normal to the direction of beam propagation as possible. The voltage of the demodulated signal is proportional to the velocity of vibration of the surface being measured [3].

The vibration induced in the target by the sound source was measured by a Polytec PDV-100 Laser Doppler Vibrometer. The PDV-100 has a frequency range of 0-22 kHz covering the frequency band output by the sound source with a velocity resolution of $\frac{0.02\mu m}{\sec\sqrt{frequency}}$ [30].

The PDV-100 is rated to a range of 30m but had superior signal to noise performance at the closer ranges that were used to collect data. Even at a distance of one meter, limitations on the PDV-100 were suspected to be a source of noise. The PDV-100 operates on 12 volts DC from either a battery or a converter from 120 volts AC, and had significantly reduced noise levels when using battery power due to the presence of line noise from wall power. Although PDV-100 battery is rated for four hours of operating

time, the PDV-100's sensitivity degraded after approximately an hour of continuous use requiring frequent recharging periods [30].

When in focus, the PDV-100 measures the vibration of a single point a few millimeters in diameter. In order to scan a potential minefield, it would be more useful to measure a larger area by using a wider beam. Simulating this experimentally requires leaving the beam out of focus which generates insufficient signal strength. Even with a narrowly focused beam, the PDV-100 was only able to attain a sufficient signal to noise ratio when a piece of reflective tape is adhered to the target's surface.

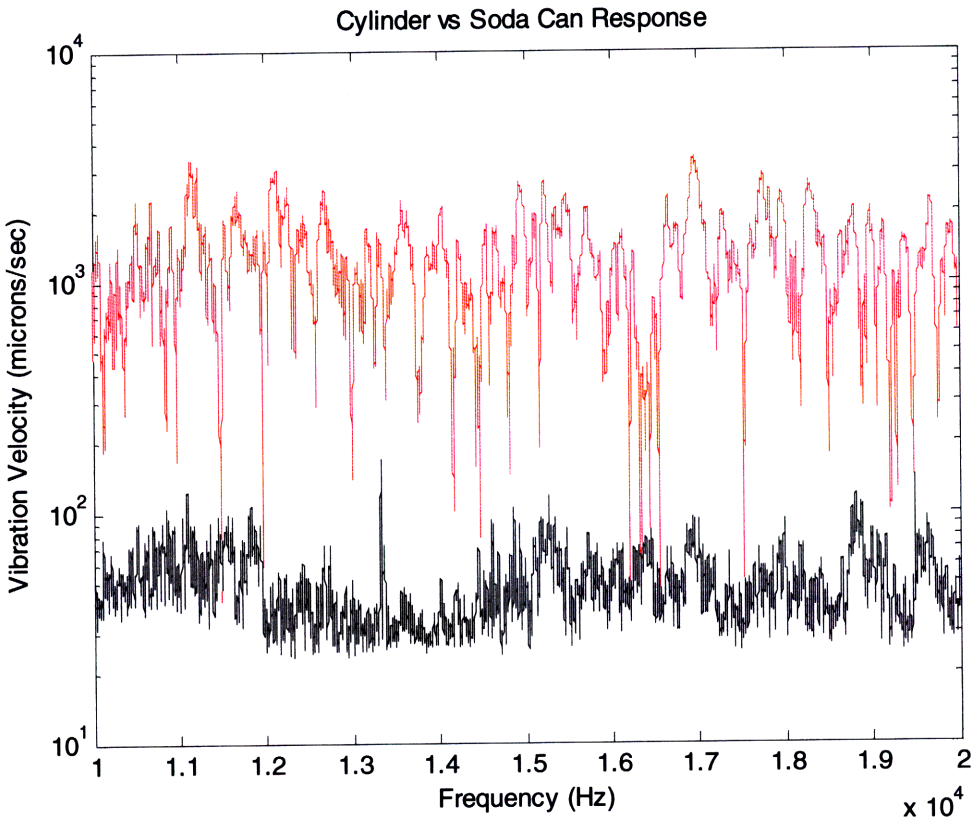
Wavebook Data Acquisitions Device

The data was collected by a Wavebook/516E digitizer. The Wavebook manufactured by IOTech converts voltages measurements into a digital signal that can be transferred to a computer via an Ethernet connection with the Waveview software package. The Wavebook can be triggered to begin collecting data with a variety of methods; in this instance, a user actuated trigger was used [35]. This creates a lag time that is eliminated with a data processing technique described later.

The Wavebook supports up to 72 channels of data which is more than sufficient for the purposes of this experiment where only 2 channels were needed for the Earthworks microphone and PDV-100 LDV. Furthermore, the Wavebook's sampling rate of 62.5 kHz/channel surpassed the Nyquist criteria of twice the maximum frequency of the signal, in this case 20 kHz. For this experiment, a sampling rate of 50 kHz was used.

Noise Floor

In order to evaluate the amount of noise present, a solid dense object such as a rock or aluminum cylinder was used as a calibration measurement. The absence of hollow spaces and very low compliance produced vibration velocity amplitudes in either the rock or the aluminum cylinder that were several orders of magnitude less than those in the landmine targets and clutter objects used in the study. An object not subject to vibration has a response that is equivalent to the noise floor of the experimental set up. In this case, the noise floor is measured with the response of a solid aluminum cylinder. In the figure below the noise floor is graphed in black in comparison to the response of an empty aluminum soda can in red.

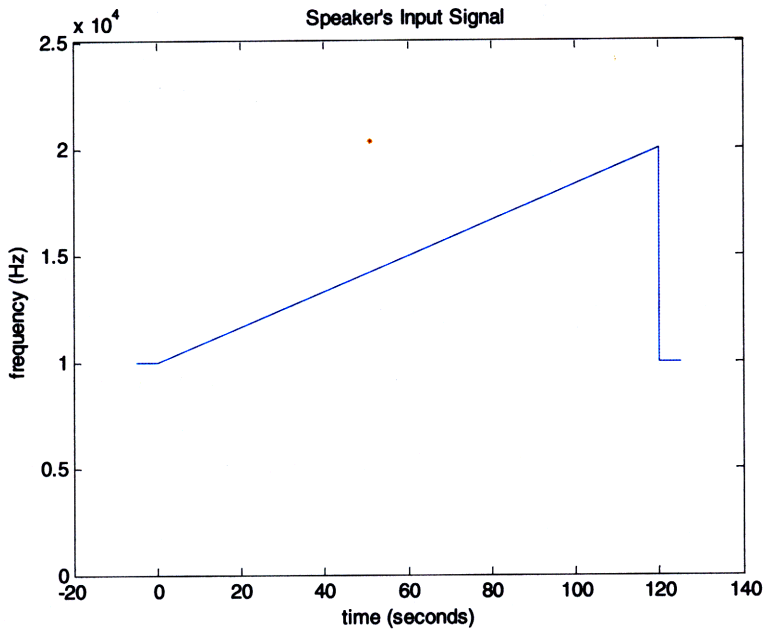


Through a trial and error process, the experimental setup was adjusted to generate the lowest noise floor possible. One such adjustment that contributed to a lower noise

floor was the elimination of the tripod used for the PDV-100. The PDV-100's tripod increased the noise floor of the target signatures due to resonances of the tripod. The tripod was replaced by bolting the PDV-100 to a block of solid aluminum. This lowered the noise floor; however, some noise was always present with the equipment and materials available.

Data Collection

Using the above described processes and equipment, data collection was conducted in a laser optics range at MIT Lincoln Labs. The PDV-100 was focused on a small piece of reflexite tape adhered to the surface of the target. Reflexite tape was used to normalize the reflectance on all the study targets. A sound source generated a 120 second long signal with a frequency that increased linearly with time across a range of frequencies. For this data set, the range of frequencies used was from 10,000 Hz to 20,000 Hz.



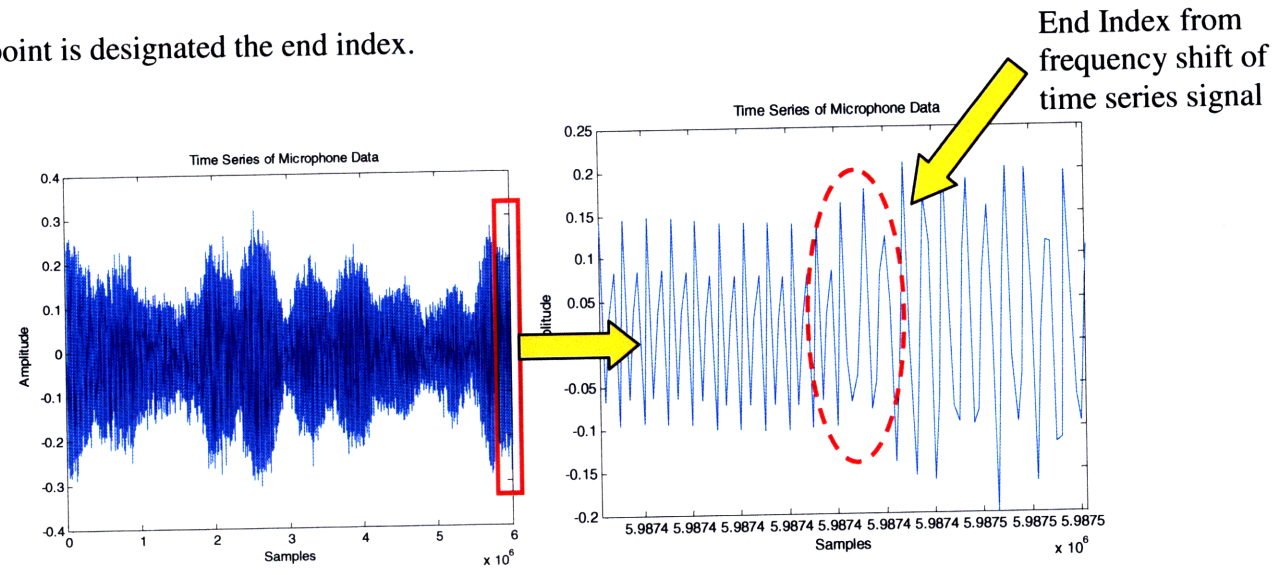
Additional recording time was added to the beginning and end of the 120 second long signal in order to compensate for a random delay when the Wavebook is triggered to begin collecting data. The random delay generated is eliminated through a post processing algorithm which is described in Appendix A.

Typically, the target was placed approximately one meter from the sound source and oriented so that the reflective tape was as close to normal to both the direction of sound propagation and the laser beam as possible. Consecutive trials of data for a single target were collected with the LDV and target reoriented between trials to conduct measurements on a different point. After several consecutive trials for an individual target, a single trial of data was collected using a solid aluminum cylinder as the target in order to assess any changes in the noise floor over the preceding trials. Collecting measurements with a ground truth of approximately zero vibration from the cylinder provided a way to assess whether the other trials were corrupted by electronic noise sources such as draining battery power supplied to the PDV-100 or other experimental error. The PDV-100 battery was fully recharged following no more than thirty cumulative trials in order to minimize such error.

Data Processing

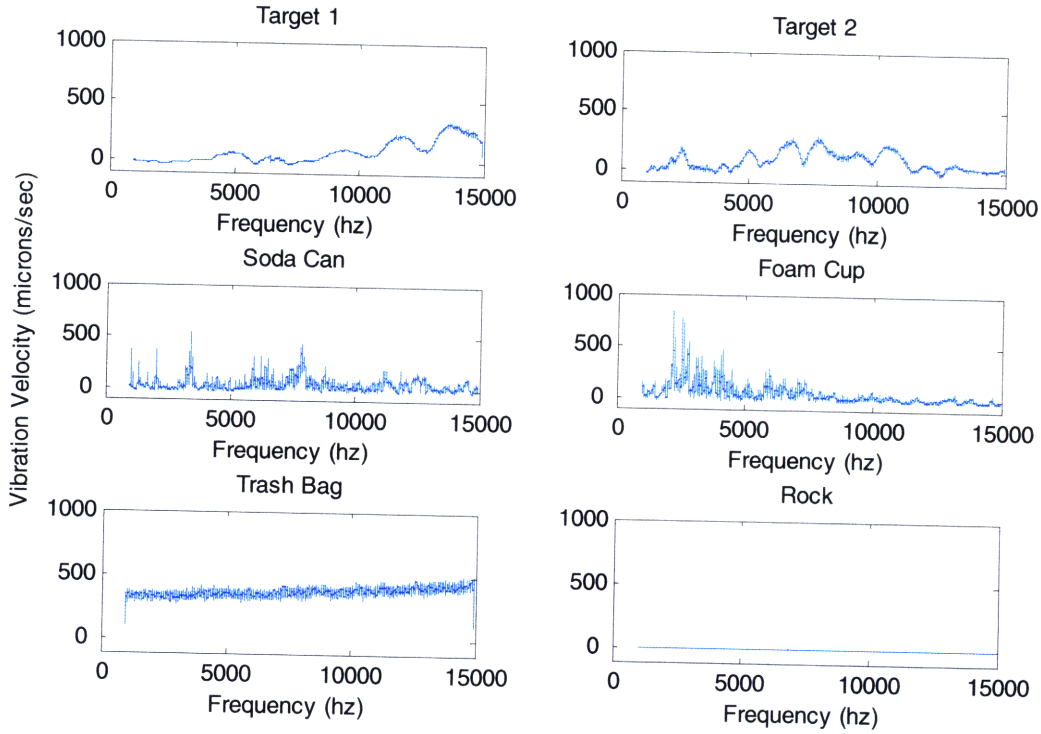
The above described methodology generates a time series of raw voltage measurements from each channel of data from the laser and the microphone to a single string of data. As previously mentioned, this time series has slack time at the beginning and end to compensate for the randomized delay in triggering the Wavebook. This slack time was cropped from the time series by exploiting the fact that the signal generated by the Agilent function generator outputs a signal with linearly increasing frequency until it

reaches the maximum frequency of the signal. Then the Agilent outputs a signal with constant frequency equal to the starting frequency of the sweep. The algorithm 'tau_calc.m' (see Appendix A) locates the point in the time series when the frequency of the microphone data rapidly drops from its high end value to its low start value. This point is designated the end index.

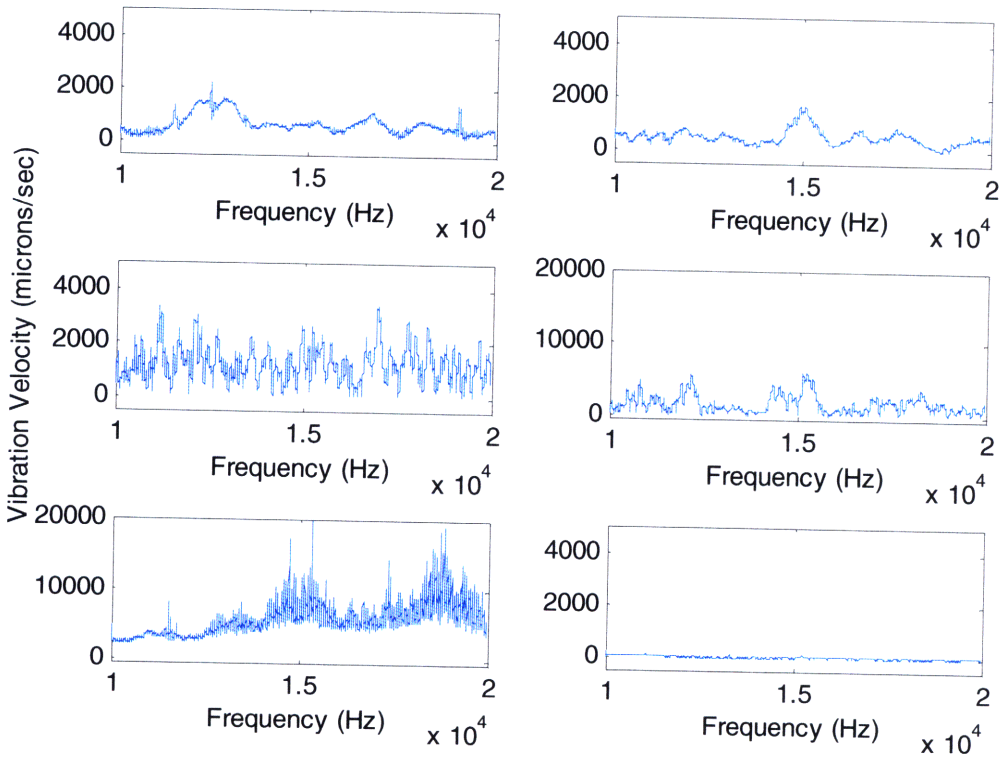


The start index is determined by counting backwards from the end index a specified number of increments in this case 5,700,000 samples. The segments of the time series before the start index and after the end index are then truncated.

The frequency spectrum is then computed from the Fast Fourier Transform (FFT) of the newly truncated time series signals from the LDV. This frequency spectrum is then downsampled by a factor of 100 to compensate for limited computer memory. The frequency spectrum is equivalent to the responsiveness of the target to a series of sound frequencies. These frequency spectrums are referred to as target signatures. A sample of the target signatures of different objects generated from the data set collected by Haupt and with the above methodology is shown below:



Haupt Simulated Landmine Variant 1&2 and Clutter Objects Velocity Profiles



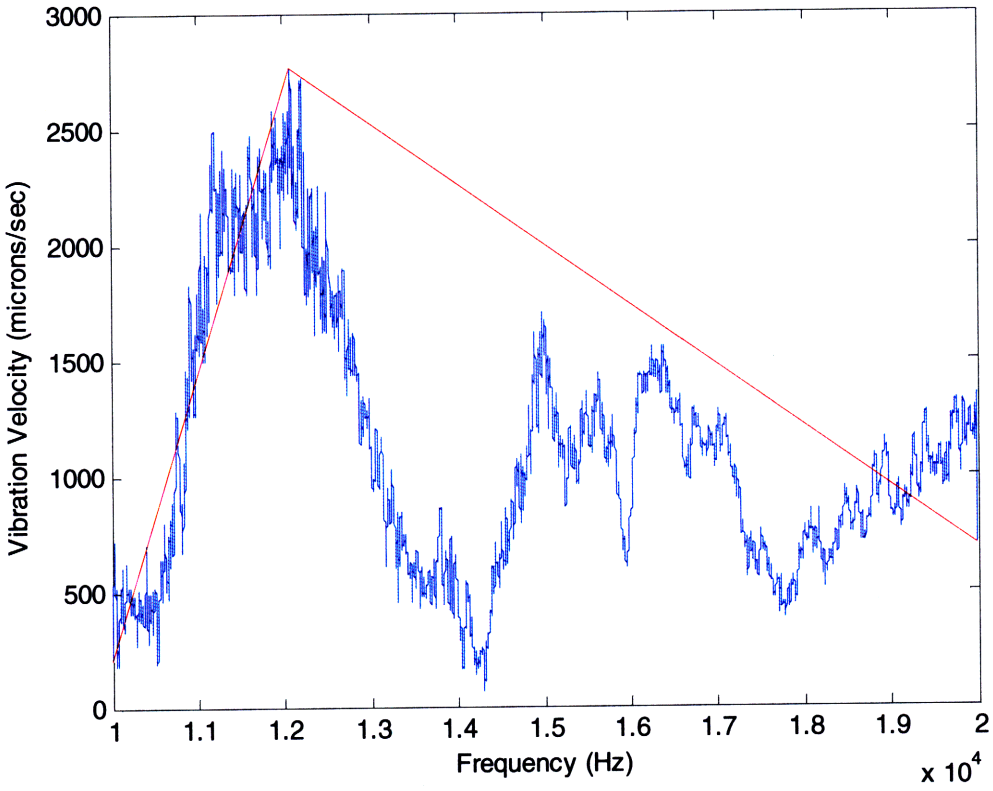
Doherty Simulated Landmine Variant 1&2 and Clutter Objects Velocity Profiles

It can be seen above that the nomenclature target signature is appropriate given the distinctive shape of the frequency response of the simulated landmines labeled 'Target1' and 'Target2' in comparison to the clutter objects used. Also the overall characteristics of the target signatures seem consistent across different frequency bands. Furthermore, the target signatures associated with Target 1 and Target 2 seem to have a consistent range of amplitudes within each data set as well as possess wide bandwidth features in general. The differences in characteristics between target signatures within these data sets will be exploited to discriminate targets from clutter. Although the identity of a target signature may be apparent to an observer, in order to eliminate the possibility of user bias, an algorithm with no operator interaction is required to identify targets and clutter.

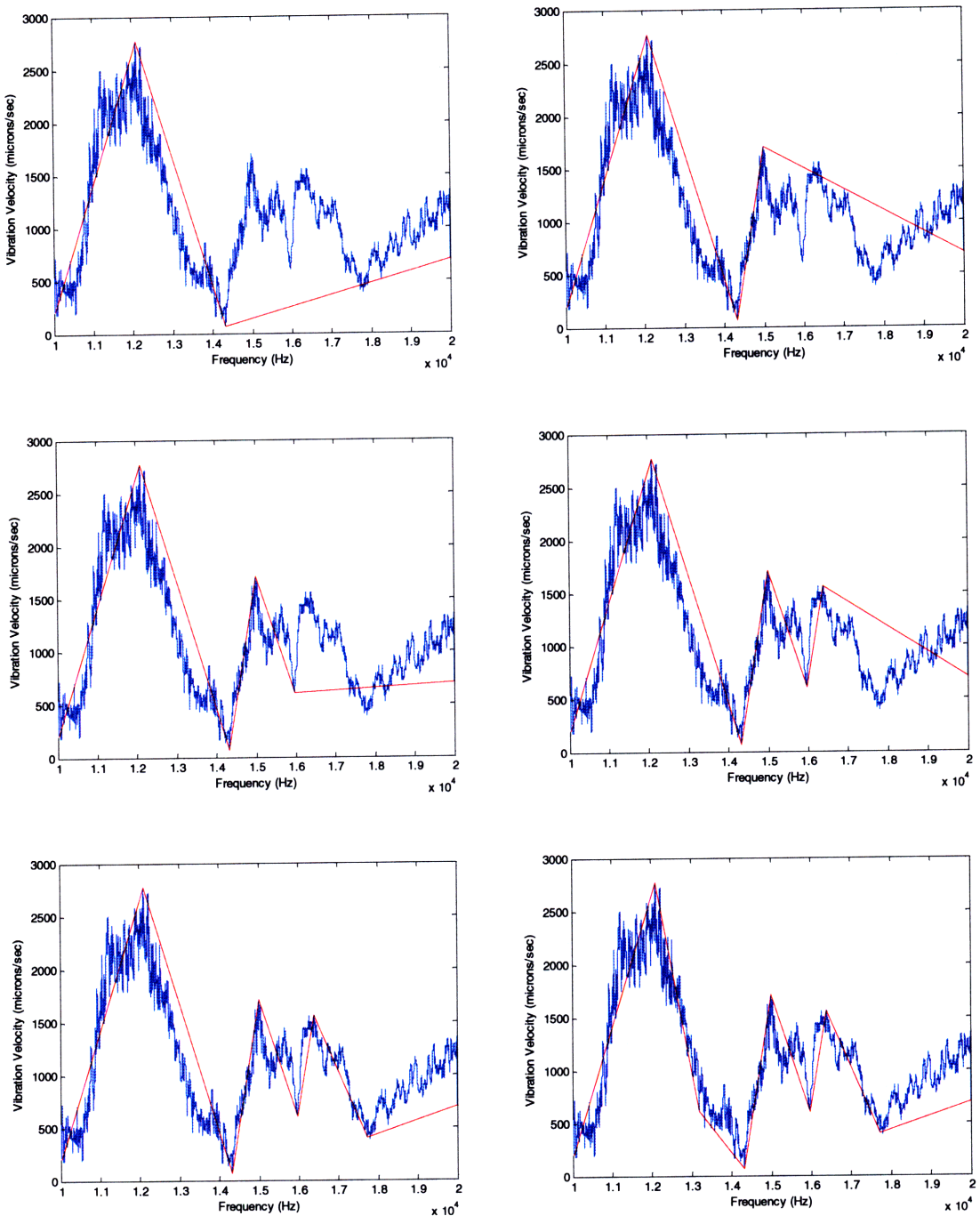
Feature Selection

The objective of an identification algorithm is to predict whether an unknown object is a landmine or clutter without prior knowledge or user input; this classification is done by a regression analysis which correlates parameters with the identity of known objects. A consistent methodology to select and numerically estimate parameters from the target signatures is needed in order to perform this regression analysis. Using a derivative form of a peak picking algorithm developed by Wallace for mass spectral data analysis, features from individual target signatures are identified with the MATLAB algorithm 'id_compute_trial3_JD.m' (see Appendix A) [19]. This recursive process begins by creating a line between the start and end points of the target signature. The point on the original target signature with amplitude furthest in vertical distance from this line is then identified (as opposed to perpendicular distance as in Wallace version) [19].

This point is then designated a 'strategic point' and used to draw a linear approximation of the target signature by connecting the start and end points with any intermediate strategic points in order of their frequency as demonstrated below with a sample target signature (from Target1):

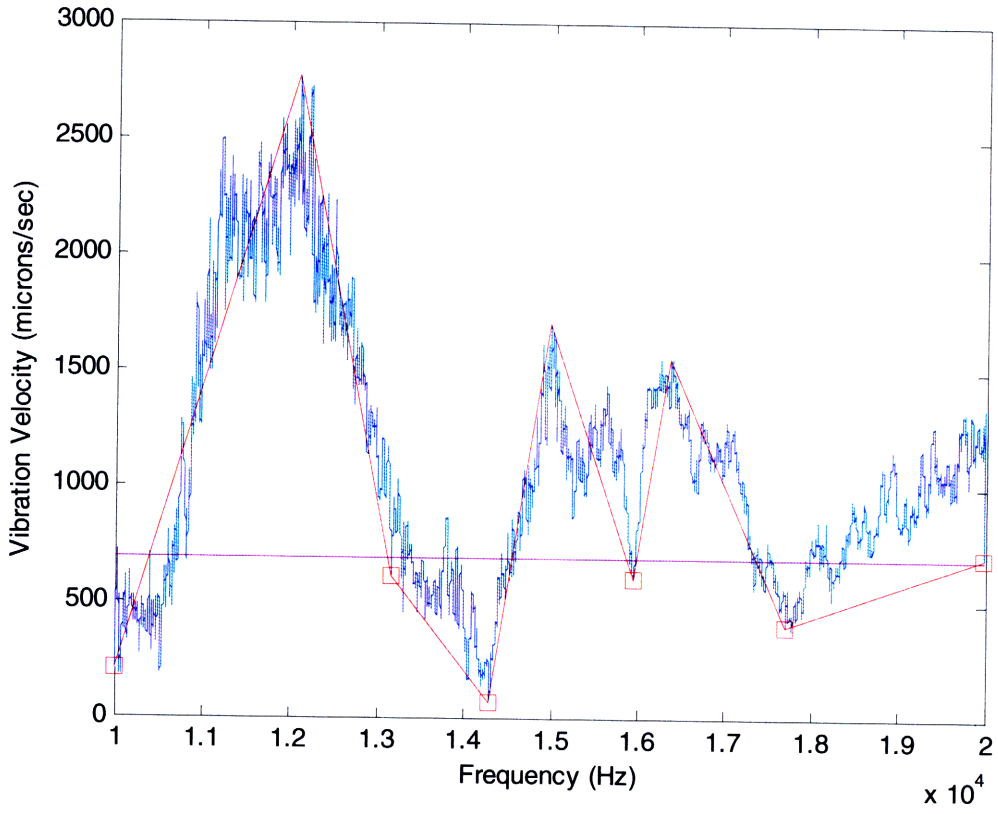


This process continues with additional strategic points selected in this same manner until all the points of the original target signature are within a certain preset threshold distance from the red line which connects the strategic points. Successive steps of this iterative process are shown below:



Individual features of the target signature are then selected by identifying strategic points that have amplitude less than a preset selection threshold. These strategic points are referred to as marker points. The first and last strategic points are also classified as marker points regardless of their amplitude. The marker point selection threshold and the

marker points are illustrated in the figure below with the threshold in magenta and the marker points represented by red boxes.

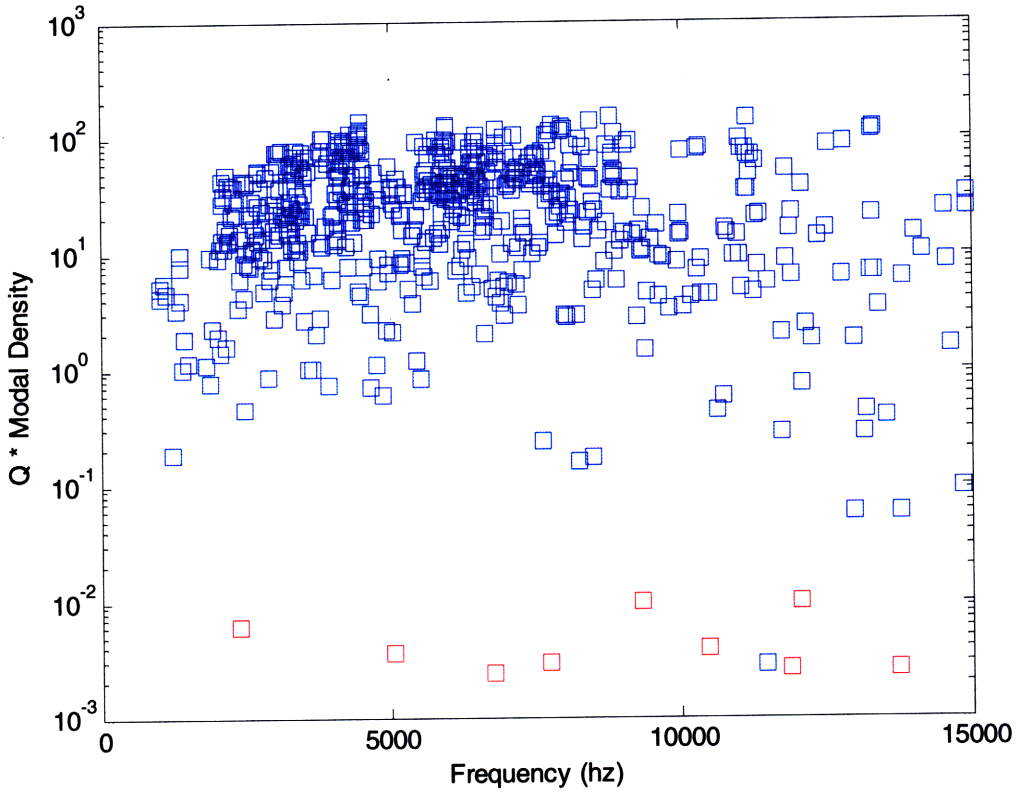


The regions between marker points are designated to be part of a single feature of the target signature. The linear approximation shown in red derived above is used to approximate the shape of the feature. Parameters of the individual features are then estimated with the results used to categorize targets and clutter based on a numerical analysis of estimated parameters rather than a subjective evaluation of the target signatures by the operator.

Parameter Estimation

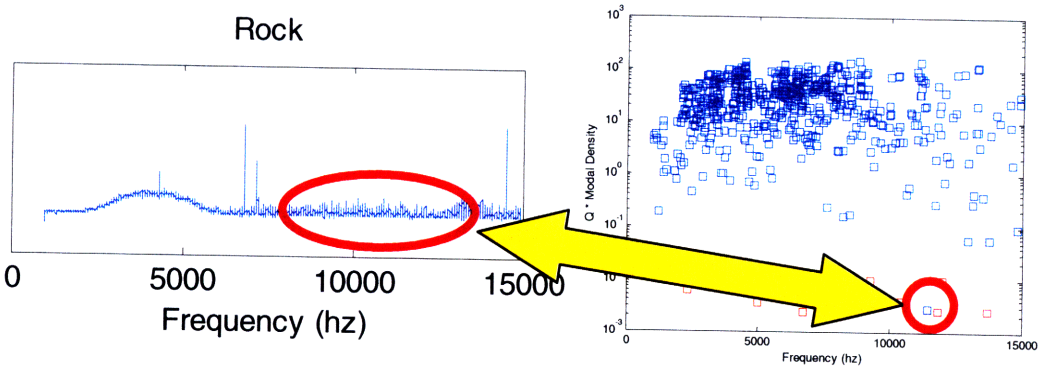
In order to accomplish this, a set of parameters that highly correlates to the identity of the object must be identified and estimated. On casual observation, the landmine target signatures have a smoother overall shape than the clutter objects. This

qualitative observation is roughly equivalent to the bandwidths of the features from the landmine target signatures having greater numerical value than those from clutter objects. This observation can be used by comparing the values of parameters dependent on bandwidth such as Q and modal density (Q is the frequency of a feature's maximum amplitude divided by its bandwidth and modal density is the number of peaks of a feature divided by its bandwidth). The frequency of the maximum amplitude of each feature versus the logarithm of Q times modal density is plotted below for the Haupt data set.



In the above graph, each feature of the data set collected by Haupt is represented by a box. The red boxes represent individual features from the two target signature trials and the features from clutter are represented by blue boxes. There is a linear separation between the landmines and the majority of the clutter (the single exception belonging to a

broad feature of the rock that can be discounted by its other attributes such as its low amplitude).



However, this classification methodology did not linearly separate landmines and clutter for all the sets of data collected. This is attributable to the fact that this data set had a small number of trials providing few opportunities for an anomalous trial to fail to separate. Although the target signatures' value for Q times modal density has not provide a perfect discriminator for landmines and clutter in all cases, it does serve as a foundation upon which other parameters can be further included.

The SVM uses a training matrix of parameters organized by column and trials organized by row. The SVM imposes constraints on how these parameters are chosen. The rows must have the same number of columns, and entries cannot be blank. Below is a sample training matrix of two trials with three parameters per trial.

Trial 1 Parameter 1	Trial 1 Parameter 2	Trial 1 Parameter 3
Trial 2 Parameter 1	Trial 2 Parameter 2	Trial 2 Parameter 3

Sample Training Matrix of N trials and N parameters

If objects are compared trial by trial, each target signature must be its own row. This precludes using parameters from every feature per target signature because the number of features selected per target signature is not constant (as seen in Appendix B).

Alternatively each row of the training matrix could correspond to an individual feature instead of a target signature; however, isolating and comparing individual features inhibits the classification of objects based on their target signatures overall characteristics.

In order to conform to the training matrix size restrictions, only the feature or features with the largest area and highest amplitude from each target signature were used. Due to their size, these features are less susceptible to noise distortions and better represent the characteristics of the target signature as a totality. Parameters from these two features as well as parameters of the target signature as a whole were then estimated and used to generate that target signature's respective row in the training matrix.

The parameters that were ultimately included in the training matrix were selected on the basis of whether their inclusion or exclusion marginally increased or decreased the accuracy of the SVM classification results discussed later. Using this criterion, the following parameters were selected:

1. Logarithm of the product of Q and modal density for the feature with the largest area
2. Logarithm of the product of Q and modal density for the feature with the greatest amplitude
3. Area under the target signature curve
4. Total number of features selected for the entire target signature
5. Weighted average with respect to the bandwidth of each feature of the logarithm of Q times modal density of all the target signature's features

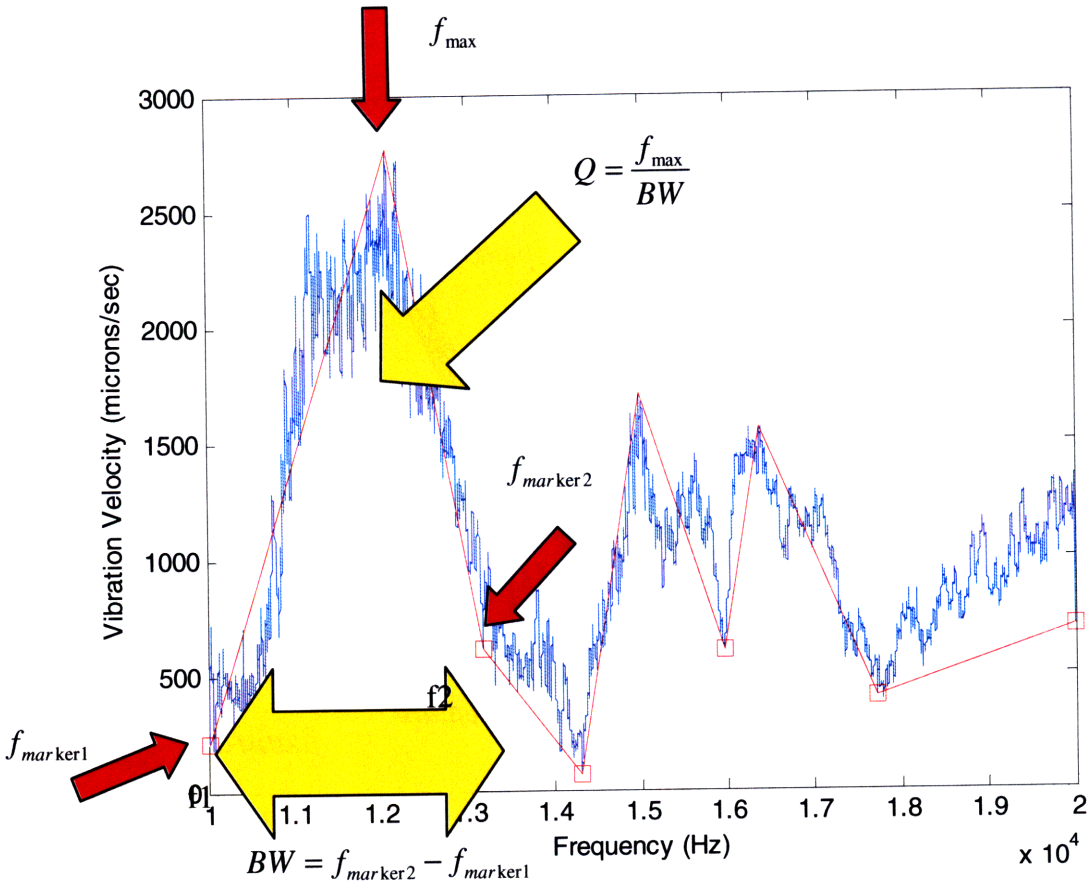
6. Weighted average with respect to the maximum amplitude of each feature of the logarithm of Q times modal density of all the target signature's features
7. Weighted average with respect to the area of each feature of the slope of each feature from its minima to maxima

These selected parameters were a subset of a larger group of parameters that also included the bandwidth, modal density, Q, total number of peaks, the maximum absolute value slope of the feature, the peak value of the feature, the area of the feature, the ratio of the area of the feature to the area of the entire target signature, and weighted averages with respect to the features' bandwidth, amplitude, and area of these individual parameters among others. The MATLAB algorithm 'id_compute_trial3_JD.m' estimated the value of each of these parameters.

The quality factor or Q of the target physically represents the amount of dampening present or the number of oscillations that occur before the amplitude of vibration attenuates to a negligible value [37 38]. In this instance we would anticipate rigid incompressible targets to have a high Q and the simulated landmines to have lower Q due to the compliances of their casings. Q was estimated by dividing the frequency of the maximum value of a feature by its bandwidth. The bandwidth was approximated as the difference between the frequencies of a feature's marker points.

$$Q = \frac{f_{\max}}{BW}$$
$$BW = f_{marker2} - f_{marker1}$$

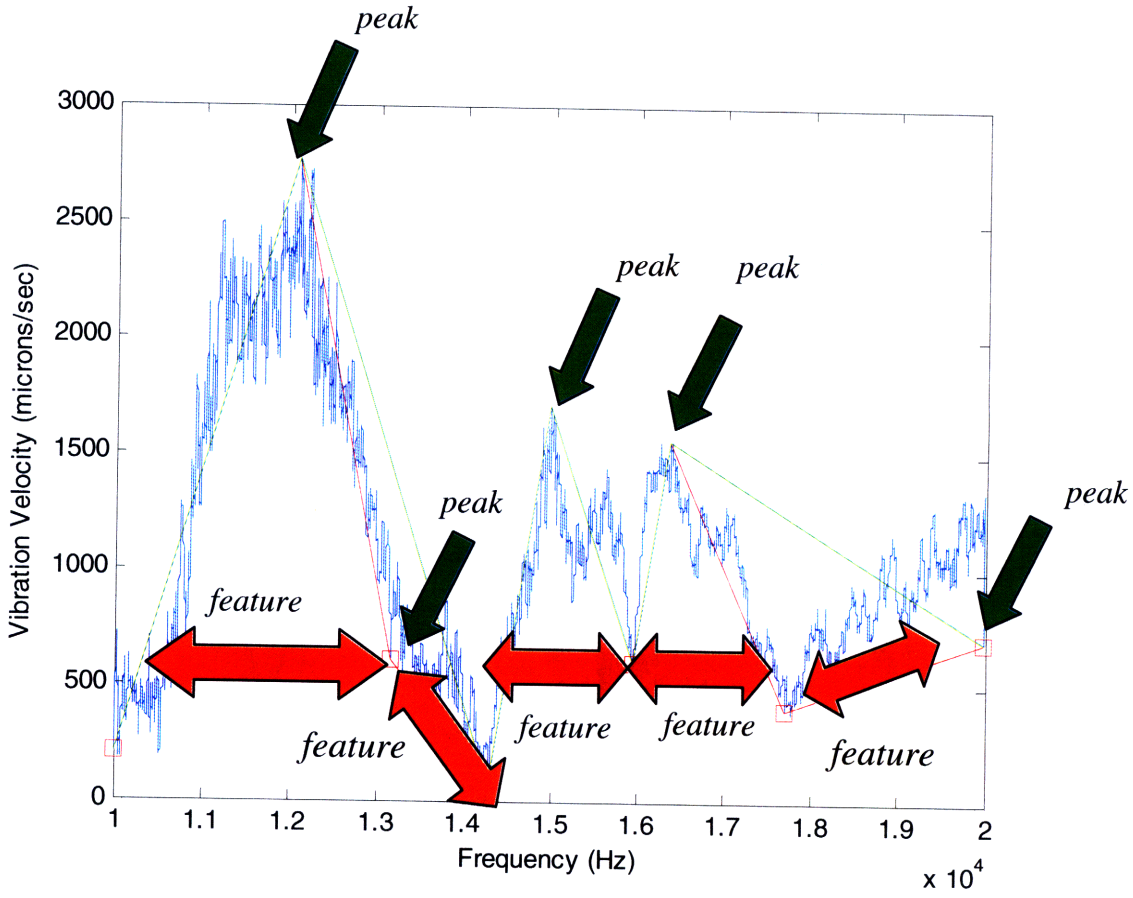
A graphical representation of the process used to estimate these parameters is shown below with the relevant frequencies designated with red arrows.



The feature's modal density is its number of local maxima divided by its bandwidth. Since the number of local maxima of the target signature is artificially inflated from noise, the maxima of a linear approximation of the target signature are used instead. This linear approximation is derived with the same methodology used to select features but with different threshold values. In this instance each of the features has at most a single peak designated by the green arrows with the features designated with orange arrows. In the event a feature does not have any peaks, an endpoint is considered to be a peak.

$$Modal \quad Density = \frac{Number\_of\_Peaks}{BW}$$

This linear approximation is superimposed in green over the previous example.



The area under the curve for each feature or target signature is represented by the sum of all its points. The total number of features per target signature is one less than the number of marker points.

Once parameters relevant to discriminating between targets and clutter have been identified and estimated, the algorithm 'svm_variables_helper.m' organizes the training matrix of these parameters and a column vector of binary solutions of one for targets and zero for clutter for each row of the training set. The SVM performs a regression analysis of this training set. This analysis is used to classify unknown sets of parameters.

Classification with Machine Learning Algorithm

The regression and classification is performed with a Support Vector Machine developed by Steve Gunn of the University of Southampton [15]. Regression analysis is conducted by mapping each row of parameters from the training matrix. These points are

separated into two classes, in this case landmines and clutter, with a binary column matrix representing ground truth. A boundary region between the two classes is generated by the algorithm 'svc.m' (see Appendix A) based on user inputs. Unidentified objects are then classified based on where their parameters are mapped in relation to the boundary region.

The SVM is better suited for regression and classification of sparse data sets than methodologies that minimize error in the training set generating a 'best fit' that often generalizes poorly for unseen data. Structural Risk Minimization minimizes an upper bound on the expected risk, as opposed to Error Risk Minimization which minimizes the error on the boundary region around the training data [15]. It is this difference which equips SVM with a greater ability to generalize, which is the goal in statistical learning [15]. With a training set that is closer in size to the total population, the difference between these two regression methodologies would be diminished. For sparse data sets, the SVM is considered superior [15].

Since the total number of trials is relatively small (59), a 'jack-knife' technique is used rather than having a dedicated set of trials for training the SVM and another set for testing [39 40]. Using this technique, the entire set of trials comprises the training set except for a single trial [39 40]. The lone trial is then classified based on a regression analysis of all the other trials [39 40]. This 'jack-knife' technique is repeated for every individual trial.

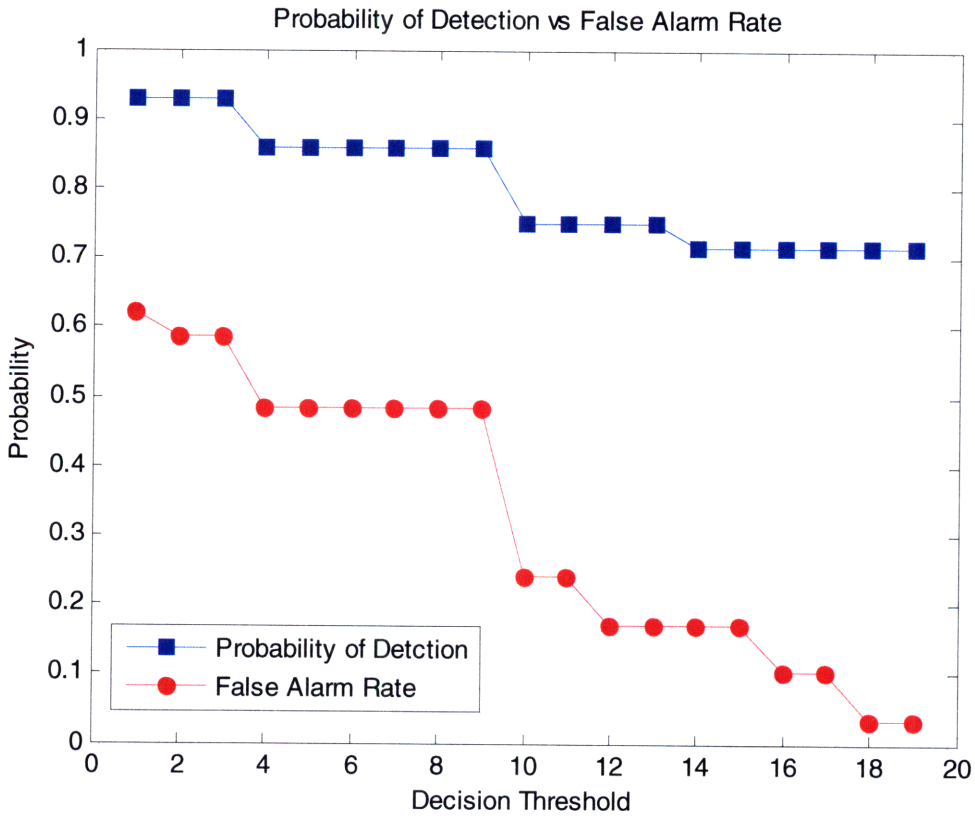
The algorithm 'svc.m' maps parameters of the training set in n-dimensional space (n being the number of parameters selected per target signature) and creates a linear, polynomial, radial, sinusoidal, or other nonlinear boundary region. The individual trial

that is excluded from the training set is classified on the basis of whether the mapping of its parameters falls within the boundary region. The operator inputs the type of function for the boundary of the decision region and the bias, which is the amount of tolerance for error within the training set the support vector machine will tolerate [15]. By allowing this user input, the SVM methodology takes advantage of the user's intuition of the degree of likelihood that members of the training data are misclassified. If the user has a high degree of confidence in the precision of the training set, a higher order function with low bias should be selected [15]. It is unlikely that the 29 clutter trials accurately reflect the universe of possible urban clutter target signatures. For this reason, the boundary regions used were confined to linear or low order polynomials with a high tolerance for error. In this case, the parameters chosen for the training matrix are believed to separate the landmine and clutter target signatures linearly with the exception of area under the curve which is second order.

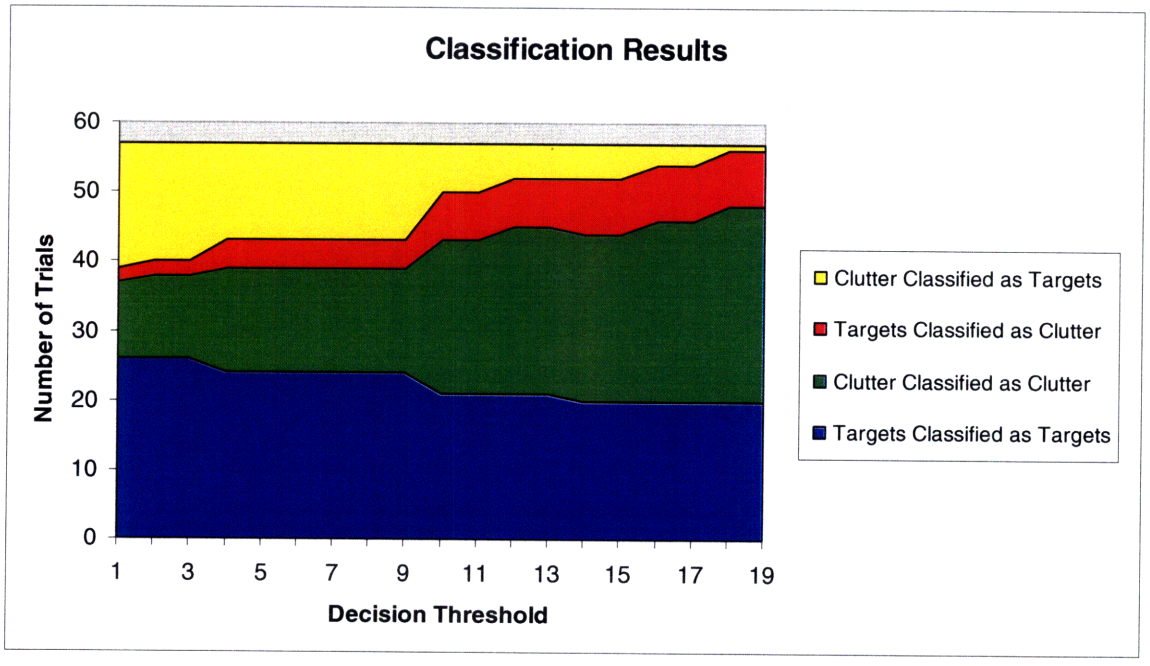
The type of function chosen for the boundary region, slack, and various threshold levels for the peak picking algorithm described earlier are all user-defined and influence the accuracy of the SVM. The algorithm 'svm_avg_butter.m' (see Appendix A) classifies a data set based on a single set of user-defined values for the threshold levels of the peak picking function, threshold for selection of marker points, type of function for the boundary region, and the amount of bias. A trial and error process was used to determine which combination of user input settings most accurately classified unknown objects. No individual set of these threshold values and SVM settings was found that provided a high probability of detection and low false alarm rate.

However by combining the results derived from several different variations of user settings, the accuracy of these predictions was improved. Adding the binary output from several iterations of the SVM produces a range of values. This provides an opportunity for the user to set a decision threshold for classifying an object as a landmine or clutter. The decision threshold is the number of individual SVM iterations that are required to independently classify an object as a landmine to predict that an object is a landmine. Control of the decision threshold allows the operator to trade off probability of detection for fewer false alarms.

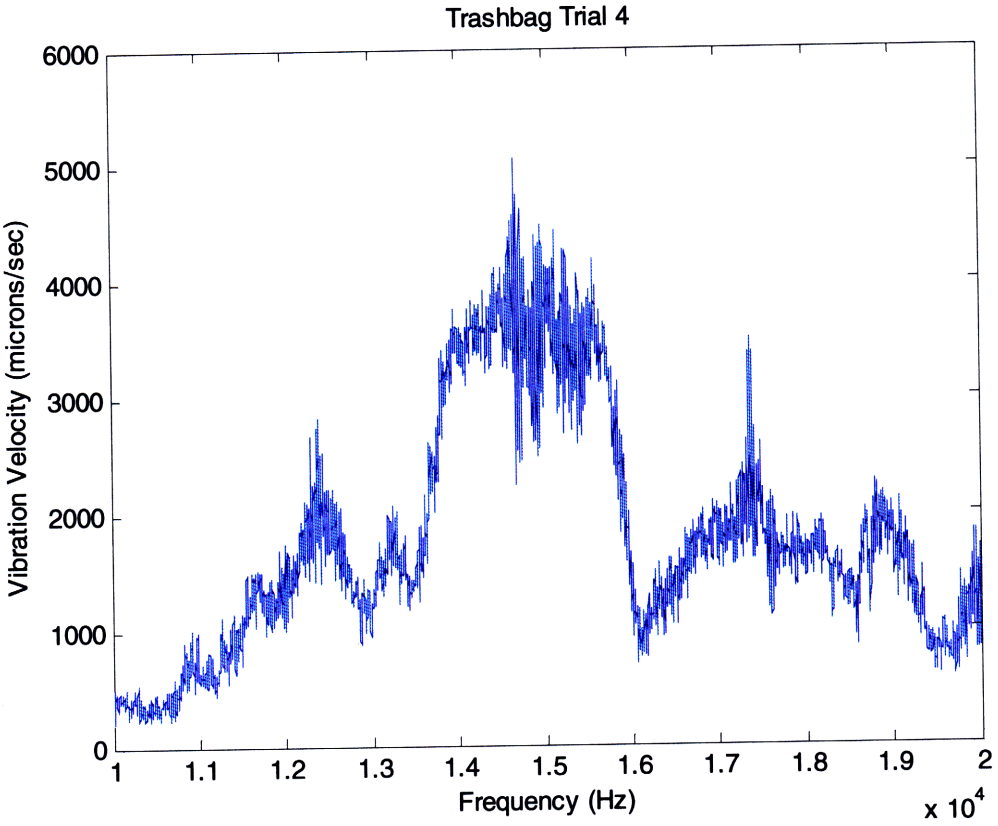
The results from a sparse data set (containing 28 target signatures and 29 clutter signatures provided in Appendix B) was used to test this methodology. The results from each trial are compared to ground truth in order to compute probability of detection and false alarm rate seen in the graph below. In this case ten different SVM iterative runs were used with each signature given a score from one to nineteen (one corresponding to all the SVM runs classifying the object as clutter and nineteen being all SVM runs classifying the object as a target). The decision threshold value that designates the cutoff score for an object to be classified as a target is user defined.



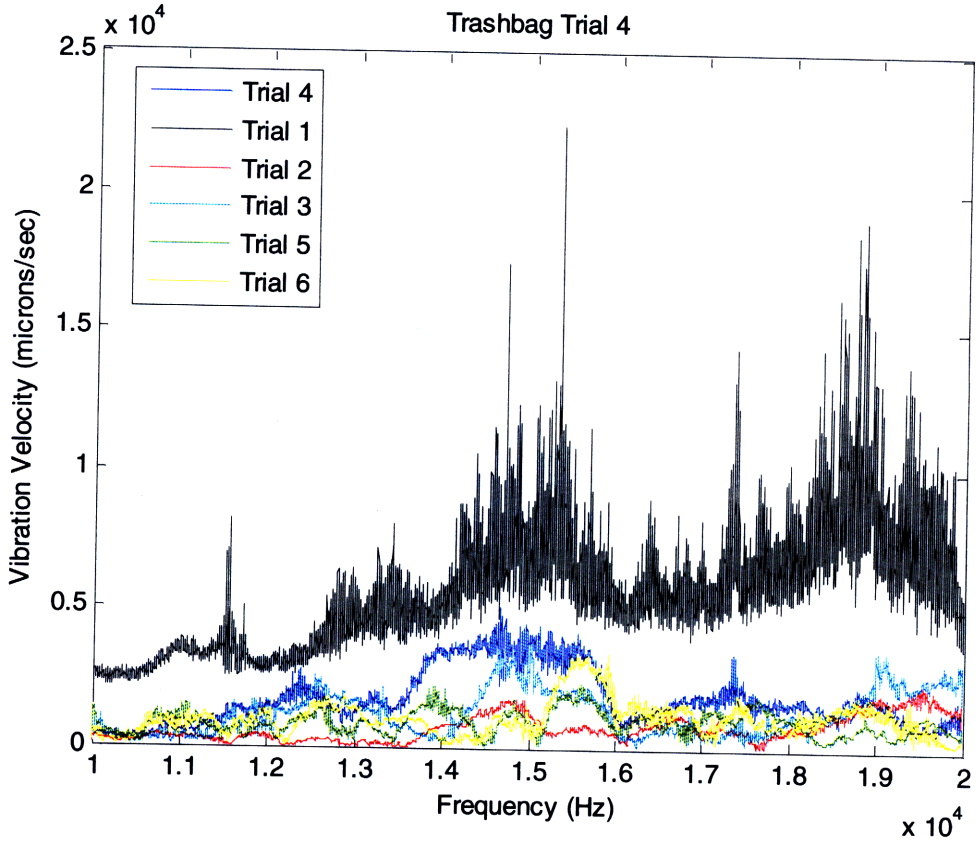
Since this data set is sparse, the results of the SVM classification are presented below in absolute numbers in lieu of percentages:



It is worth noting that for the highest decision threshold; there was only one false alarm belonging to the target signature of the fourth trial of the trashbag shown below:



Qualitatively, it seems reasonable for this target signature to be misclassified as an landmine. This trashbag's target signature has smooth features with high bandwidth and amplitude values consistent with the landmine target signatures. It does not have any obvious distinguishing features that are inconsistent with a landmine or possibly attributable to noise. The trashbag was partially filled with office trash that shifted between trials making this trial unrepeatable once the bag has been moved. However in comparison to other trashbag trials, this trial signature did not resemble all the other trashbag vibration signatures.



Conclusions

The results of the classification technique demonstrate that the phenomenology used by Sabatier and Haupt for landmine detection can be used for discrimination in an urban environment. Operationally, the frequency of landmine encounters is likely to be orders of magnitude lower than that of urban clutter. This suggests that the lowest false alarm rate possible is desired; otherwise the system would generate an alarm almost constantly. Using the highest decision threshold in order to reduce false alarms, the standoff acoustic/laser methodology yielded a probability of detection of 0.71 and a false alarm rate of 0.03.

The data processing and classification methodology seem to function well when used with a data set with high signal to noise and few anomalies. However, generating the precision measurements this requires was consistently problematic in a laboratory

environment and would likely be much more difficult on the battlefield. To a large degree this was caused by poor instrumentation equipment that can be replaced.

The PDV-100 was suspected to be the source of a great deal of noise. The PDV-100 had observably degraded performance with time. This is possibly due to degrading battery performance. Furthermore, its lack of sensitivity necessitated the application of reflective tape to the surface of the targets. The use of reflective tape on the target raises the possibility that the experimentally generated target signatures are partially a function of the responsiveness to the sound source of the reflective tape as well as the target. The tape's influence was minimized by firmly adhering it to as small a surface area as possible, but without an alternate measurement method where all other factors are kept constant it is impossible to know the level of noise contributed by the tape. Obviously, the requirement that the target have a highly reflective surface makes the entire methodology worthless; therefore, it is necessary to have a more sensitive LDV system as a prerequisite for an operational system.

Also the sample size of clutter objects that were tested was limited in comparison to the size of the population. However, the ability to add trials to the training set as the system continues to be developed would reduce false alarms from anomalous clutter objects and provide flexibility in engaging emerging threats.

Recommendations

Although the instrumentation equipment with which experimental data was collected was not as reliable as would be desired in an operational system, this experiment did demonstrate that the acoustic/laser technique could detect targets and discriminate targets from clutter with a reasonable degree of accuracy. It is hoped that

further study and better equipment could enhance the performance of the acoustic/laser detection methodology in these areas.

Using a Parametric Acoustic Array in an operational system would significantly enhance its capabilities. The PAA would provide increased sound pressure levels that would generate higher amplitude vibration in the target. This would enhance signal strength without the need to improve any other sensors or methods used.

In an operational system, a more sophisticated LDV could enhance system performance by improving the signal to noise ratio. Furthermore, a more sensitive LDV is necessary to eliminate the need for the target to have a highly reflective surface. The requirement that the target have a reflective surface negates any operational utility of the system. Polytec manufactures a more sophisticated LDV system the PSV. The PSV was tested in the lab and generated improved signal to noise levels in comparison to the PDV-100. The PSV also had its own self-contained data acquisitions device that would eliminate the need for the Wavebook which troublingly generated the occasional corrupted '.dsc' header file. Commercially available multipixel LDV and scanning LDV systems are already on the market [41 42]. The VibroMet Multi Beam Laser Doppler Vibrometer manufactured by MetroLaser is potentially suitable for use in the acoustic/laser landmine detection apparatus [41]. The VibroMet can measure multiple locations on a target with high sensitivity simultaneously [41]. It is hoped that a military LDV system would have even superior sensitivity than the PSV or VibroMet systems providing enhanced signal to noise performance and eliminating the requirement that the target be highly reflective.

The classification methodology could also be improved upon to enhance performance. It is likely that there exist clutter objects that were not tested in this experiment that would generate a false alarm unless additional parameters are used to supplement the training set.

It is hoped that with the above described improvements an operationally useful system will emerge.

Bibliography

1. R.W. Haupt, K.D. Rolt, "Standoff Acoustic Laser Technique to Locate Buried Land Mines," *Lincoln Laboratory Journal* Vol. 15 pp 3-22, 2005.

2. R.W. Haupt, "Standoff acoustic-to-seismic landmine detection," *Proc. SPIE* Vol. 5415, pp 100-110,

3. N. Xiang, J.M. Sabatier, "An experimental study on antipersonnel landmine detection using acoustic-to-seismic coupling," *Journal of Acoustic Society of America* Vol. 113, pp 1333-1341, 2003.

4. J.M. Sabatier, N. Xiang, "An Investigation of Acoustic-to-Seismic Coupling to Detect Buried Antitank Landmines," *IEEE Transactions on Geoscience and Remote Sensing,* Vol. 39, pp 1146-1154, 2001.

5. S.W. Kercel, "Acoustic resonance for nonmetallic mine detection," *Proc. SPIE* Vol. 3392, pp 848-860, 1998.

6. S.W. Knight, C. DiMarzio, W. Li, D.O. Hogenboom, G. Sauermann, "Laser-induced acoustic detection of buried objects," *SPIE* Vol. 3392, pp 841-847, 1998.

7. R.B. Fair, M. Pollack, V. Pamula, "MEMS devices for detecting the presence of explosive material residues in mine fields," *Proc. SPIE* Vol. 3392, pp 409-417, 1998.

8. W.R. Scott, C. Schroeder, J.S. Martin, "An acoustic-electromagnetic sensor for locating land mines," *SPIE* Vol. 3392, pp 176-186, 1998.

9. W. R. Scott, J.S. Martin, G.D. Larson, "Experimental Model for a Seismic Landmine Detection System," *IEEE Transactions on Geoscience and Remote Sensing* Vol. 39, pp 1155-1164, 2001.

10. P.J. Westervelt, "Parametric Acoustic Array," *Journal of the Acoustical Society of America* Vol. 35, pp 535-537, 1963.

11. M.A. Biot, "Theory of Propagation of Elastic Waves in a Fluid-Saturated Porous Solid. I. Low Frequency Range," *Journal of the Acoustical Society of America* Vol. 28, pp 168-178, 1956.

12. N. Cressie, A.B. Lawson, "Bayesian hierarchical analysis of minefield data," *Proc. SPIE* Vol. 3392, pp 930-940, 1998.

13. D.M. Donskoy, "Nonlinear vibro-acoustic technique for landmine detection," *Proc. SPIE* Vol. 3392, pp 211-217, 1998.

14. D.M. Donskoy, "Detection and discrimination of nonmetallic land mines," *Proc. SPIE* Vol. 3392, pp 239-246, 1998.

15. S.R. Gunn, "Support Vector Machines for Classification and Regression," Technical Report, University of Southampton. May 10, 1998.

16. O.O. Bilukha, M. Brennan, B.A. Woodruff, "Death and Injury from Landmines and Unexploded Ordinance in Afghanistan," *Journal of the American Medical Association* Vol. 290, pp 650-653, 2003.

17. A.J. Dumian, C.M. Rappaport, "Enhanced Detection and Classification of Buried Mines with an UWB Multistatic GPR," *IEEE* 2005.

18. V. George, T.W. Altshuler, E.M. Rosen, "DARPA backround clutter data collection experiment excavation results," *SPIE* Vol. 3392, pp. 1000-1011, 1998.

19. W.E. Wallace, A.J. Kearsley, C.M. Guttman, "An Operator-Independent Approach to Mass Spectral Peak Identification and Integration," *Analytical Chemistry* Vol. 76, pp. 2246-2452, 2004.

20. S.W. Kercel, R.S. Burlage, D.R. Patek, C.M. Smith, "Novel Methods for Detecting Buried Explosives," Technical Report, Oak Ridge National Laboratory. April 10, 1997.

21. E.M. Rosen, K.D Sherbondy, "Performance Assessment of Mine Detection Systems," *Proc. SPIE* Vol. 4038, pp. 1225-1236, 2000.

22. E.M. Rosen, K.D Sherbondy, J.M. Sabatier, "Performance Assessment of Blind Test Using the University of Mississippi's Acoustic/Seismic Laser Doppler Vibrometer (LDV) Mine Detection Apparatus at Fort A.P. Hill," *Proc. SPIE* Vol. 4038, pp. 656-666, 2000.

23. M. Yoneyama, J. Fujimoto, "The audio spotlight: An application of nonlinear interaction of sound waves to a new type of loudspeaker design," *Journal of the Acoustical Society of America* Vol. 73, pp. 1532-1536, 1983.

24. M.B. Bennett, D.T Blackstock, "Parametric array in air," *Journal of the Acoustical Society of America* Vol. 57, pp. 562-568, 1975.

25. M.S. Korman, J.M. Sabatier, "Nonlinear acoustic technique for landmine detection," *Journal of the Acoustical Society of America* Vol. 116, pp. 3354-3369, 2004.

26. www.globalsecurity.org

27. www.icbl.org

28. C.E. Baum, *Detection and Identification of Visually Obscured Objects*, Taylor & Francis Publishing: Philadelphia, PA. 1999.

29. www.janes.com

30. www.polytec.com

31. www.eminence.com

32. S.H. Yu, A. Gandhe, T.R. Witten, R.K. Mehra, "Physically Based Method for Automatic Mine Detection using Acoustic Data- A Transmission Zero Approach," *Proc. SPIE* Vol. 4742, pp. 701-708, 2002.

33. A. Zagrai, D. Donskoy, A. Ekimov, " Resonance Vibrations of Buried Landmines," *Proc. SPIE* Vol. 5415, pp. 21-29, 2004.

34. H.D. vom Stein, H.J. Pfeifer, "A Doppler Difference Method for Velocity Measurements," *Metrologia* Vol. 5, pp. 59-61, 1969.

35. www.iotech.com

36. S.H. Yu, A. Gandhe, T.R. Witten, R.K. Mehra, "Automatic Mine Detection Based on Multiple Features,"

37. B. Cromwell, "Vibrations and Waves," *Light and Matter Online Text Series*, Chap. 2, 2006.

38. W.M. Siebert, *Circuits, Signals, and Systems*, MIT Press: Cambridge, MA. 1986.

39. S.C. Sharma, C. Giaccotto, "Power and robustness of jackknife and likelihood-ratio tests for grouped heteroscedasticity," *Journal of Econometrics* Vol. 29, pp. 343-372, 1991.

40. J. Shao, C.F.J. Wu, "A General Theory for Jackknife Variance Estimation," *The Annals of Statistics* Vol. 17, pp. 1176-1197, 1989.

41. www.metrolaserinc.com

42. M.J. Connelly, P.M. Szecowka, R. Jallapuram, S. Martin, V. Toal, M.P. Whelan, "Multipoint laser Doppler vibrometry using holographic optical elements and a CMOS digital camera," *Optics Letter* Vol. 33, pp. 330-332, 2008.

Appendix A: MATLAB Code

```
function [trial,money,pred_all] =
crowd_wisdom_butter_JD(target1_info,target2_info,trashbag_info,cylinder
_info,sodacan_info,pepsibottle_info,foamcup_info,crushedcan_info)



combo1 = [.3 .6 .25; .4 .6 .25];

multiplier4 = .7;
multiplier5 = .7;

money_k = 1;


for j = 1:length(combo1(:,1))
    multiplier1 = combo1(j,1);
    multiplier2 = combo1(j,2);
    multiplier3 = combo1(j,3);
%     multiplier4 = combo1(j,4);
%     multiplier5 = combo1(j,5);
    for laser = 1:3
        tic
        'laser'
        laser
        'T1'
        for i = 1: length(target1_info)
%             i
            x_info = target1_info{i};
            [x_id] =
id_compute_butter_JD(x_info,multiplier1,multiplier2,multiplier3,multipl
ier4,multiplier5,laser);
            target1_id{i} = x_id;
        end
        'T2'
        for i = 1: length(target2_info)
%             i
            x_info = target2_info{i};
            [x_id] =
id_compute_butter_JD(x_info,multiplier1,multiplier2,multiplier3,multipl
ier4,multiplier5,laser);
            target2_id{i} = x_id;
        end
        'TB'
        for i = 1: length(trashbag_info)
%             i
            x_info = trashbag_info{i};
            [x_id] =
id_compute_butter_JD(x_info,multiplier1,multiplier2,multiplier3,multipl
ier4,multiplier5,laser);
            trashbag_id{i} = x_id;
        end
        'CY'
```

```
        for i = 1: length(cylinder_info)
%             i
            x_info = cylinder_info{i};
            [x_id] =
id_compute_butter_JD(x_info,multiplier1,multiplier2,multiplier3,multipl
ier4,multiplier5,laser);
            cylinder_id{i} = x_id;
        end
        'SC'
        for i = 1: length(sodacan_info)
%             i
            x_info = sodacan_info{i};
            [x_id] =
id_compute_butter_JD(x_info,multiplier1,multiplier2,multiplier3,multipl
ier4,multiplier5,laser);
            sodacan_id{i} = x_id;
        end
        'PB'
        for i = 1: length(pepsibottle_info)
%             i
            x_info = pepsibottle_info{i};
            [x_id] =
id_compute_butter_JD(x_info,multiplier1,multiplier2,multiplier3,multipl
ier4,multiplier5,laser);
            pepsibottle_id{i} = x_id;
        end
        'FC'
        for i = 1: length(foamcup_info)
%             i
            x_info = foamcup_info{i};
            [x_id] =
id_compute_butter_JD(x_info,multiplier1,multiplier2,multiplier3,multipl
ier4,multiplier5,laser);
            foamcup_id{i} = x_id;
        end
        'CC'
        for i = 1: length(crushedcan_info)
%             i
            x_info = crushedcan_info{i};
            [x_id] =
id_compute_butter_JD(x_info,multiplier1,multiplier2,multiplier3,multipl
ier4,multiplier5,laser);
            crushedcan_id{i} = x_id;
        end


% [FA5 FA6 FA7 FAV3 FAV4 FAV5 FM5 FM6 FM7 FMV1 FMV2 G3 G4 G10 BW1 BWV4
M1]

        combo2 = [2 .001 1; 1 .001 1]

        for k = 1:length(combo2(:,1))
            order = combo2(k,1);
            slack = combo2(k,2);
            if combo2(k,3) == 1
                func = 'poly';
```

```
            else
                func = 'lin';
            end

            [predicted_temp,Pd_temp,False_Alarm_temp] =
svm_avg_butter(target1_id,target2_id,trashbag_id,foamcup_id,sodacan_id,
crushedcan_id,pepsibottle_id,cylinder_id,order,slack,func);

            prediction(:,j,k,laser) = predicted_temp;
            if j == 1 && k ==1
                pred_all = predicted_temp;
            else
                pred_all = pred_all+predicted_temp;
            end

            Pd(j,k,laser) = Pd_temp;
            False(j,k,laser) = False_Alarm_temp;


            mult_temp = [multiplier1 multiplier2 multiplier3
multiplier4 multiplier5]';

            trial{j,k,laser} =
struct('mult',mult_temp,'pred',predicted_temp,'Pd',Pd_temp,'False',Fals
e_Alarm_temp);

            for i = 1:length(Pd_temp)
                if Pd_temp(i) > .7 && False_Alarm_temp(i) < .3
                    money(money_k) =
struct('mult',mult_temp,'order',order,'slack',slack,'func',combo2(k,3))
;
                    money_k = money_k+1;
                end
            end
        end
    toc
    end
end

pred_all
```

```
function [target_identifier] =
id_compute_butter_JD(target_info,multiplier1,multiplier2,multiplier3,mu
ltiplier4,multiplier5,laser)

f = target_info.f;
if laser == 1
    data = abs(target_info.las1);
end
if laser == 2
    data = abs(target_info.las2);
end
if laser == 3
    data = abs(target_info.las1+target_info.las2);
end

data = data';
data_true = data;



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

start_freq = f(1);
start_pres = data(1);
start_index = 1;

end_freq = f(length(f));
end_pres = data(length(data));
end_index = length(f);

strat_freq(1) = start_freq;
strat_pres(1) = start_pres;
strat_index(1)= 1;

strat_freq(2) = end_freq;
strat_pres(2) = end_pres;
strat_index(2)= length(f);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



temp_index(1) = 1;
temp_index(2) = length(f);

max(data);
std(data);

dt = f(2)-f(1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ONLY USER DEFINED THRESHOLD SET BELOW:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
limit = multiplier1*6e-4;
limit_alt = multiplier1*max(data);

if limit > limit_alt
    limit = limit_alt;
end

limit2 = multiplier2*3e-4;
limit2_alt = multiplier2*mean(data);

if limit2 > limit2_alt
    limit2 = limit2_alt;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


i = 3;
counter = max(data);

while counter > limit
    line_data = (end_pres-start_pres)/(end_freq-start_freq)*(f-
start_freq)+start_pres;


    temp_data = abs(data(start_index:end_index)-
line_data(start_index:end_index));


    temp_index(i) = find(temp_data(1:length(temp_data)) ==
max(temp_data),1,'first')+start_index-1;
    temp_freq(i) = f(temp_index(i));
    temp_pres(i) = data(temp_index(i));


    % SORT TEMP POINTS; REORDER STRATEGIC POINTS
    temp_index = sort(temp_index);
    temp_index;
    for j = 1:length(temp_index)
        strat_index(j)= temp_index(j);
        strat_freq(j) = f(strat_index(j));
        strat_pres(j) = data(strat_index(j));
    end


    %SELECT NEXT SCANNING REGION BASED ON STRAT POINTS INTERVAL HAS
    %BIGGEST DISTANCE FROM LINE TO DATA
    for j = 1:(length(strat_index)-1)
        distance(j) = max(abs(data(strat_index(j):strat_index(j+1))-
(strat_pres(j)+(strat_pres(j+1)-strat_pres(j))/(strat_index(j+1)-
strat_index(j))*([strat_index(j):1:strat_index(j+1)]-
strat_index(j)))));
    end
```

```
    for j = 1:length(distance)
        if distance(j) == max(distance)
            start_index = strat_index(j);
            start_freq = f(strat_index(j));
            start_pres = data(strat_index(j));
            end_index = strat_index(j+1);
            end_freq = f(strat_index(j+1));
            end_pres = data(strat_index(j+1));
        end
    end
    i=i+1;

    counter = max(distance);
end

strat_freq;
strat_pres;



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% USE STRATEGIC POINTS TO PICK OUT FUNDAMENTAL SHAPES
% MEASURE BW, NUMBER OF MICROPEAKS, Q, AREA UNDER CURVE
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
bw_threshold = multiplier3*(max(data));

if bw_threshold <
min(data(floor(length(data)/10):floor(length(data)*.9)))
    bw_threshold = multiplier3*(max(data)-min(data))+min(data);
%     'bw alt'
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%



for i = 1:length(strat_freq)-1
    marker_slope(i) = abs((strat_pres(i+1)-
strat_pres(i))./(strat_freq(i+1)-strat_freq(i)));
end

marker_slope(length(marker_slope)+1) =
marker_slope(length(marker_slope));

bw_marker_freq(1) = strat_freq(1);
bw_marker_index(1) = strat_index(1);
bw_marker_pres(1) = strat_pres(1);

k = 2;

for i = 2:length(strat_freq)-1
    if strat_pres(i) < bw_threshold
```

```
%         if strat_pres(i) < strat_pres(i-1) && strat_pres(i) <
strat_pres(i+1)
            bw_marker_freq(k) = strat_freq(i);
            bw_marker_index(k) = strat_index(i);
            bw_marker_pres(k) = strat_pres(i);
            k=k+1;
%         end
    end
end

% 'point2'

bw_marker_freq(k) = strat_freq(length(strat_freq));
bw_marker_index(k) = strat_index(length(strat_freq));
bw_marker_pres(k) = strat_pres(length(strat_freq));


bw_marker_index = sort(bw_marker_index);
bw_marker_pres = data(bw_marker_index);
bw_marker_freq = f(bw_marker_index);

temp(1) = bw_marker_index(1);

k = 2;

for i = 2:length(bw_marker_index)
    if bw_marker_index(i) ~= bw_marker_index(i-1)
        temp(k) = bw_marker_index(i);
        k = k+1;
    end
end

bw_marker_index = temp;
bw_marker_pres = data(bw_marker_index);
bw_marker_freq = f(bw_marker_index);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

k = 1;

for i = 2:length(bw_marker_index)
%     for j = 1:length(f)
        tempmax = max(data(bw_marker_index(i-1):bw_marker_index(i)));
        bw_max_index(k) = find(data(bw_marker_index(i-
1):bw_marker_index(i)) == tempmax,1,'first')+bw_marker_index(i-1)-1;
        bw_max_freq(k) = f(bw_max_index(k));
        bw_max_pres(k) = data(bw_max_index(k));
%     end
    k = k+1;
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% USE STRAT PEAK PICKING CODE TO COUNT NUMBER OF PEAKS WITH
DIFFERENT/LOWER
% THRESHOLD
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


start_freq2 = f(1);
start_pres2 = data(1);
start_index2 = 1;

end_freq2 = f(length(f));
end_pres2 = data(length(data));
end_index2 = length(f);

strat_freq2(1) = start_freq2;
strat_pres2(1) = start_pres2;
strat_index2(1)= 1;

strat_freq2(2) = end_freq2;
strat_pres2(2) = end_pres2;
strat_index2(2)= length(f);

temp_index2(1) = 1;
temp_index2(2) = length(f);


i = 3;
counter2 = max(data);

while counter2 > limit2
    line_data2 = (end_pres2-start_pres2)/(end_freq2-start_freq2)*(f-
start_freq2)+start_pres2;
    temp_data2 = abs(data(start_index2:end_index2)-
line_data2(start_index2:end_index2));


    temp_index2(i) = find(temp_data2(1:length(temp_data2)) ==
max(temp_data2),1,'first')+start_index2-1;
    temp_freq2(i) = f(temp_index2(i));
    temp_pres2(i) = data(temp_index2(i));


    % SORT TEMP POINTS; REORDER STRATEGIC POINTS
    temp_index2 = sort(temp_index2);
    temp_index2;
    for j = 1:length(temp_index2)
        strat_index2(j)= temp_index2(j);
        strat_freq2(j) = f(strat_index2(j));
        strat_pres2(j) = data(strat_index2(j));
    end


    %SELECT NEXT SCANNING REGION BASED ON STRAT POINTS INTERVAL HAS
```

```
    %BIGGEST DISTANCE FROM LINE TO DATA
    for j = 1:(length(strat_index2)-1)
        distance2(j) = max(abs(data(strat_index2(j):strat_index2(j+1))-
(strat_pres2(j)+(strat_pres2(j+1)-strat_pres2(j))/(strat_index2(j+1)-
strat_index2(j))*([strat_index2(j):1:strat_index2(j+1)]-
strat_index2(j)))));
    end


    for j = 1:length(distance2)
        if distance2(j) == max(distance2)
            start_index2 = strat_index2(j);
            start_freq2 = f(strat_index2(j));
            start_pres2 = data(strat_index2(j));
            end_index2 = strat_index2(j+1);
            end_freq2 = f(strat_index2(j+1));
            end_pres2 = data(strat_index2(j+1));
        end
    end
    i=i+1;

%     pause
    counter2 = max(distance2);
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

num_peaks1 = zeros(size(bw_max_freq));

for k = 1:(length(bw_marker_index)-1)
    for i = 2:(length(strat_index)-1)
        if strat_pres(i) > strat_pres(i-1) && strat_pres(i) >
strat_pres(i+1) && f(strat_index(i)) > bw_marker_freq(k) &&
f(strat_index(i)) < bw_marker_freq(k+1)
            num_peaks1(k) = num_peaks1(k)+1;
        end
    end
end

num_peaks2 = zeros(size(bw_max_freq));

for k = 1:(length(bw_marker_index)-1)
    for i = 2:(length(strat_index2)-1)
        if strat_pres2(i) > strat_pres2(i-1) && strat_pres2(i) >
strat_pres2(i+1) && f(strat_index2(i)) > bw_marker_freq(k) &&
f(strat_index2(i)) < bw_marker_freq(k+1)
            num_peaks2(k) = num_peaks2(k)+1;
        end
    end
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

k = 1;

for i = 2:length(bw_marker_index)
    bw(k) = bw_marker_freq(i)-bw_marker_freq(i-1);
    Q(k) = bw_max_freq(k)/bw(k);
    modal_dens1(k) = num_peaks1(k)/bw(k);
    modal_dens2(k) = num_peaks2(k)/bw(k);
    if modal_dens1(k) == 0
        modal_dens1(k) = 1/bw(k);
    end
    if modal_dens2(k) == 0
        modal_dens2(k) = 1/bw(k);
    end
    k = k+1;
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


for k = 1:length(bw_max_freq)
    area1(k) = sum(data(bw_marker_index(k):bw_marker_index(k+1)))*dt;
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


for k = 1:length(bw_max_freq)
    spikiness1(k) = 0;
    spikiness2(k) = 0;
    if (bw_max_freq(k)-f(bw_marker_index(k))) == 0
        spikiness1(k) = (bw_max_pres(k)-data(bw_marker_index(k+1)))./(-
bw_max_freq(k)+f(bw_marker_index(k+1)));
    else
        spikiness2(k) = (bw_max_pres(k)-
data(bw_marker_index(k)))./(bw_max_freq(k)-f(bw_marker_index(k)));
    end
    spikiness(k) = max([spikiness1(k) spikiness2(k)]);
end


for k = 1:length(bw_max_freq)
    if bw_max_pres(k) == data(bw_marker_index(k+1))
        noisiness(k) = (bw_max_pres(k))./(data(bw_marker_index(k)));
    end
    if bw_max_pres(k) == data(bw_marker_index(k))
        noisiness(k) = (bw_max_pres(k))./(data(bw_marker_index(k+1)));
    end
    if bw_max_pres(k) ~= data(bw_marker_index(k)) && bw_max_pres(k) ~=
data(bw_marker_index(k+1))
```

```
        noisiness(k) =
(bw_max_pres(k))./(data(bw_marker_index(k+1))/2+data(bw_marker_index(k)
)/2);
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
area_threshold = multiplier4*max(area1);
amp_threshold = multiplier5*max(data);

bw_width_threshold = 100;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


g_area = sum(data)*dt;
g_std = std(data);
g_num_peaks1 = sum(num_peaks1);
g_num_peaks2 = sum(num_peaks2);
g_modal_dens1 = g_num_peaks1/(max(f)-min(f));
g_modal_dens2 = g_num_peaks2/(max(f)-min(f));
g_max_pres = max(data);
g_max_new = max(data)./max(data_true);
g_num_points1 = length(strat_freq);
g_num_points2 = length(strat_freq2);



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

k = 1;

f_value1 = [];
f_value2 = [];
f_value3 = [];
f_value4 = [];

f_value6 = [];
f_value7 = [];
f_value8 = [];


f_freq = [];
f_Q = [];
f_bw = [];
f_modal_dens1 = [];
f_modal_dens2 = [];
f_area = [];
f_spikiness = [];
f_noisiness = [];
f_num_peaks1 = [];
```

```
f_num_peaks2 = [];
f_max_pres = [];



for i = 1: length(bw_max_freq)
%     if bw_max_pres(i) > bw_threshold
%         if spikiness(i) > spikiness_threshold
            if area1(i) >= area_threshold || bw_max_pres(i) >=
amp_threshold
%                 if bw(i) > bw_width_threshold
                    f_freq(k) = bw_max_freq(i);
                    f_bw(k) = bw(i);
                    f_Q(k) = Q(i);
                    f_modal_dens1(k) = modal_dens1(i);
                    f_modal_dens2(k) = modal_dens2(i);
                    f_spikiness(k) = spikiness(i);
                    f_noisiness(k) = noisiness(i);
                    f_area(k) = area1(i);
                    f_num_peaks1(k) = num_peaks1(i);
                    f_num_peaks2(k) = num_peaks2(i);
                    f_max_pres(k) = bw_max_pres(i);

                    f_value1(k) = Q(i).*modal_dens1(i);
                    f_value2(k) = Q(i).*modal_dens2(i);
                    f_value3(k) = 10*log(Q(i).*modal_dens1(i));
                    f_value4(k) = 10*log(Q(i).*modal_dens2(i));

                    f_value5(k) = (area1(i)./g_area);
                    f_value6(k) = (bw_max_pres(i)./max(data));
                    f_value7(k) =
(area1(i)./g_area).*(bw_max_pres(i)./max(data));

                    k = k+1;
%                 end
            end
%         end
%     end
end

g_num_features = k-1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

id_bw = bw/sum(bw);
id_area = area1/g_area;
id_max = bw_max_pres/max(data);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

a_bw_Q = sum(Q.*id_bw);
a_bw_modal_dens1 = sum(modal_dens1.*id_bw);
a_bw_modal_dens2 = sum(modal_dens2.*id_bw);
a_bw_spikiness = sum(spikiness.*id_bw);
a_bw_noisiness = sum(noisiness.*id_bw);
```

```
a_bw_value1 = sum(Q.*modal_dens1.*id_bw);
a_bw_value2 = sum(Q.*modal_dens2.*id_bw);
a_bw_value3 = sum(10*log(Q.*modal_dens1).*id_bw);
a_bw_value4 = sum(10*log(Q.*modal_dens2).*id_bw);

a_area_Q = sum(Q.*id_area);
a_area_modal_dens1 = sum(modal_dens1.*id_area);
a_area_modal_dens2 = sum(modal_dens2.*id_area);
a_area_spikiness = sum(spikiness.*id_area);
a_area_noisiness = sum(noisiness.*id_area);

a_area_value1 = sum(Q.*modal_dens1.*id_area);
a_area_value2 = sum(Q.*modal_dens2.*id_area);
a_area_value3 = sum(10*log(Q.*modal_dens1).*id_area);
a_area_value4 = sum(10*log(Q.*modal_dens2).*id_area);

a_max_Q = sum(Q.*id_max);
a_max_modal_dens1 = sum(modal_dens1.*id_max);
a_max_modal_dens2 = sum(modal_dens2.*id_max);
a_max_spikiness = sum(spikiness.*id_max);
a_max_noisiness = sum(noisiness.*id_max);

a_max_value1 = sum(Q.*modal_dens1.*id_max);
a_max_value2 = sum(Q.*modal_dens2.*id_max);
a_max_value3 = sum(10*log(Q.*modal_dens1).*id_max);
a_max_value4 = sum(10*log(Q.*modal_dens2).*id_max);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

marker_data =
struct('mfreq',bw_marker_freq,'mpres',bw_marker_pres,'s1freq',strat_fre
q,'s1pres',strat_pres,'s2freq',strat_freq2,'s2pres',strat_pres2);

global_data =
struct('area',g_area,'std',g_std,'peaks1',g_num_peaks1,'peaks2',g_num_p
eaks2,'modaldens1',g_modal_dens1,'modaldens2',g_modal_dens2,'max',g_max
_pres,'loop',g_max_new,'num_points1',g_num_points1,'num_points2',g_num_
points2,'features',g_num_features);

feature_data =
struct('freq',f_freq,'Q',f_Q,'bw',f_bw,'modaldens1',f_modal_dens1,'moda
ldens2',f_modal_dens2,'area',f_area,'spike',f_spikiness,'noise',f_noisi
ness,'numpeaks1',f_num_peaks1,'numpeaks2',f_num_peaks2,'max',f_max_pres
,'value1',f_value1,'value2',f_value2,'value3',f_value3,'value4',f_value
4,'value5',f_value5,'value6',f_value6,'value7',f_value7);

average_bw_data =
struct('Q',a_bw_Q,'modaldens1',a_bw_modal_dens1,'modaldens2',a_bw_modal
_dens2,'spike',a_bw_spikiness,'noise',a_bw_noisiness,'value1',a_bw_valu
e1,'value2',a_bw_value2,'value3',a_bw_value3,'value4',a_bw_value4);
average_area_data =
struct('Q',a_area_Q,'modaldens1',a_area_modal_dens1,'modaldens2',a_area
_modal_dens2,'spike',a_area_spikiness,'noise',a_area_noisiness,'value1'
```

```
,a_area_value1,'value2',a_area_value2,'value3',a_area_value3,'value4',a
_area_value4);
average_max_data =
struct('Q',a_max_Q,'modaldens1',a_max_modal_dens1,'modaldens2',a_max_mo
dal_dens2,'spike',a_max_spikiness,'noise',a_max_noisiness,'value1',a_ma
x_value1,'value2',a_max_value2,'value3',a_max_value3,'value4',a_max_val
ue4);

average_data =
struct('bw',average_bw_data,'area',average_area_data,'max',average_max_
data);

target_identifier =
struct('marker',marker_data,'feature',feature_data,'global',global_data
,'average',average_data);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
smoother = 3;
nuclear = 250;

f_true = f;
data_true = data;

% 'working signal'

if length(strat_freq2) > nuclear
    data = binavg(data,smoother);
    f = f(1:smoother:length(data)*smoother);
    'nuke'
%     close all
    [target_identifier] =
id_compute_loop_new_JD(f,data,f_true,data_true,multiplier1,multiplier2,
multiplier3,multiplier4,multiplier5,global_data);
end
```

```
function [predicted,Pd,False_Alarm] =
svm_avg_butter(target1_id,target2_id,trashbag_id,foamcup_id,sodacan_id,
crushedcan_id,pepsibottle_id,cylinder_id,order,slack,func)

[predicted_temp,alpha,bias,variables,Y,Pd_temp,False_Alarm_temp] =
svm_tally_order_butter(target1_id,target2_id,trashbag_id,foamcup_id,sod
acan_id,crushedcan_id,pepsibottle_id,cylinder_id,order,slack,func);
predicted = predicted_temp;
Pd = Pd_temp;
False_Alarm = False_Alarm_temp;
```

```
function [predicted,alpha,bias,variables,Y,Pd,False_Alarm] =
svm_tally_order_butter(target1_id,target2_id,trashbag_id,foamcup_id,sod
acan_id,crushedcan_id,pepsibottle_id,cylinder_id,order,slack,func)

global p1
p1 = order;

[variables,trial,result] =
svm_id_process_butter_JD(target1_id,target2_id,trashbag_id,cylinder_id,
sodacan_id,pepsibottle_id,foamcup_id,crushedcan_id);

Y=result;

for i = 1:length(result)
    if i == 1
        [nsv alpha bias] =
svc(variables(i+1:length(result),:),Y,func,slack,0);
        predicted =
svcoutput(variables(i+1:length(result),:),Y,variables(i,:),func,alpha,b
ias,0);
    end
    if i == length(result)
        [nsv alpha bias] = svc(variables(1:length(result)-
1,:),Y,func,slack,0);
        predicted = svcoutput(variables(1:length(result)-
1,:),Y,variables(i,:),func,alpha,bias,0);
    end
    if i ~=1 && i ~= length(result)
        [nsv alpha bias] = svc(variables([1:i-1
i+1:length(result)],:),Y,func,slack,0);
        predicted = svcoutput(variables([1:i-1
i+1:length(result)],:),Y,variables(i,:),func,alpha,bias,0);
    end
    charlie(:,i) = alpha;
    delta(i) = bias;
    echo(i) = predicted;
end

predicted = echo;
alpha = charlie;
bias = delta;

targets = 0;
clutter = 0;

t_right = 0;
t_wrong = 0;
c_right = 0;
c_wrong = 0;


for i = 1:length(result)
    if result(i) == 1
        targets = targets+1;
        if predicted(i) == result(i)
```

```
            t_right = t_right+1;
        else
            t_wrong = t_wrong+1;
        end
    end
    if result(i) == -1
        clutter = clutter+1;
        if predicted(i) == result(i)
            c_right = c_right+1;
        else
            c_wrong = c_wrong+1;
        end
    end
end


predicted = predicted';

[result predicted trial];

Pd = t_right/targets;

False_Alarm = c_wrong/clutter;
```

```
function [variables,trial,result] =
svm_id_process_butter_JD(target1_id,target2_id,trashbag_id,cylinder_id,
sodacan_id,pepsibottle_id,foamcup_id,crushedcan_id)

%
[train_variables,train_trial,train_result,test_variables,test_trial,tes
t_result] =
svm_id_process_2008_JD(target1_id20,target2_id20,trashbag_id20,cylinder
_id20,sodacan_id20,pepsibottle_id20,foamcup_id20,crushedcan_id20)

% [test_variables,test_trial,test_result] =
svm_id_process_2008_JD(target1_id14,target2_id14,trashbag_id14,cylinder
_id14,sodacan_id14,pepsibottle_id14,foamcup_id14,crushedcan_id14)

% [test_variables,test_trial,test_result] =
svm_data_2008_JD(target1_id,target2_id,trashbag_id,plate_id,sodacan_id,
pepsibottle_id,foamcup_id)


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


x_variables = [];
x_trial = [];
x_result = [];

x_id = target1_id;

k = 1;
for i = 1:length(x_id(:))
%     for j = 1:length(x_id{i}.feature.freq)
        x_variables(k,:) = svm_variables_helper(x_id{i});

        x_trial(k) = i;
        x_result(k) = 1;
        k = k+1;
%     end
end


target1_variables = x_variables;
target1_trial = x_trial;
target1_result = x_result;


x_variables = [];
x_trial = [];
x_result = [];

x_id = target2_id;

k = 1;
```

```
for i = 1:length(x_id(:))
%     for j = 1:length(x_id{i}.feature.freq)
        x_variables(k,:) = svm_variables_helper(x_id{i});

        x_trial(k) = i;
        x_result(k) = 1;
        k = k+1;
%     end
end


target2_variables = x_variables;
target2_trial = x_trial;
target2_result = x_result;


x_variables = [];
x_trial = [];
x_result = [];



x_id = cylinder_id;

k = 1;
for i = 1:length(x_id(:))
%     for j = 1:length(x_id{i}.feature.freq)
        x_variables(k,:) = svm_variables_helper(x_id{i});

        x_trial(k) = i;
        x_result(k) = -1;
        k = k+1;
%     end
end


cylinder_variables = x_variables;
cylinder_trial = x_trial;
cylinder_result = x_result;


x_variables = [];
x_trial = [];
x_result = [];



x_id = sodacan_id;

k = 1;
for i = 1:length(x_id(:))
%     for j = 1:length(x_id{i}.feature.freq)
        x_variables(k,:) = svm_variables_helper(x_id{i});
```

```
        x_trial(k) = i;
        x_result(k) = -1;
        k = k+1;
%     end
end


sodacan_variables = x_variables;
sodacan_trial = x_trial;
sodacan_result = x_result;


x_variables = [];
x_trial = [];
x_result = [];



x_id = crushedcan_id;

k = 1;
for i = 1:length(x_id(:))
%     for j = 1:length(x_id{i}.feature.freq)
        x_variables(k,:) = svm_variables_helper(x_id{i});

        x_trial(k) = i;
        x_result(k) = -1;
        k = k+1;
%     end
end


crushedcan_variables = x_variables;
crushedcan_trial = x_trial;
crushedcan_result = x_result;


x_variables = [];
x_trial = [];
x_result = [];



x_id = foamcup_id;

k = 1;
for i = 1:length(x_id(:))
%     for j = 1:length(x_id{i}.feature.freq)
        x_variables(k,:) = svm_variables_helper(x_id{i});

        x_trial(k) = i;
        x_result(k) = -1;
```

```
        k = k+1;
%     end
end


foamcup_variables = x_variables;
foamcup_trial = x_trial;
foamcup_result = x_result;


x_variables = [];
x_trial = [];
x_result = [];



x_id = trashbag_id;

k = 1;
for i = 1:length(x_id(:))
%     for j = 1:length(x_id{i}.feature.freq)
        x_variables(k,:) = svm_variables_helper(x_id{i});

        x_trial(k) = i;
        x_result(k) = -1;
        k = k+1;
%     end
end


trashbag_variables = x_variables;
trashbag_trial = x_trial;
trashbag_result = x_result;


x_variables = [];
x_trial = [];
x_result = [];



x_id = pepsibottle_id;

k = 1;
for i = 1:length(x_id(:))
%     for j = 1:length(x_id{i}.feature.freq)
        x_variables(k,:) = svm_variables_helper(x_id{i});

        x_trial(k) = i;
        x_result(k) = -1;
        k = k+1;
%     end
end
```

```
pepsibottle_variables = x_variables;
pepsibottle_trial = x_trial;
pepsibottle_result = x_result;


x_variables = [];
x_trial = [];
x_result = [];



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%5



n=1;
m=1;

x_variables = target1_variables;
x_trial = target1_trial;
x_result = target1_result;


for i=1:length(x_trial)
    variables(n,:)=x_variables(i,:);
    trial(n,1)=x_trial(i);
    result(n,1)=x_result(i);
    n=n+1;
end


x_variables = target2_variables;
x_trial = target2_trial;
x_result = target2_result;


for i=1:length(x_trial)
    variables(n,:)=x_variables(i,:);
    trial(n,1)=x_trial(i);
    result(n,1)=x_result(i);
    n=n+1;
end


x_variables = pepsibottle_variables;
x_trial = pepsibottle_trial;
x_result = pepsibottle_result;


for i=1:length(x_trial)
```

```
    variables(n,:)=x_variables(i,:);
    trial(n,1)=x_trial(i);
    result(n,1)=x_result(i);
    n=n+1;
end


x_variables = trashbag_variables;
x_trial = trashbag_trial;
x_result = trashbag_result;


for i=1:length(x_trial)
    variables(n,:)=x_variables(i,:);
    trial(n,1)=x_trial(i);
    result(n,1)=x_result(i);
    n=n+1;
end


x_variables = foamcup_variables;
x_trial = foamcup_trial;
x_result = foamcup_result;


for i=1:length(x_trial)
    variables(n,:)=x_variables(i,:);
    trial(n,1)=x_trial(i);
    result(n,1)=x_result(i);
    n=n+1;
end


x_variables = crushedcan_variables;
x_trial = crushedcan_trial;
x_result = crushedcan_result;


for i=1:length(x_trial)
    variables(n,:)=x_variables(i,:);
    trial(n,1)=x_trial(i);
    result(n,1)=x_result(i);
    n=n+1;
end


x_variables = sodacan_variables;
x_trial = sodacan_trial;
x_result = sodacan_result;


for i=1:length(x_trial)
    variables(n,:)=x_variables(i,:);
    trial(n,1)=x_trial(i);
```

```
    result(n,1)=x_result(i);
    n=n+1;
end


x_variables = cylinder_variables;
x_trial = cylinder_trial;
x_result = cylinder_result;


for i=1:length(x_trial)
    variables(n,:)=x_variables(i,:);
    trial(n,1)=x_trial(i);
    result(n,1)=x_result(i);
    n=n+1;
end
```

```
function [x_variables] = svm_variables_helper(x_id)

for i = 1:length(x_id.feature.freq)
    if x_id.feature.area(i) == max(x_id.feature.area)

%
'freq','Q','bw','modaldens1','modaldens2','area','spike','noise','numpe
aks1','numpeaks2','max','value1','value2','value3','value4','value5','v
alue6','value7'

        FA1 = x_id.feature.area(i);
        FA2 = x_id.feature.modaldens1(i);
        FA3 = x_id.feature.modaldens2(i);
        FA4 = x_id.feature.spike(i);
        FA5 = x_id.feature.noise(i);
        FA6 = x_id.feature.numpeaks1(i);
        FA7 = x_id.feature.numpeaks2(i);
        FA8 = x_id.feature.max(i);

        FAV1 = x_id.feature.value1(i);
        FAV2 = x_id.feature.value2(i);
        FAV3 = x_id.feature.value3(i);
        FAV4 = x_id.feature.value4(i);
        FAV5 = x_id.feature.value5(i);
        FAV6 = x_id.feature.value6(i);
        FAV7 = x_id.feature.value7(i);

    end

    if x_id.feature.max(i) == max(x_id.feature.max)

%
'freq','Q','bw','modaldens1','modaldens2','area','spike','noise','numpe
aks1','numpeaks2','max','value1','value2','value3','value4','value5','v
alue6','value7'

        FM1 = x_id.feature.area(i);
        FM2 = x_id.feature.modaldens1(i);
        FM3 = x_id.feature.modaldens2(i);
        FM4 = x_id.feature.spike(i);
        FM5 = x_id.feature.noise(i);
        FM6 = x_id.feature.numpeaks1(i);
        FM7 = x_id.feature.numpeaks2(i);
        FM8 = x_id.feature.max(i);

        FMV1 = x_id.feature.value1(i);
        FMV2 = x_id.feature.value2(i);
        FMV3 = x_id.feature.value3(i);
        FMV4 = x_id.feature.value4(i);
        FMV5 = x_id.feature.value5(i);
        FMV6 = x_id.feature.value6(i);
        FMV7 = x_id.feature.value7(i);

    end
end
```

```
%
'area','std','peaks1','peaks2','modaldens1','modaldens2','max_pres','ma
x_new','num_points1','num_points2','features');

        G1 = x_id.global.area;
        G2 = x_id.global.std;
        G3 = x_id.global.peaks1;
        G4 = x_id.global.peaks2;
        G5 = x_id.global.modaldens1;
        G6 = x_id.global.modaldens2;
        G7 = x_id.global.max;
        G8 = x_id.global.loop;
        G9 = x_id.global.num_points1;
        G10 = x_id.global.num_points2;
        G11 = x_id.global.features;

%
'Q','modaldens1','modaldens2','spike','noise','value1','value2','value3
','value4');

        BW1 = x_id.average.bw.Q;
        BW2 = x_id.average.bw.modaldens1;
        BW3 = x_id.average.bw.modaldens2;
        BW4 = x_id.average.bw.spike;
        BW5 = x_id.average.bw.noise;

        BWV1 = x_id.average.bw.value1;
        BWV2 = x_id.average.bw.value2;
        BWV3 = x_id.average.bw.value3;
        BWV4 = x_id.average.bw.value4;

        A1 = x_id.average.area.Q;
        A2 = x_id.average.area.modaldens1;
        A3 = x_id.average.area.modaldens2;
        A4 = x_id.average.area.spike;
        A5 = x_id.average.area.noise;

        AV1 = x_id.average.area.value1;
        AV2 = x_id.average.area.value2;
        AV3 = x_id.average.area.value3;
        AV4 = x_id.average.area.value4;

        M1 = x_id.average.max.Q;
        M2 = x_id.average.max.modaldens1;
        M3 = x_id.average.max.modaldens2;
        M4 = x_id.average.max.spike;
        M5 = x_id.average.max.noise;

        MV1 = x_id.average.max.value1;
        MV2 = x_id.average.max.value2;
        MV3 = x_id.average.max.value3;
        MV4 = x_id.average.max.value4;
```

```
% x_variables = [FA5 FA6 FA7 FAV3 FAV4 FAV5 FM5 FM6 FM7 FMV1 FMV2 G1 G3
G4 G7 G10 G11 BW1 BWV4 M1];

x_variables = [FA4 FM4 G1 G11 G7 BW4 A4 AV4 MV4];
% feature area value 4 feature max value 4 global area # features, BW
avg 4
% area spike avg area 4 mass avg 4

% x_variables = [FAV1 FMV1 BWV1 AV1 MV1 FAV2 FMV2 BWV2 AV2 MV2 G1 G2 G3
G4 G10 G11];

% x_variables = [FAV1 FMV1 BWV1 FAV2 FMV2 BWV2 G1 G2 G7];
```

```
function [ts_new,start_marker,end_marker,problem] =
tau_calc_2008_JD(ts,fl)

fs = 50000;

marker = floor(.95*length(ts));

if marker > fs*120
    marker = fs*120;
end

temp = 0;
temp2 = 0;
temp3 = 0;
i = marker;

problem = 0;

while temp == 0
    k = 1;
    l = 1;

    if i > length(ts)-2
        temp = 1;
        problem = 1;
        'Problem!'
        marker2 = i;
    end

    if ts(i) > ts(i+1) && ts(i) > ts(i-1)
        while temp2 == 0
            if ts(i+k) >= ts(i+k+1) && ts(i+k) >= ts(i+k-1)
                temp2 = 1;
                distance = k;
            else
                k = k+1;
            end
        end
    end

    step = floor(fs/fl)-1;

    if k > step
        temp = 1;
        marker2 = i;
    end


    temp2 = 0;
    temp3 = 0;
    i = i+1;
end
```

```
end_marker = marker2;

backstep = floor(.99*6000000);
start_marker = end_marker-backstep+1;

if start_marker < 1
    start_marker = 1;
    problem = 1;
end

ts_new = ts(start_marker:end_marker);
```

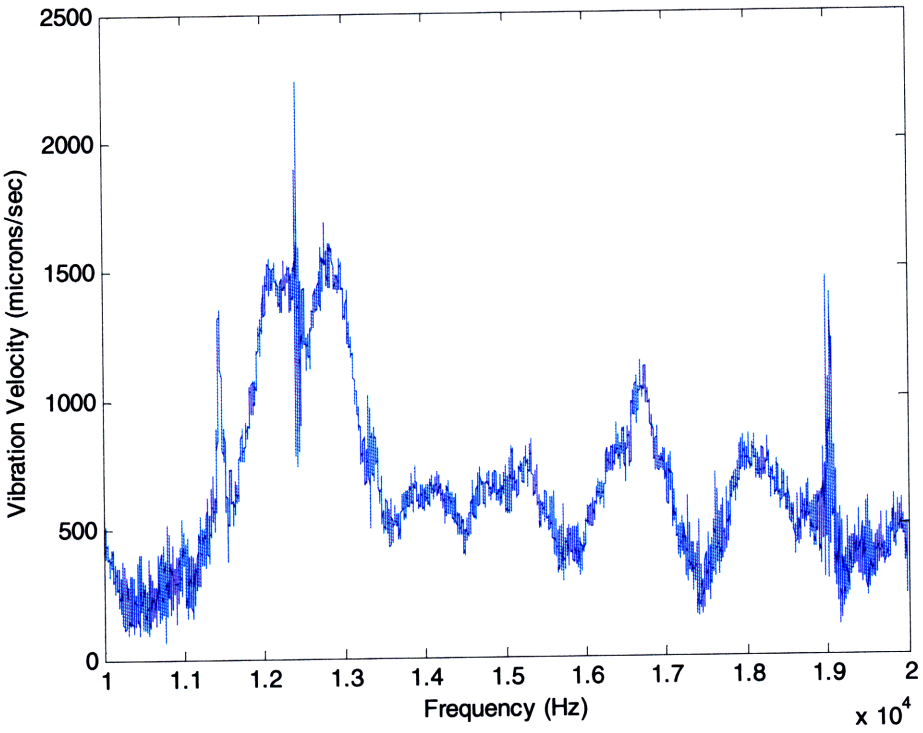
```
function [f,p] = accel_FFT(ts, dt, bf, accel, cf, ph, nf);

%[f,p] = accel_FFT(ts, dt, bf, accel, cf, ph, nf);
%[f,p] = accel_FFT(ts, 1/22e3, [1 11000], 'v', 104.4, 1);

%OUTPUTS
%f = desired frequency range
%p = fft output (velocity amplitude or acceleration)
%INPUTS
%ts = time series
%dt = data sampling time step
%bf = desired output frequency band
%accel = t,  integrate acceleration to velocity in time domain
%accel = v,  take fft of acceleration, then divide by 2pif in freq
domain
%accel = a,  fft pf acceleration
%accel = o,  compute instantaneous velocity in time domain
%ph = phase flag, 1 = p out in complex format, anything else gives
absolute
%value


%cf = cf*1e-3; %multiply by calibration factor (which needs to be
converted from mV to V/m/s^2)
ts = detrend(ts,'linear');
ts = (ts-mean(ts))/cf;       %acceleration
if accel == 't'
    ts = dt*cumsum(ts);      %integrate acceleration to get velocity --
>microns/sec

    %[b,a]=butter(3,.05, 'high');
    %ts = filtfilt(b,a,ts);
    %plot(ts)
    %pause
end
if accel == 'o'
    ts = dt*ts;
end

nSamp = length(ts);

if nf == 0
    N = fftLength(nSamp);
    ts = ts(nSamp-N:nSamp);
    nfft = N;  %over sample
else
    N = nSamp;
    nfft = nf;
end

Tmax = N*dt;

% win = boxcar(nfft);
% [s,f]    = pwelch(ts,win,nfft/2,nfft,1/dt);
% loglog(f,abs(s))
```

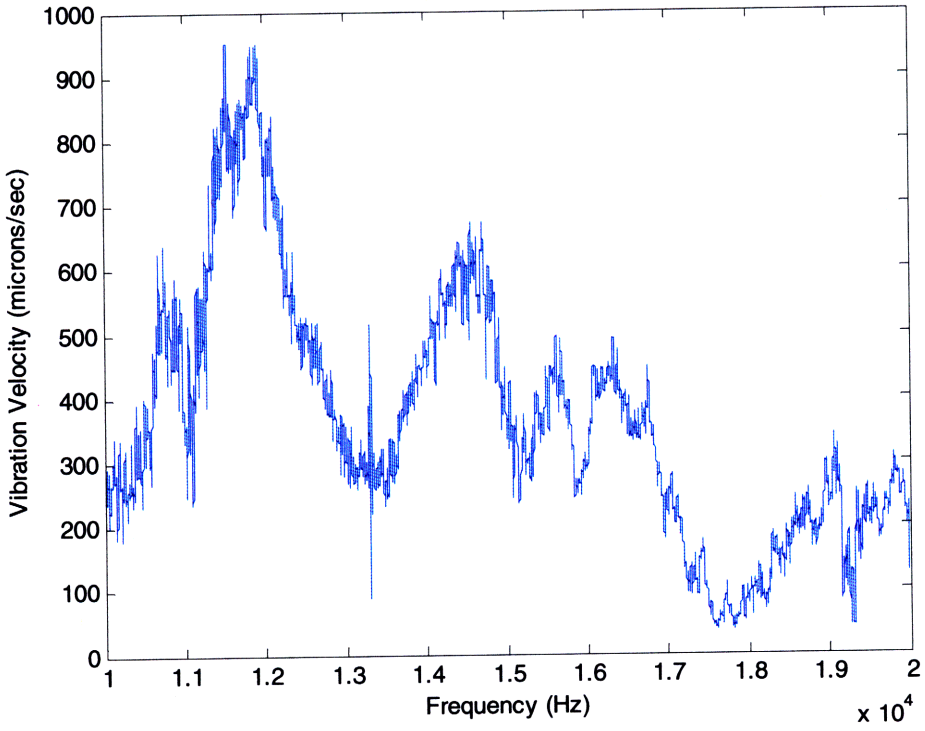
```
% pause

if length(bf) > 2
   window = hanning(length(ts));
   ts = ts.*window/.5;
end

stemp = fftshift(fft(ts,nfft))*dt/Tmax;
f = [0:-1+length(stemp)/2] / (dt*nfft);
s = 2*stemp(1+length(stemp)/2 : length(stemp));

if ph == 1
    p = s;
else
    p = abs(s);
end

if accel == 'v'
    p = p./(2*pi*f');
end

if bf(2) == 0
    return
end

if bf(1) == 0
    fl = 1;
else
    fl = findindex(f,bf(1));
end

if bf(2) > f(length(f))
    fh = length(f);
else
    fh = findindex(f,bf(2));
end

f=f(fl:fh);
p=p(fl:fh);
%-------------------------------------------------------------------------
----
function findex = findindex(f,f0);

%findex = findindex(f,f0);

fl = max(find(f<=f0));
fh = min(find(f>=f0));
d1 = abs(f(fl)-f0);
d2 = abs(f(fh)-f0);

if(d1 < d2),
    findex = fl;
else
    findex = fh;
end
```

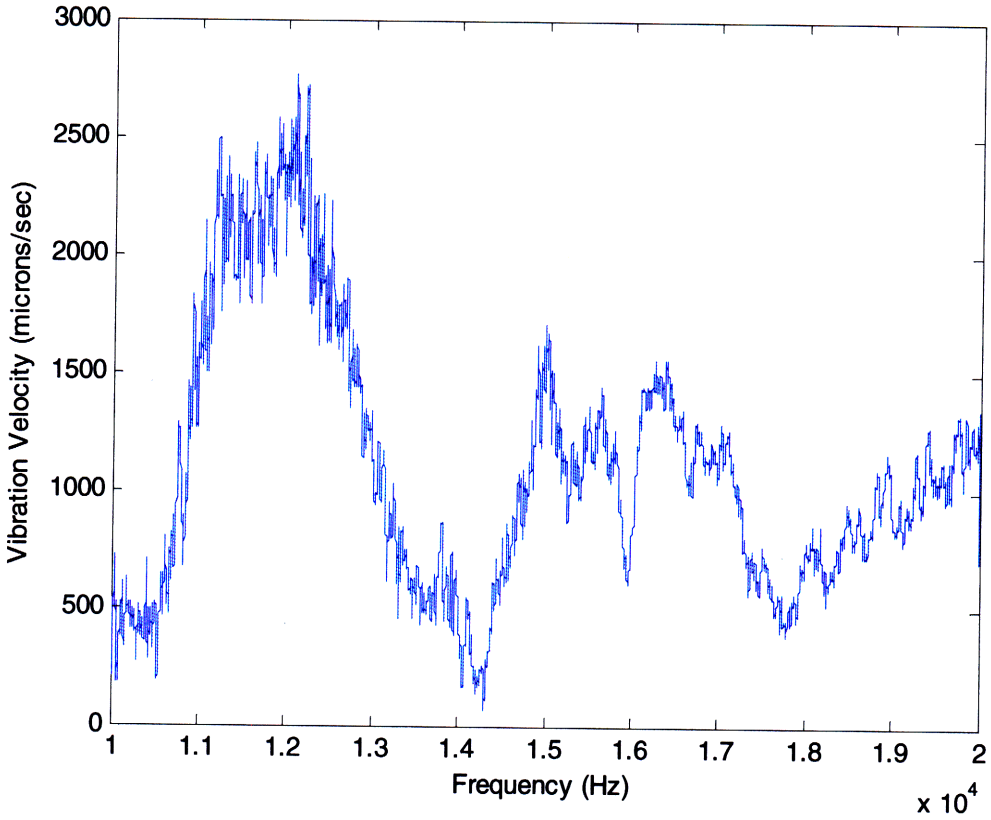
```
%-------------------------------------------------------------------------
----
function N = fftLength(nm);

ntemp = 0;
k = 0;

while ntemp < nm
    ntemp = 2^k;
    k=k+1;
end

N = 2^(k-2);

if N > 2^22
    N = 2^22;
end
```

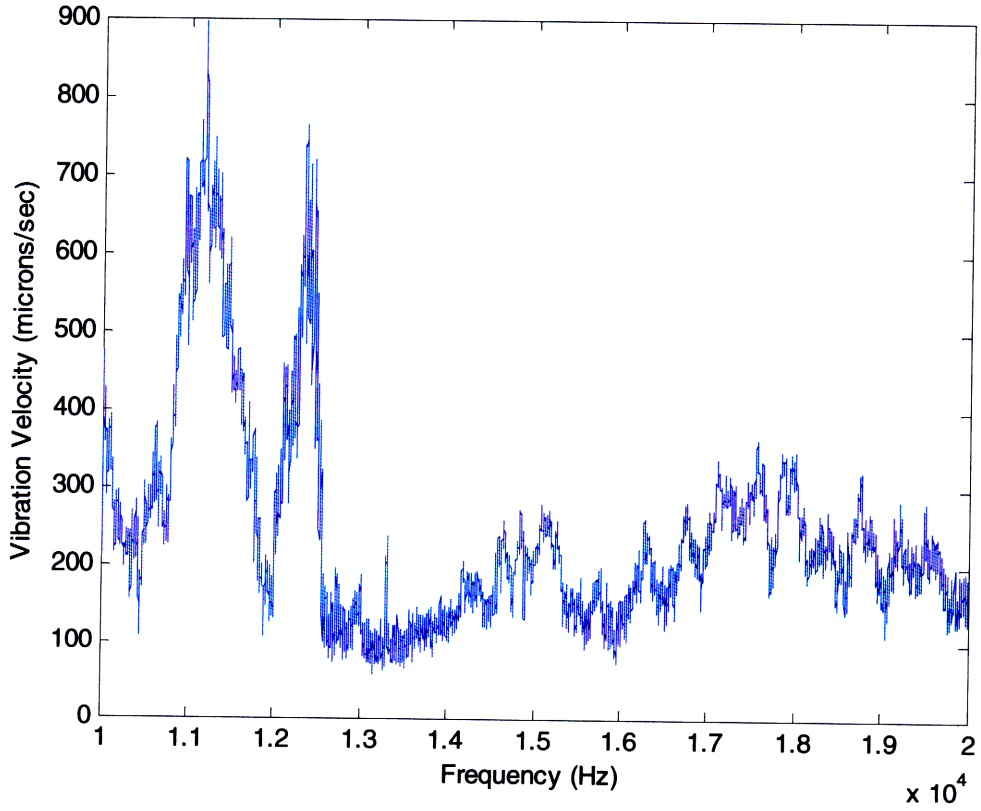
```
function ch = get_channel(file_in, chan, gain, sample_rate, nchannels);

%ch = get_channel(file_in, chan, gain, sample_rate, nchannels);

%function to grab a channel of data from a .bin file

%read entire vector unless otherwise hardcoded
file = dir(file_in);
bytes_per_samp = 2;
tot_bytes = file.bytes;
time_end = tot_bytes/bytes_per_samp/(sample_rate*nchannels);
time_start = 0;

for i = 1:length(chan)
   [data, err_out] = read_iotech(file_in, chan(i), time_start,
time_end, nchannels, sample_rate);
   ch(:,i) = data/2^16*gain;  %V
   clear data
end


%[psd, freq] = periodogram(data,
blackman(prod(size(data))),prod(size(data)),sample_rate);

%plot(freq, 10*log10(psd));

%specgram(data, sample_rate, sample_rate)
```

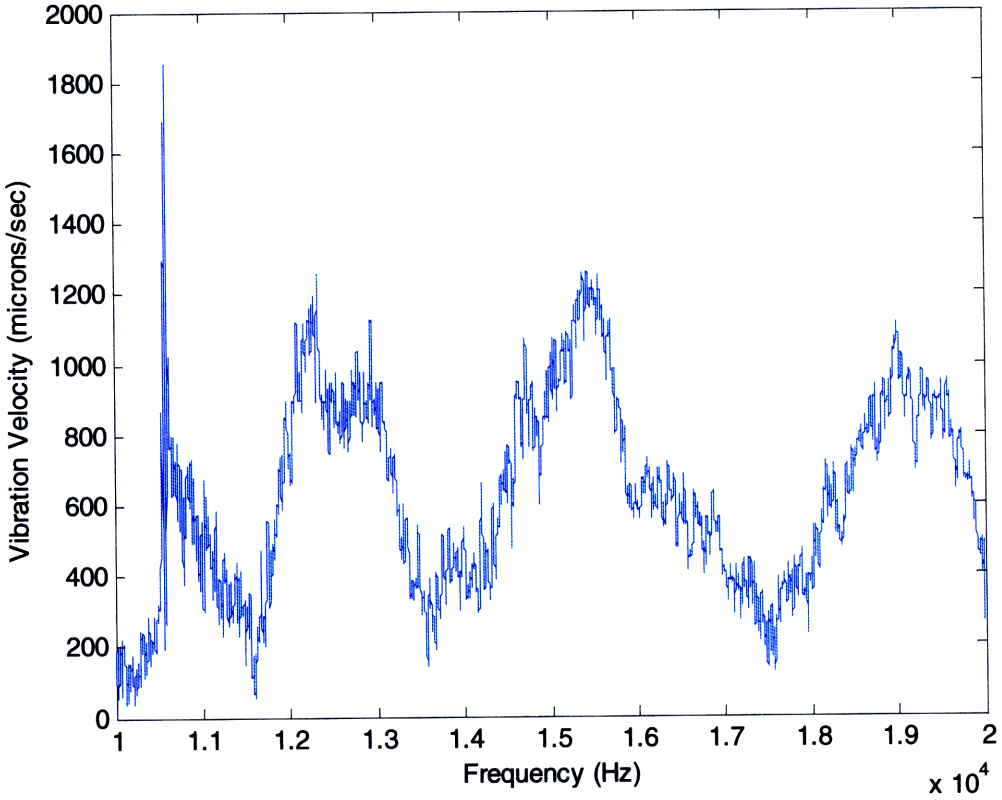
Appendix B – Experimental Target Signatures
Target 1

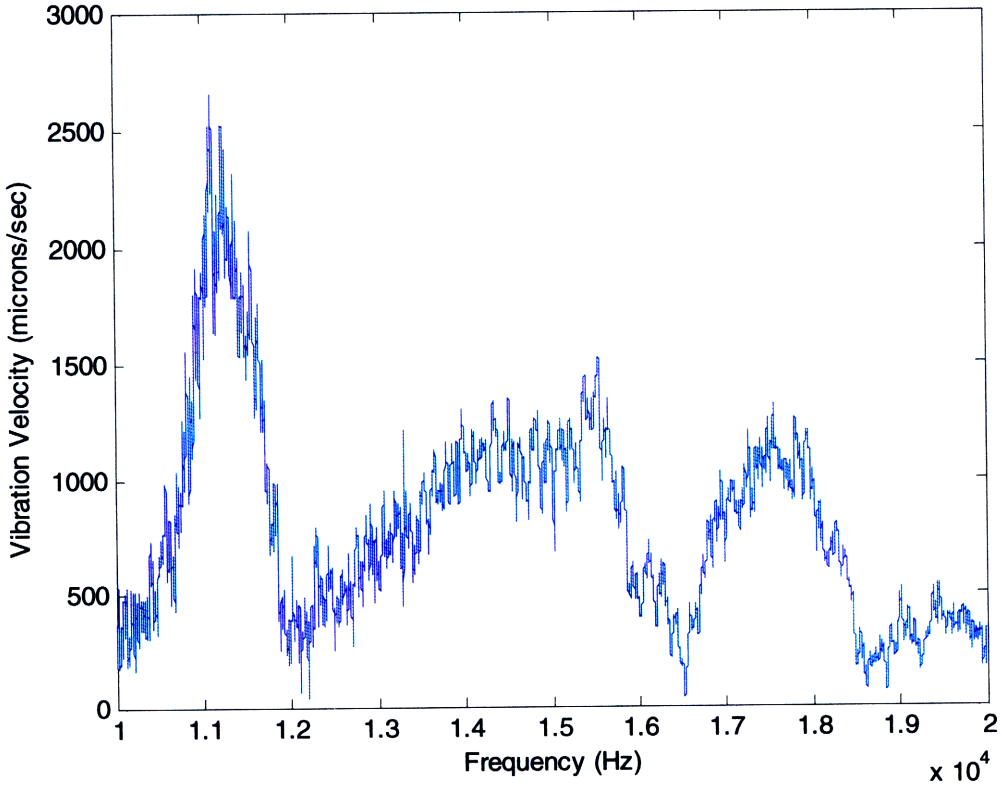


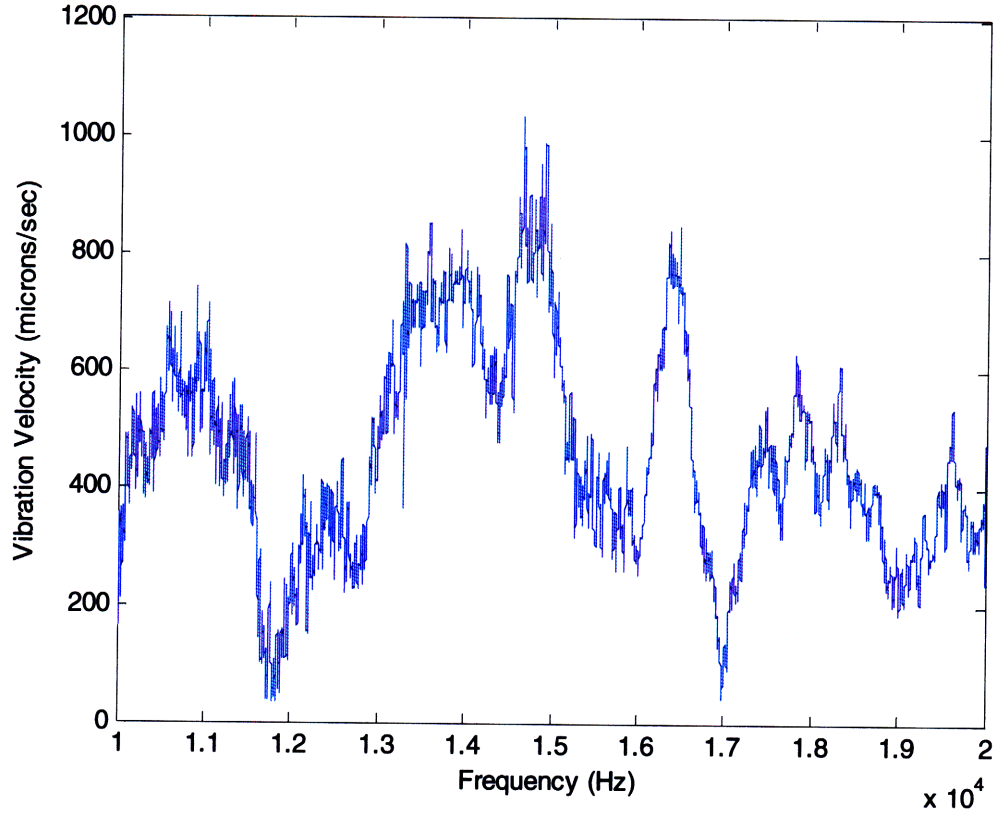


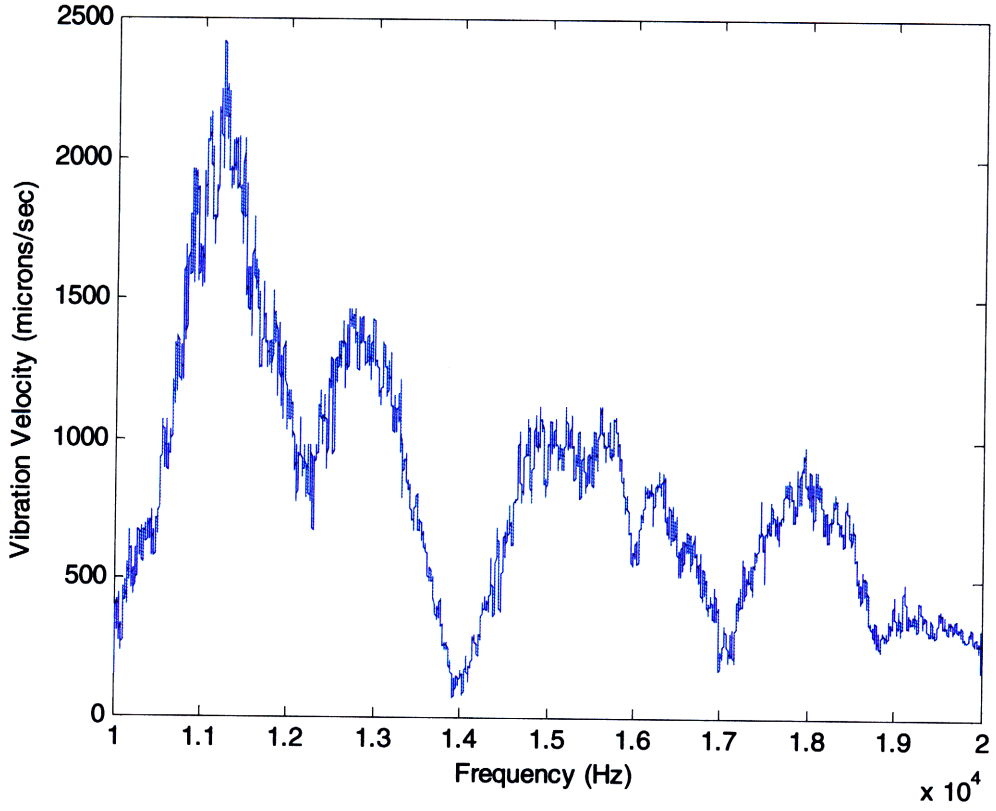

3000
2500
2000
1500
1000
500
0
Vibration Velocity (microns/sec)
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

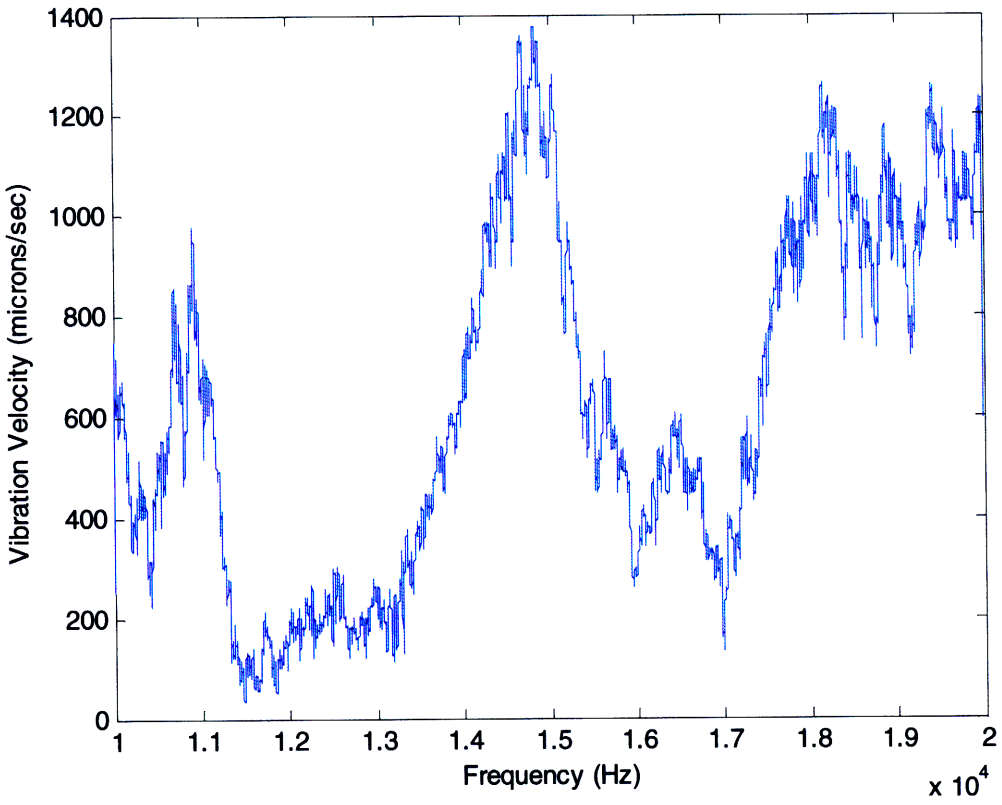

900
800
700
600
500
400
300
200
100
0
Vibration Velocity (microns/sec)
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

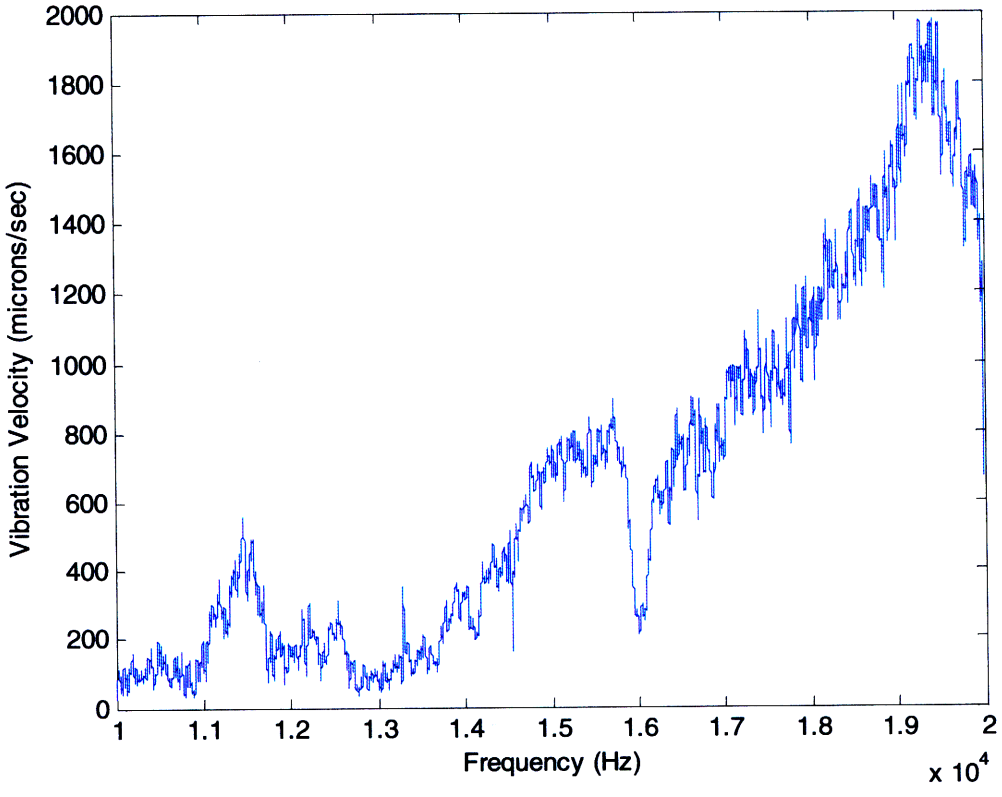

Vibration Velocity (microns/sec)
2000
1800
1600
1400
1200
1000
800
600
400
200
0
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

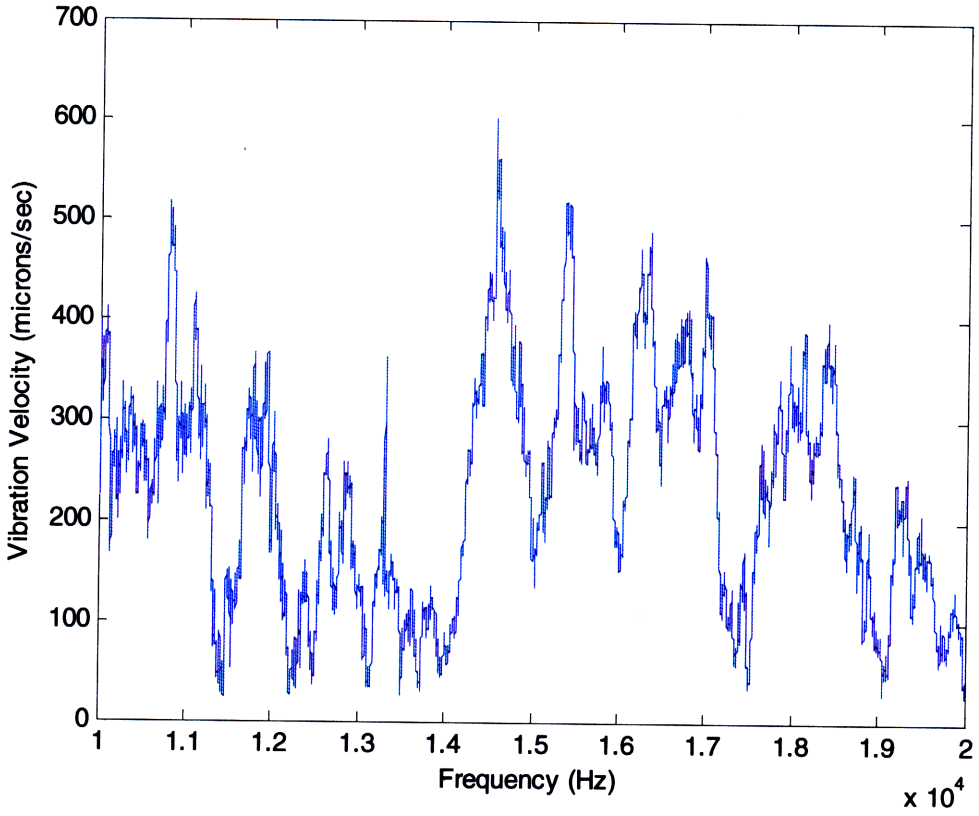

Vibration Velocity (microns/sec)
3000
2500
2000
1500
1000
500
0
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

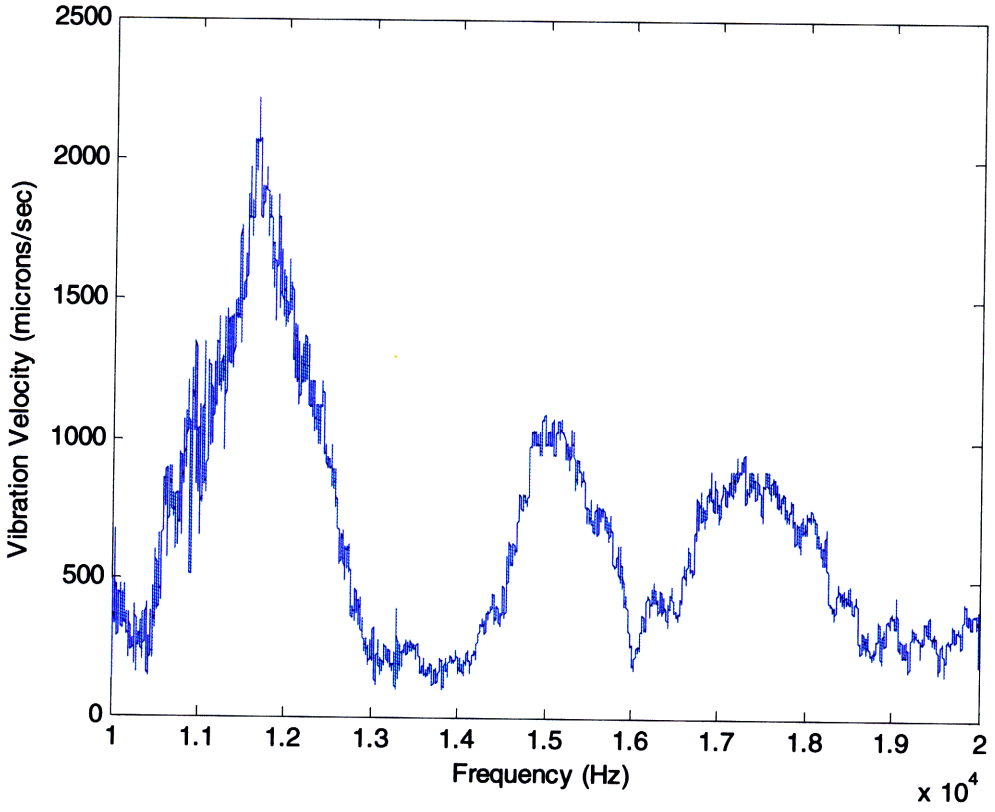

Vibration Velocity (microns/sec)
0
200
400
600
800
1000
1200
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

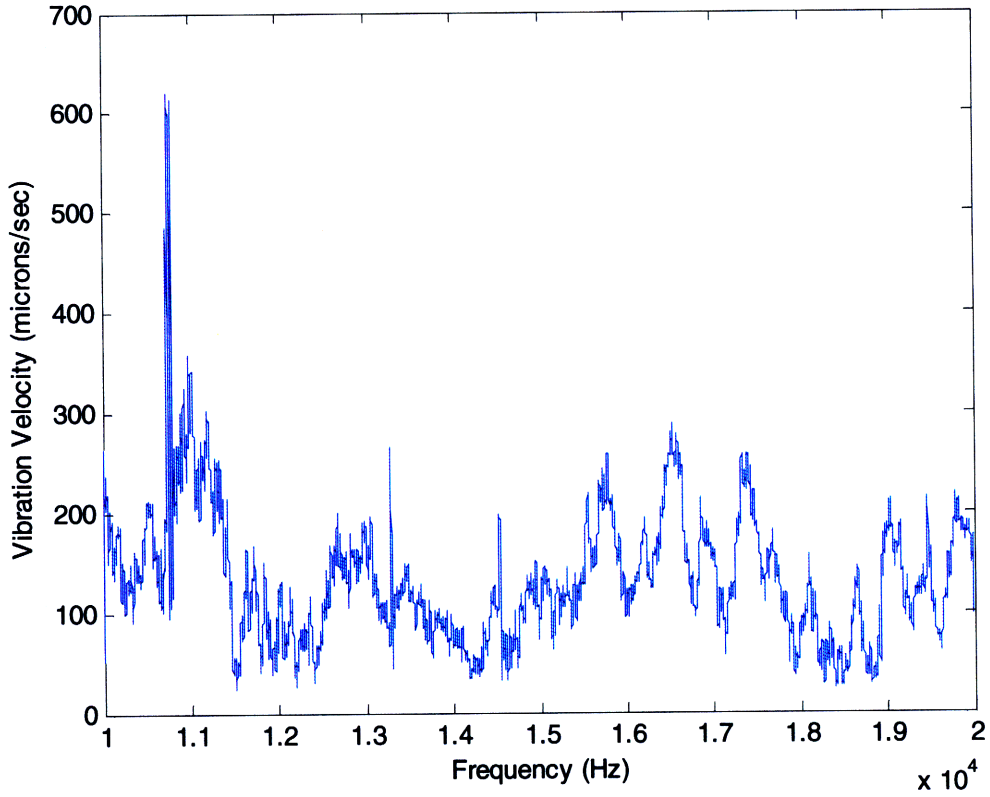

Vibration Velocity (microns/sec)
0
500
1000
1500
2000
2500
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

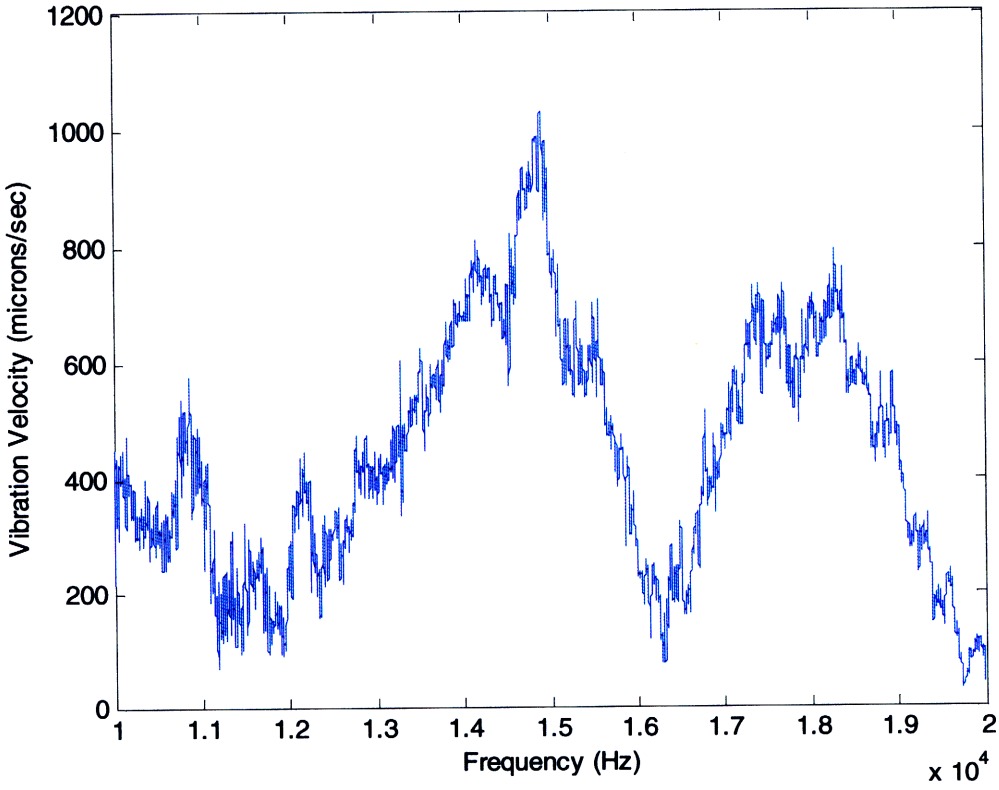

Vibration Velocity (microns/sec)
0
200
400
600
800
1000
1200
1400
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

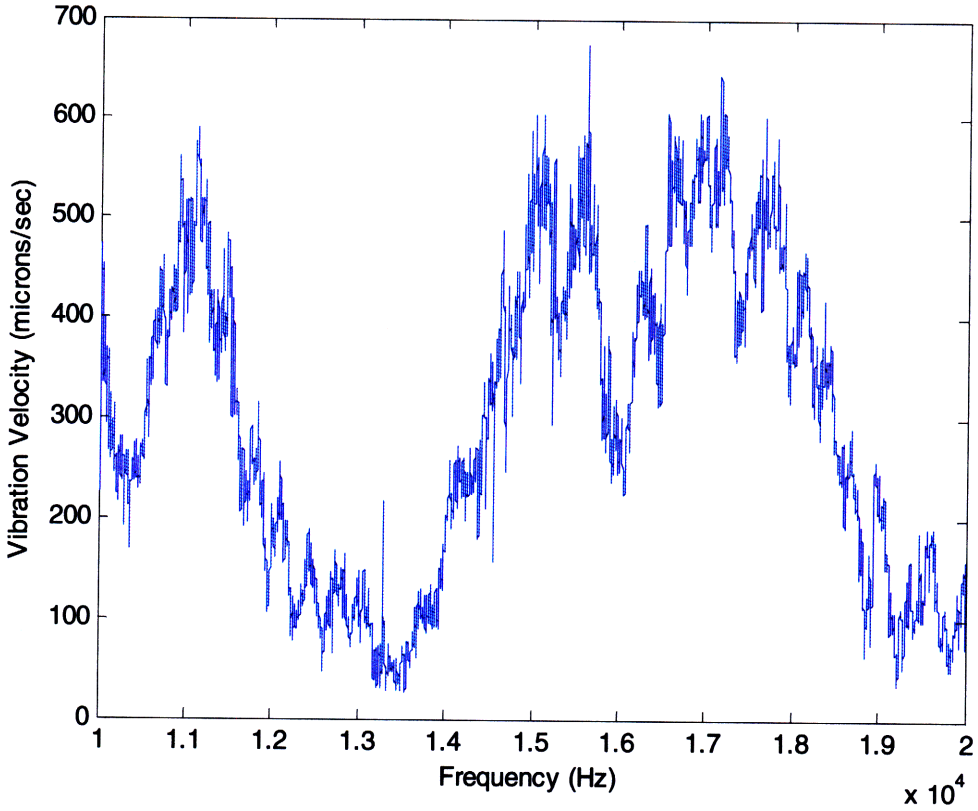

Vibration Velocity (microns/sec)
0
200
400
600
800
1000
1200
1400
1600
1800
2000
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4


Vibration Velocity (microns/sec)
700
600
500
400
300
200
100
0
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

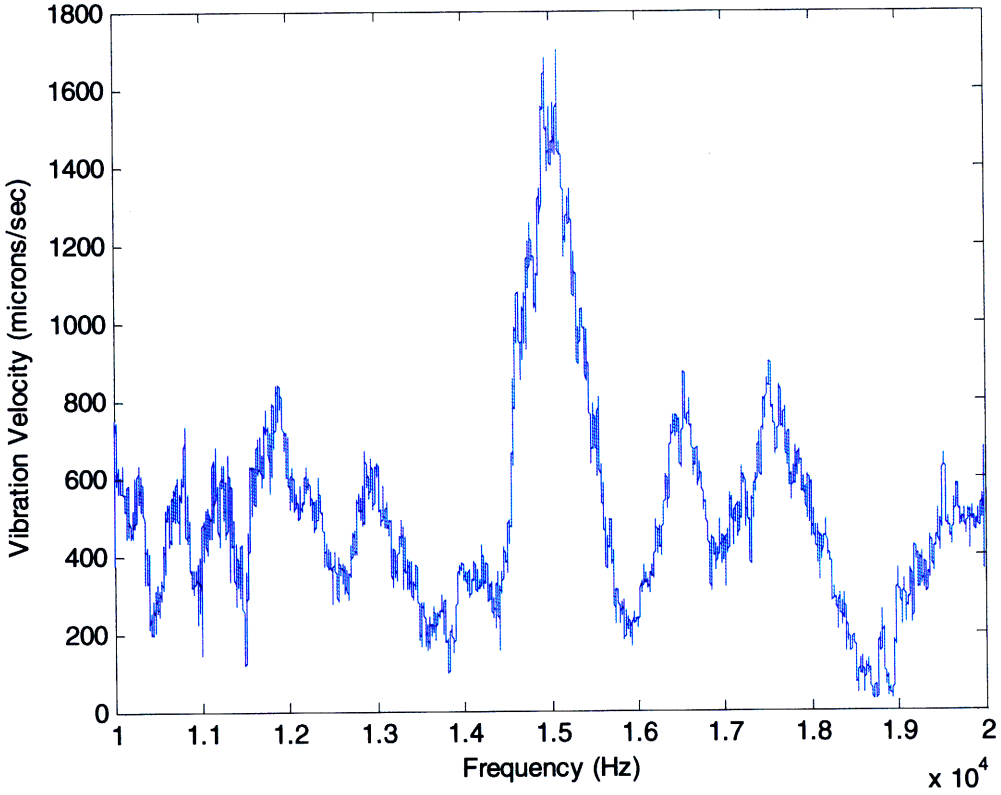

Vibration Velocity (microns/sec)
2500
2000
1500
1000
500
0
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

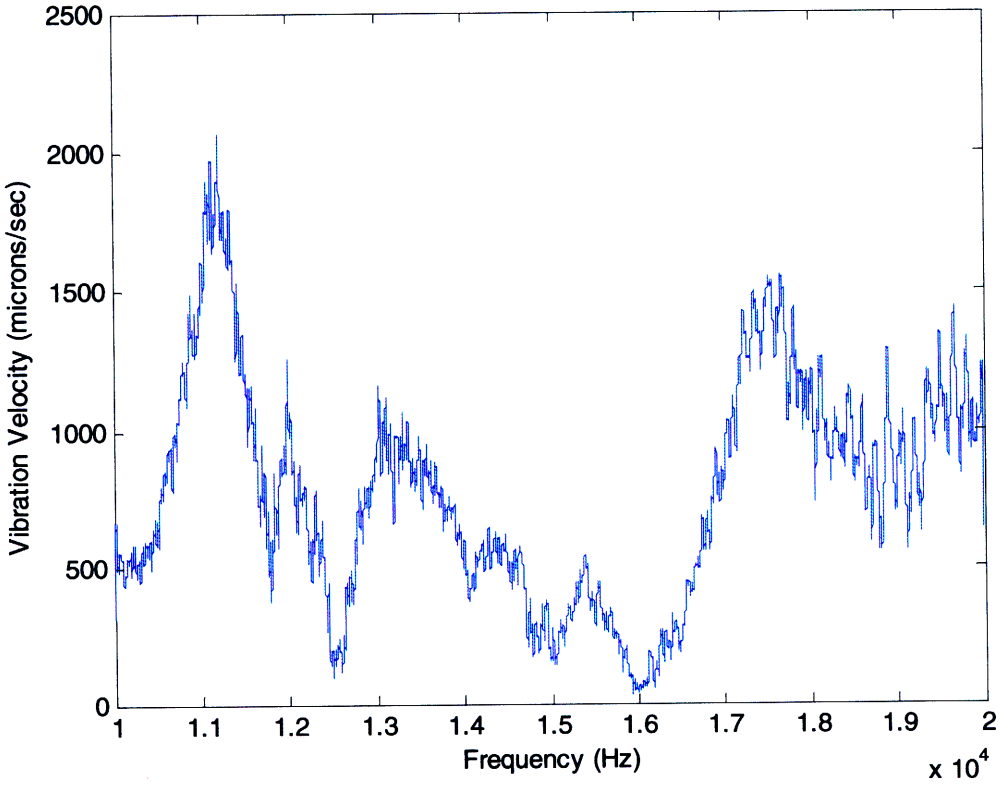

Vibration Velocity (microns/sec)
700
600
500
400
300
200
100
0
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

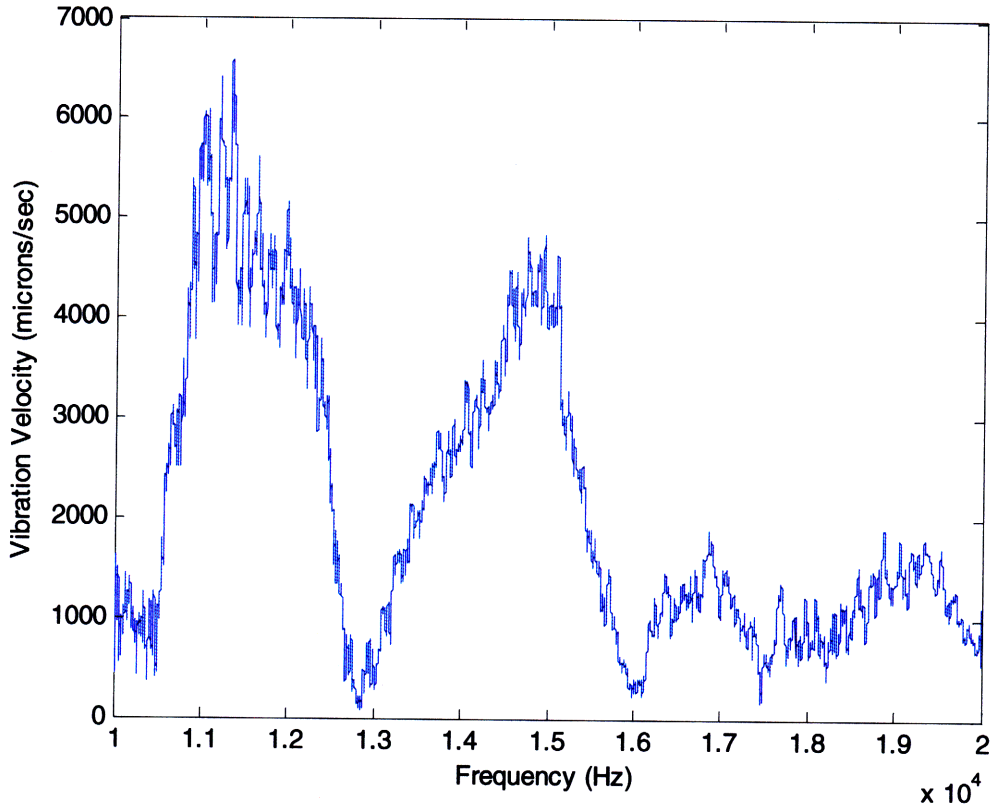

Vibration Velocity (microns/sec)
1200
1000
800
600
400
200
0
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

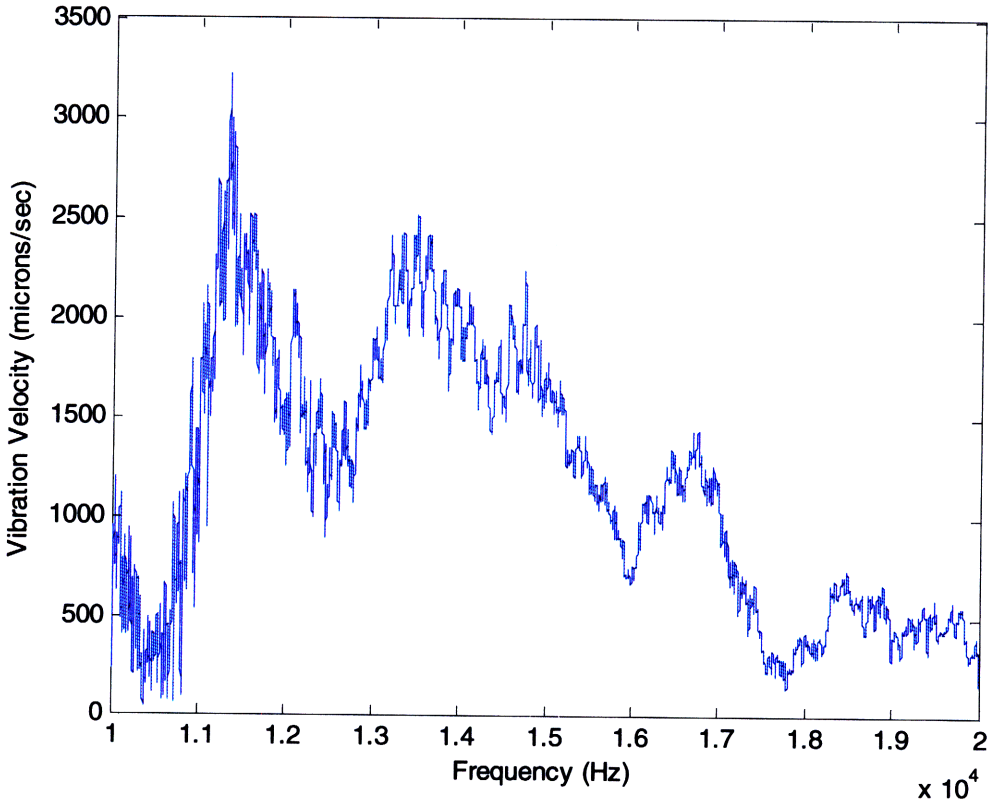

700
600
500
400
300
200
100
0
Vibration Velocity (microns/sec)
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

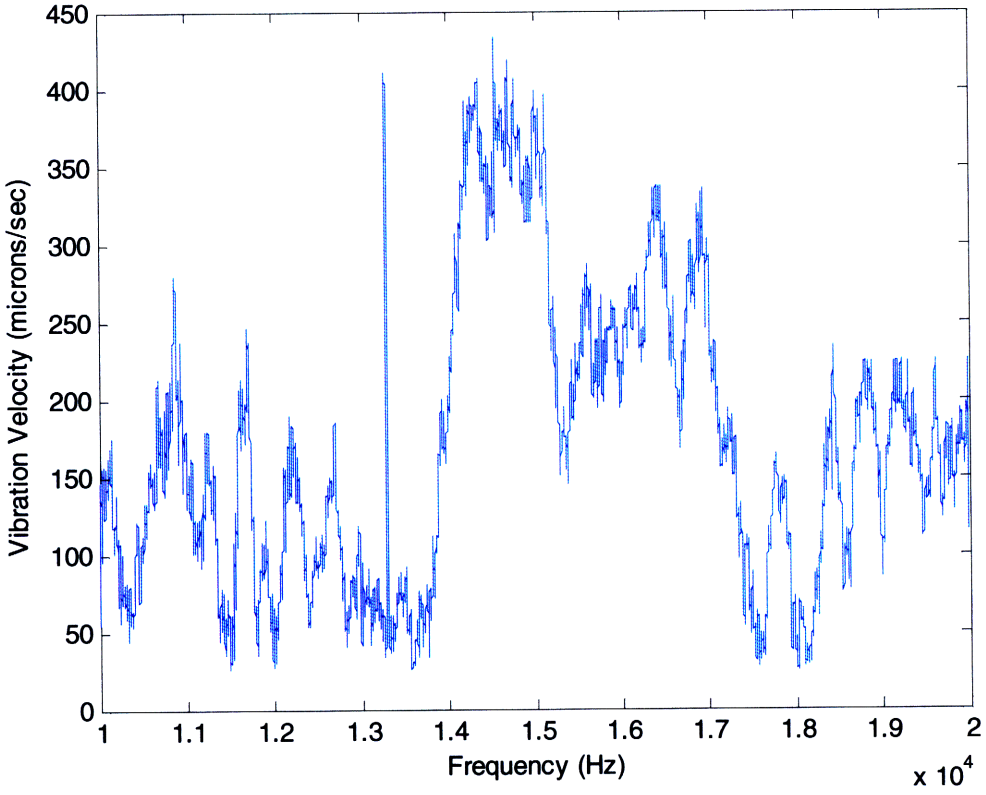
Target 2

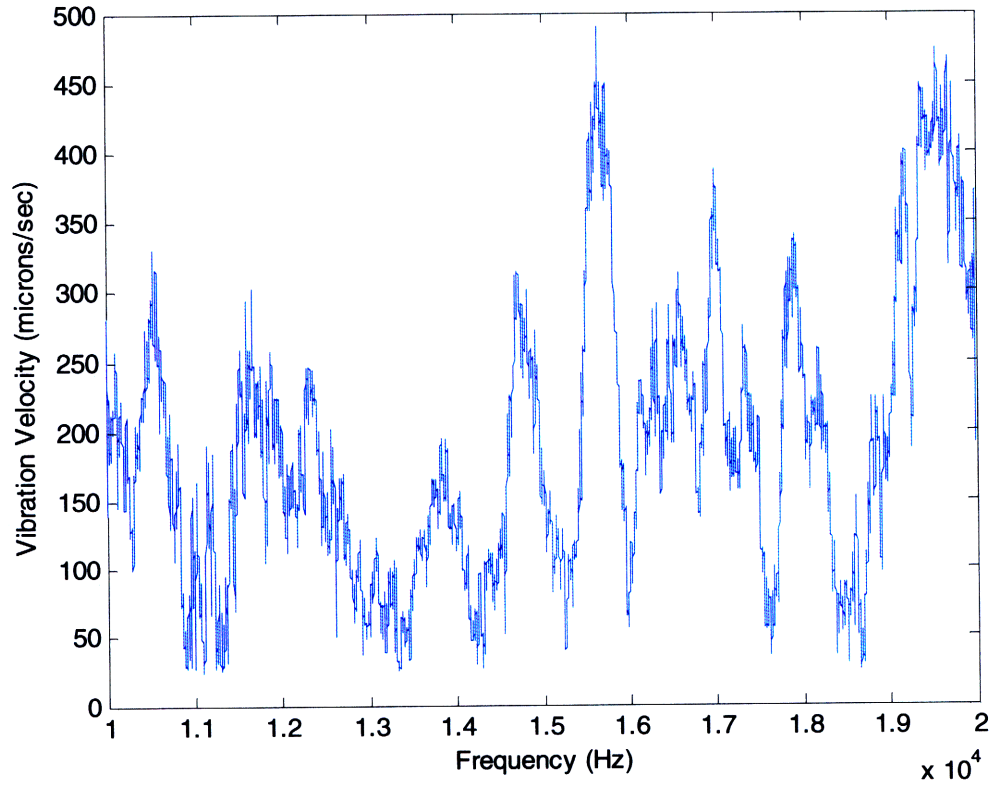

Vibration Velocity (microns/sec)
1800
1600
1400
1200
1000
800
600
400
200
0
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

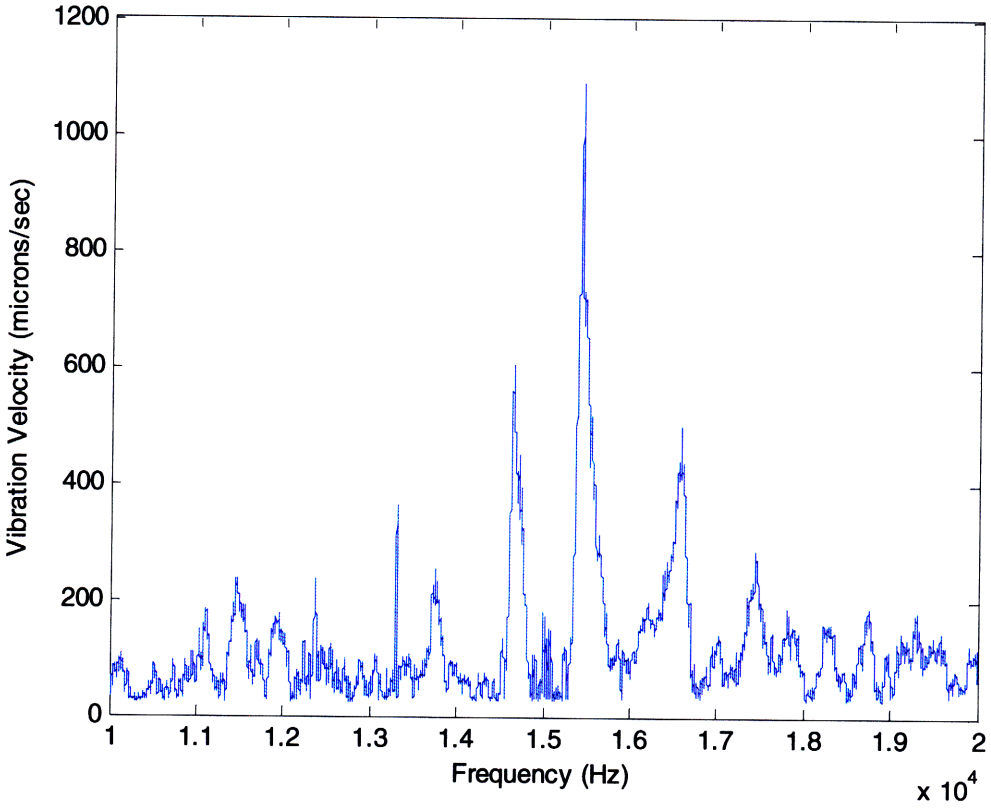

Vibration Velocity (microns/sec)
2500
2000
1500
1000
500
0
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

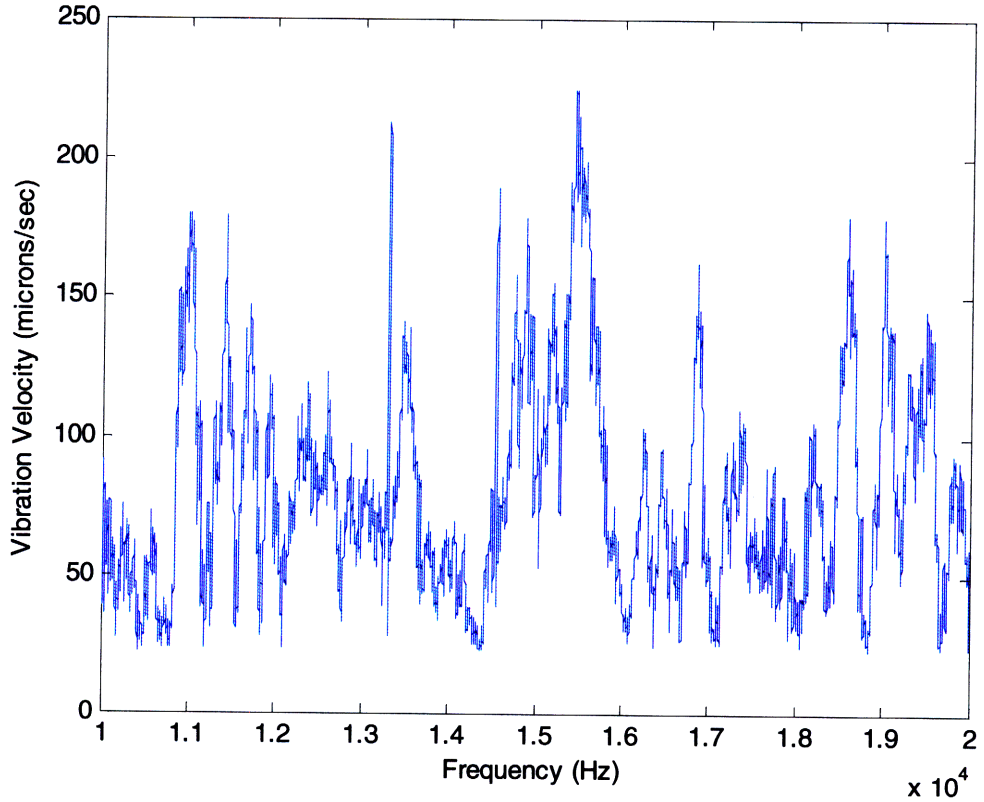

7000
6000
5000
4000
3000
2000
1000
0
Vibration Velocity (microns/sec)
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

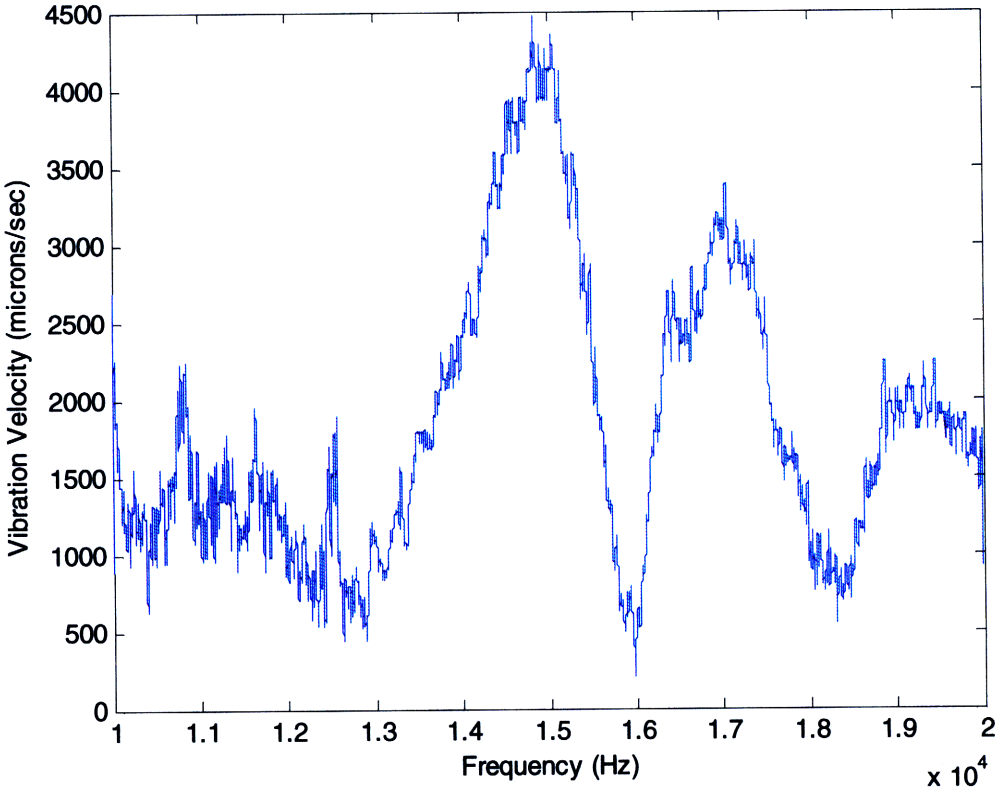

3500
3000
2500
2000
1500
1000
500
0
Vibration Velocity (microns/sec)
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

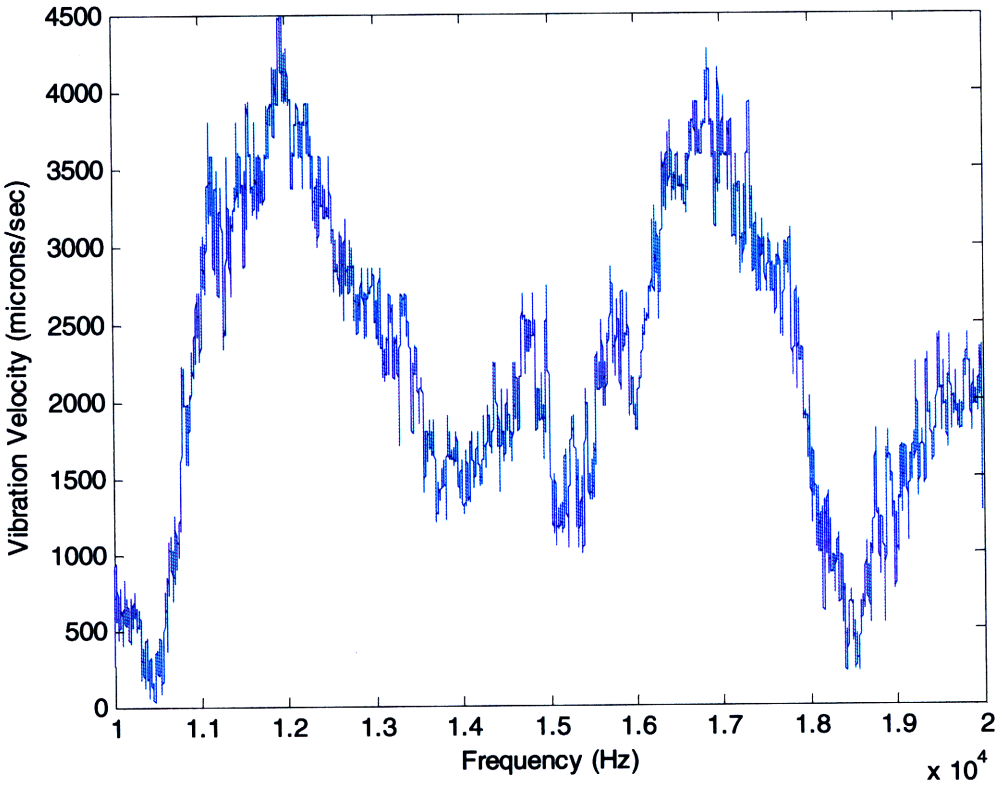

Vibration Velocity (microns/sec)
450
400
350
300
250
200
150
100
50
0
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

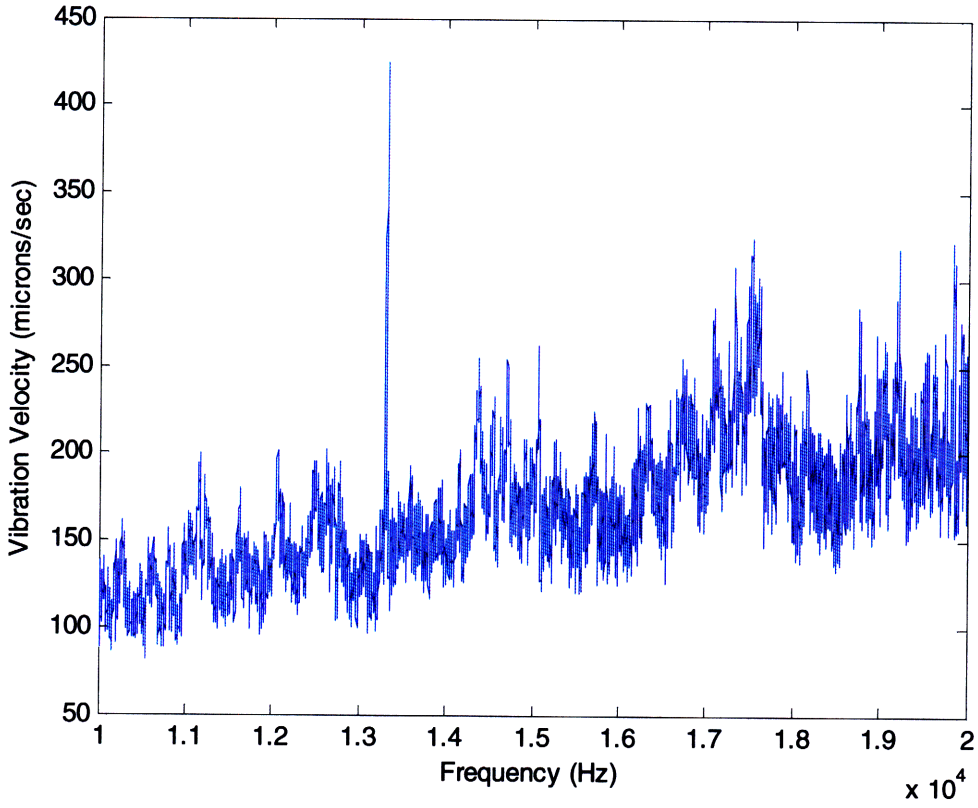

Vibration Velocity (microns/sec)
500
450
400
350
300
250
200
150
100
50
0
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

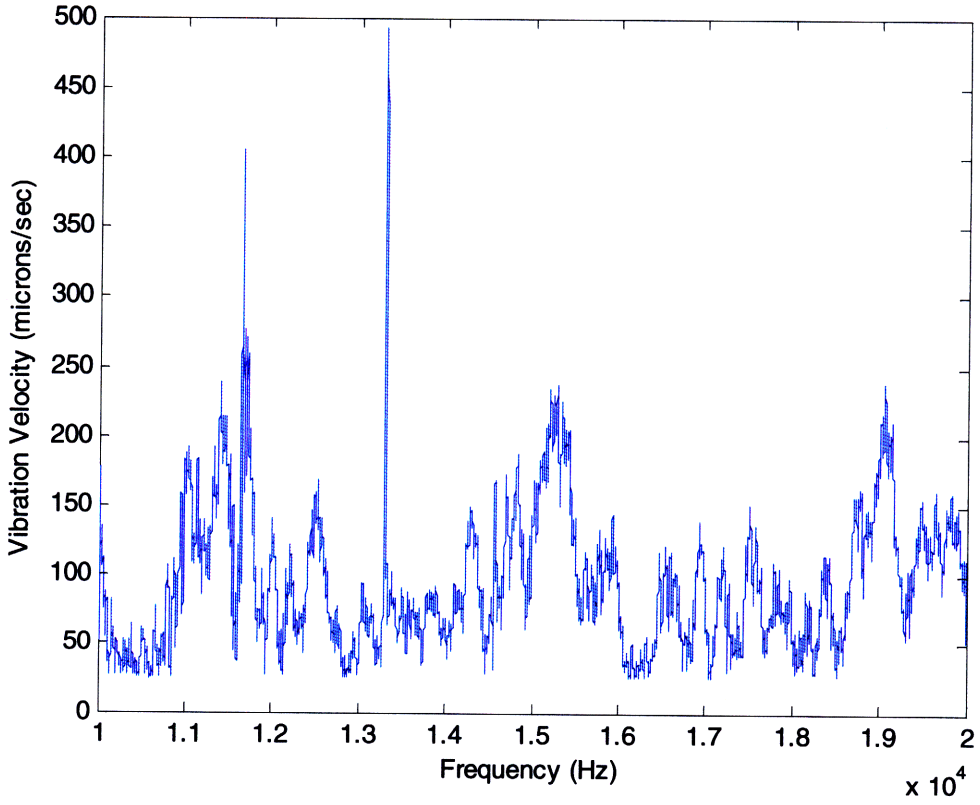

Vibration Velocity (microns/sec)
0
200
400
600
800
1000
1200
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

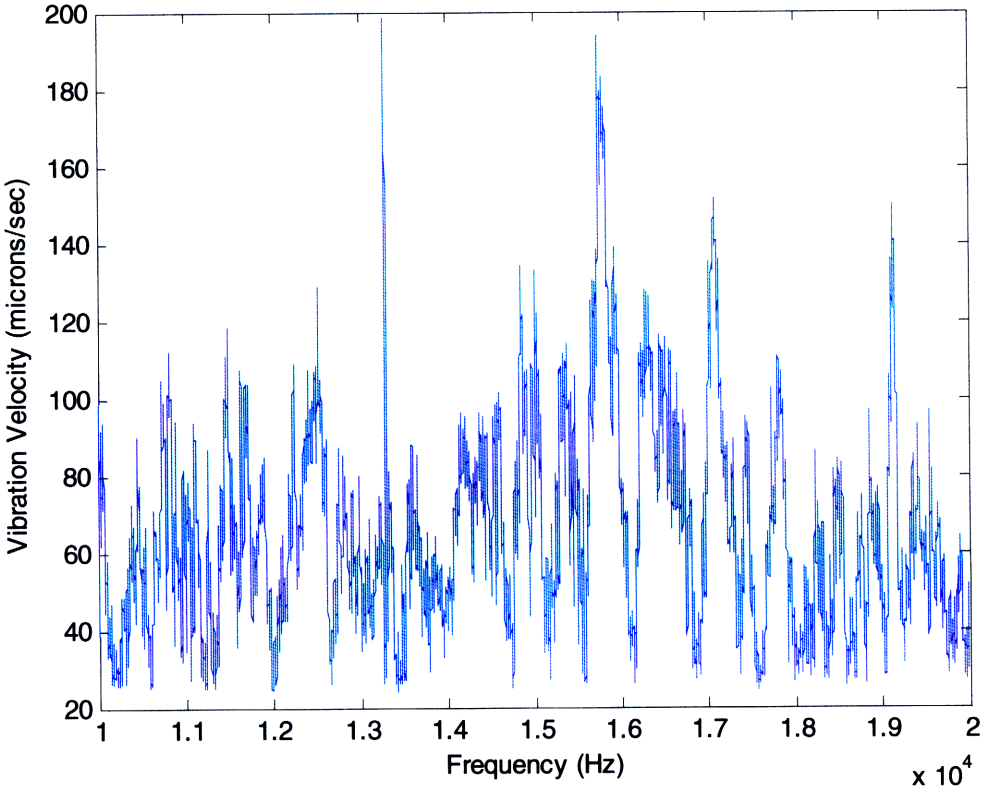

Vibration Velocity (microns/sec)
0
50
100
150
200
250
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4


Vibration Velocity (microns/sec)
4500
4000
3500
3000
2500
2000
1500
1000
500
0
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

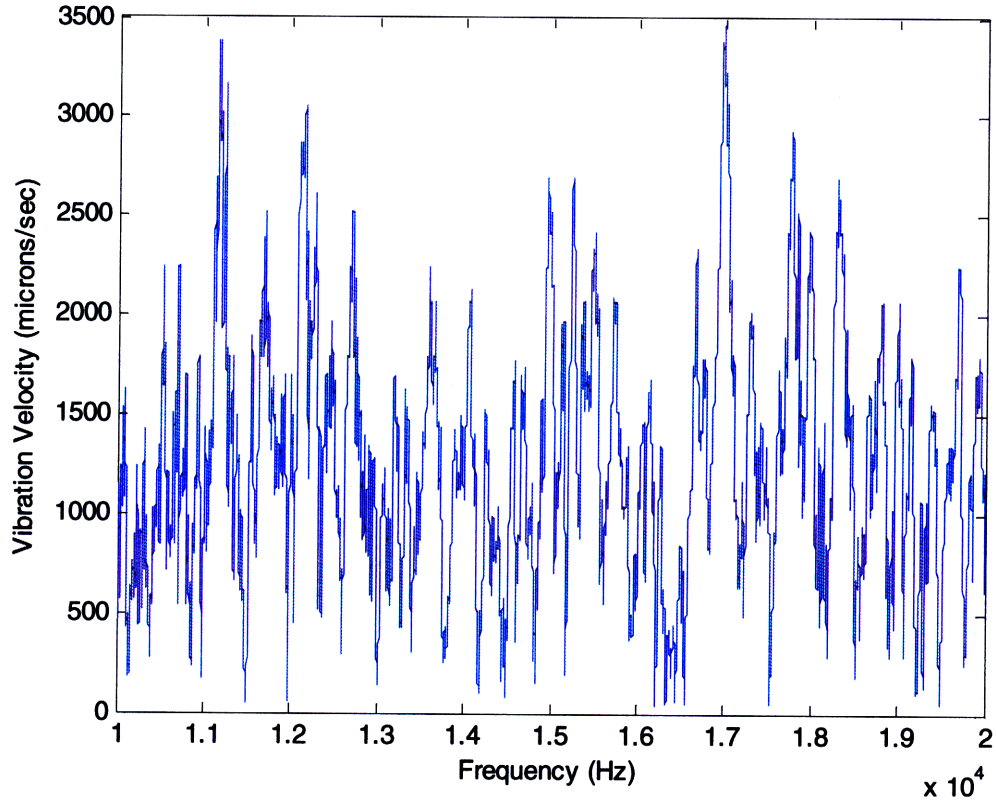

Vibration Velocity (microns/sec)
4500
4000
3500
3000
2500
2000
1500
1000
500
0
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

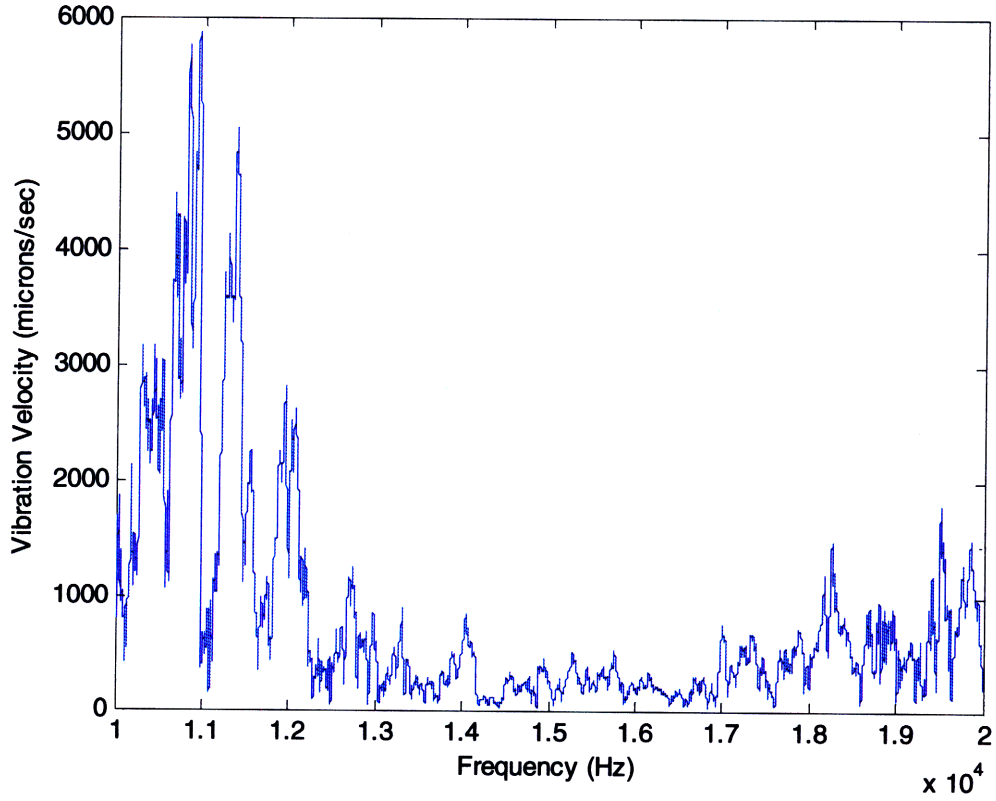

Vibration Velocity (microns/sec)
450
400
350
300
250
200
150
100
50
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

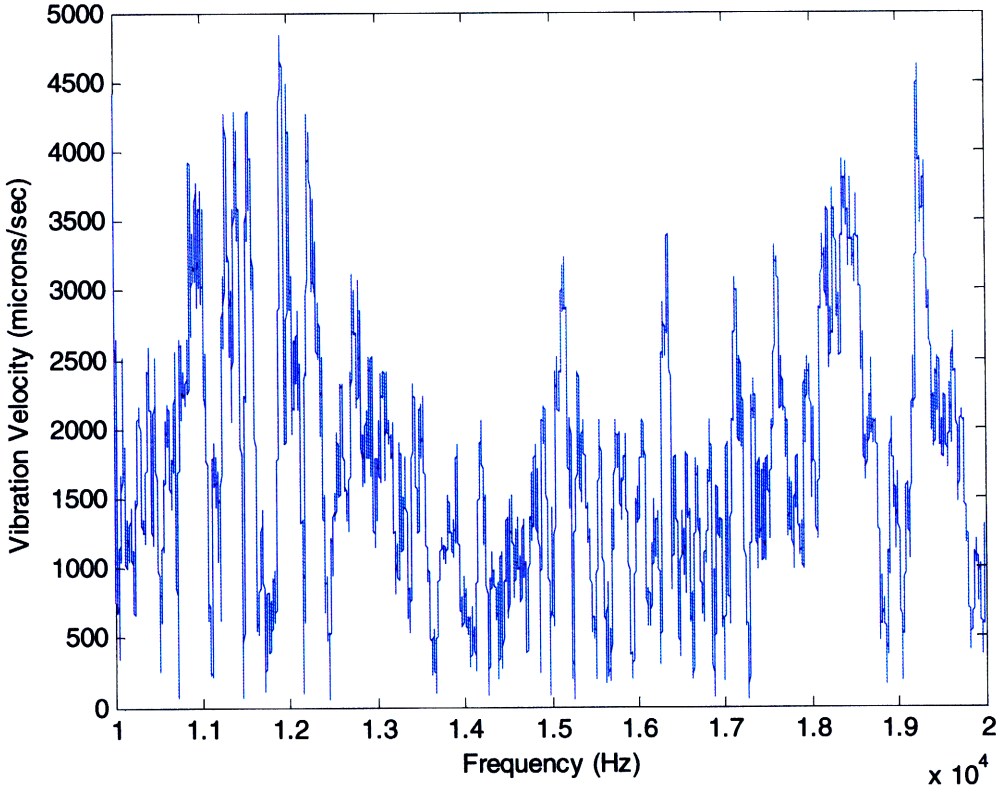

Vibration Velocity (microns/sec)
500
450
400
350
300
250
200
150
100
50
0
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

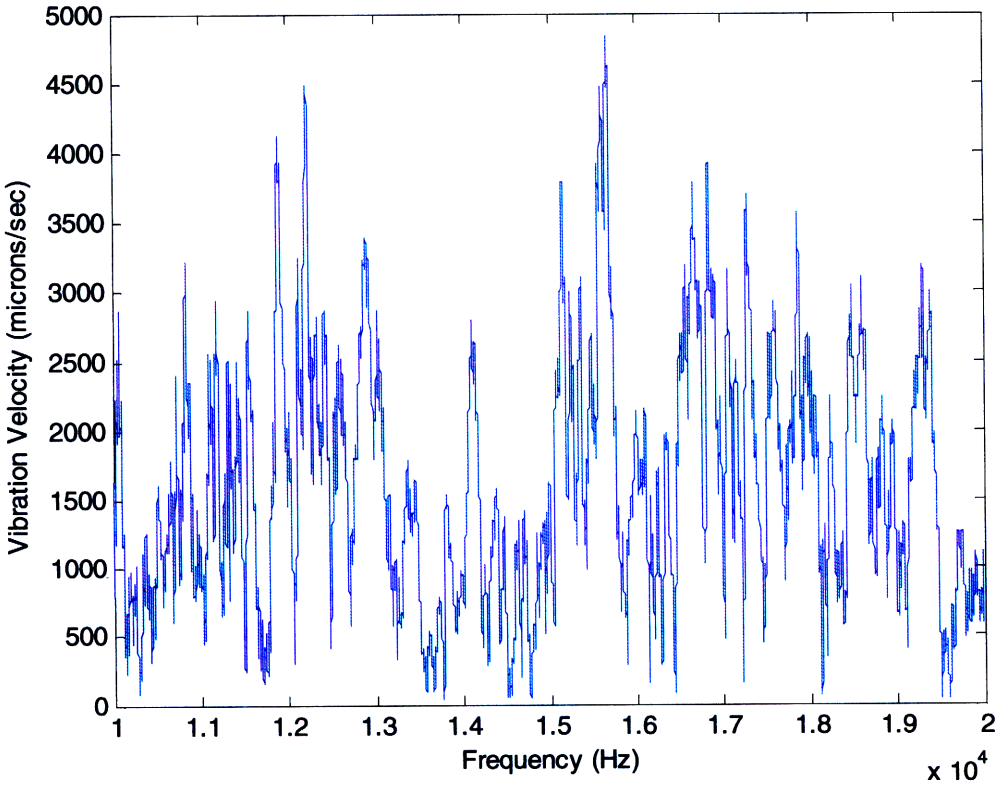

Vibration Velocity (microns/sec)
200
180
160
140
120
100
80
60
40
20
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

Soda Can

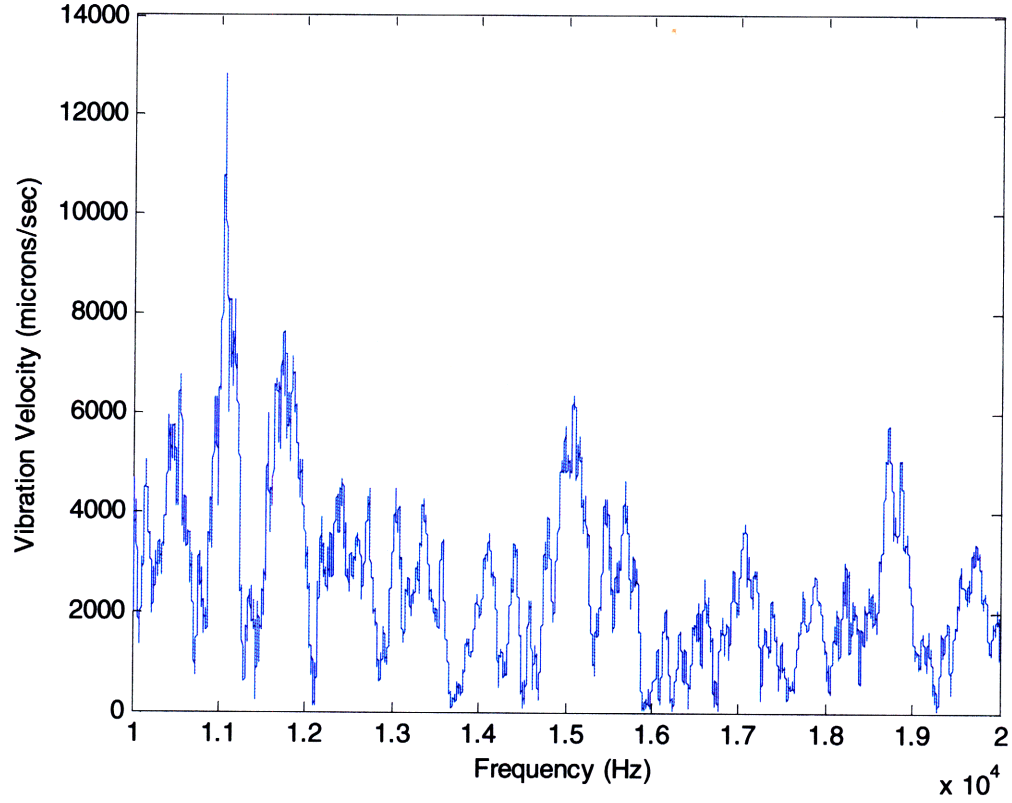

Vibration Velocity (microns/sec)
3500
3000
2500
2000
1500
1000
500
0
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

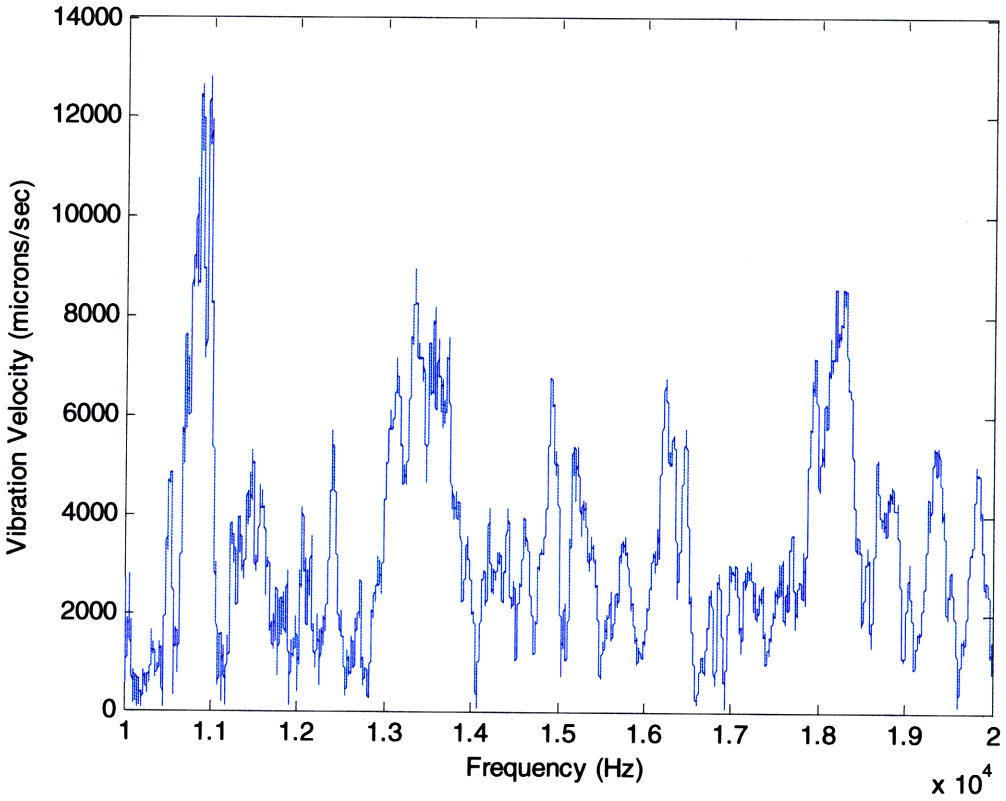

Vibration Velocity (microns/sec)
6000
5000
4000
3000
2000
1000
0
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

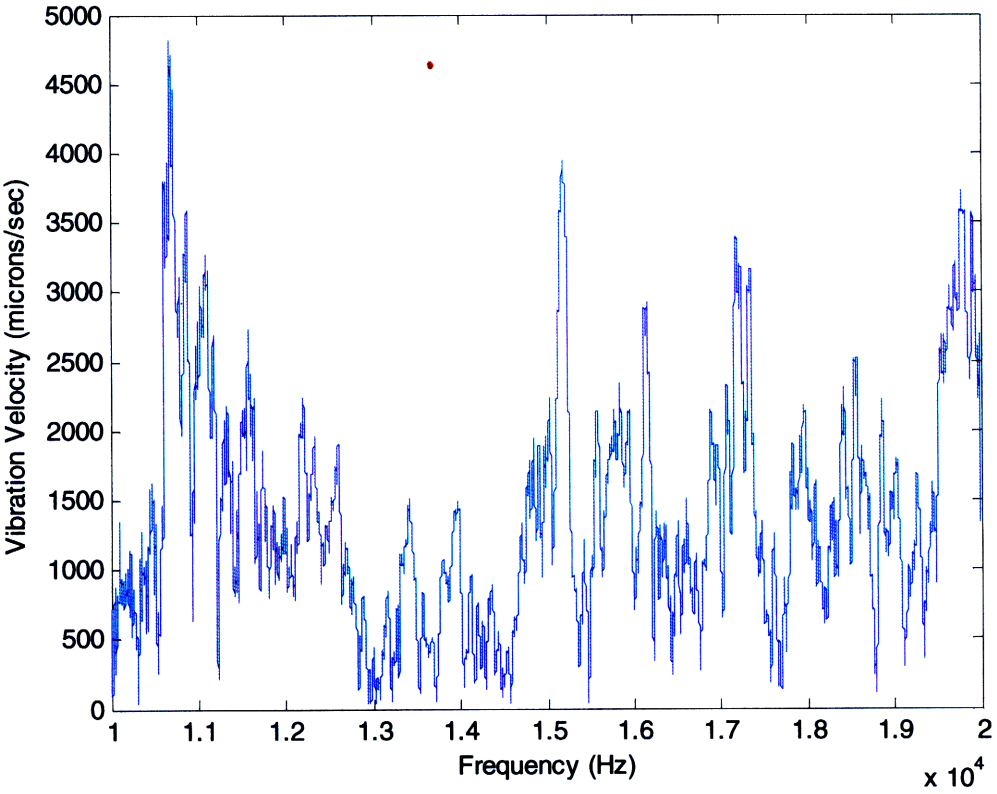

Vibration Velocity (microns/sec)
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

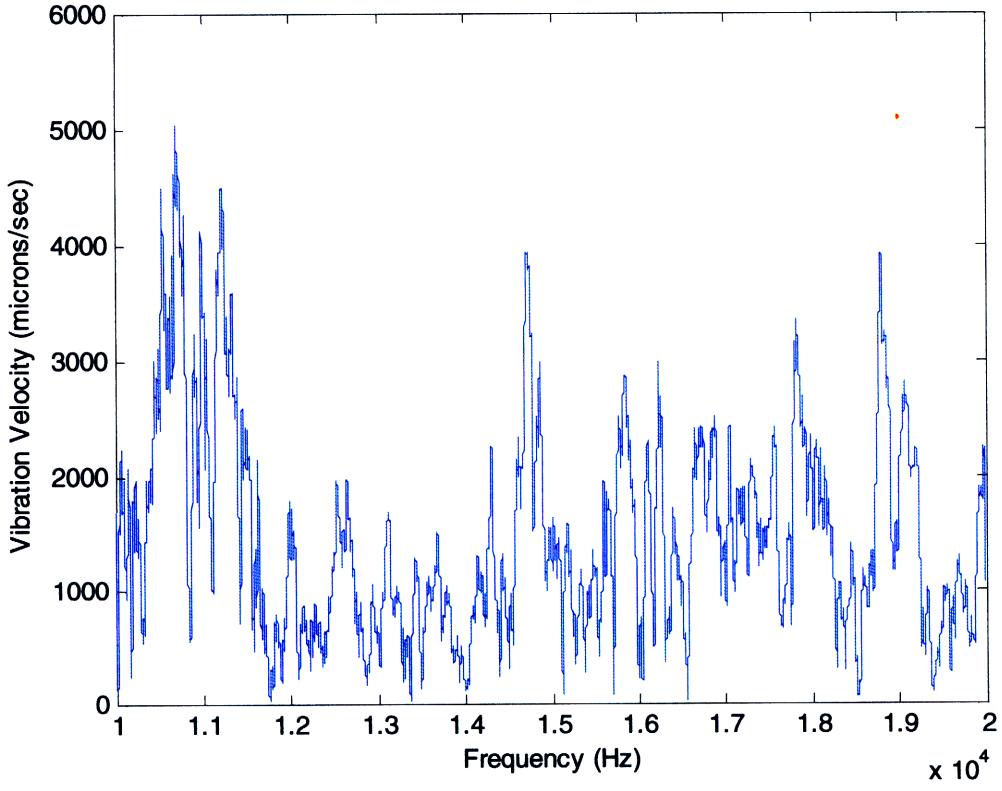

Vibration Velocity (microns/sec)
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

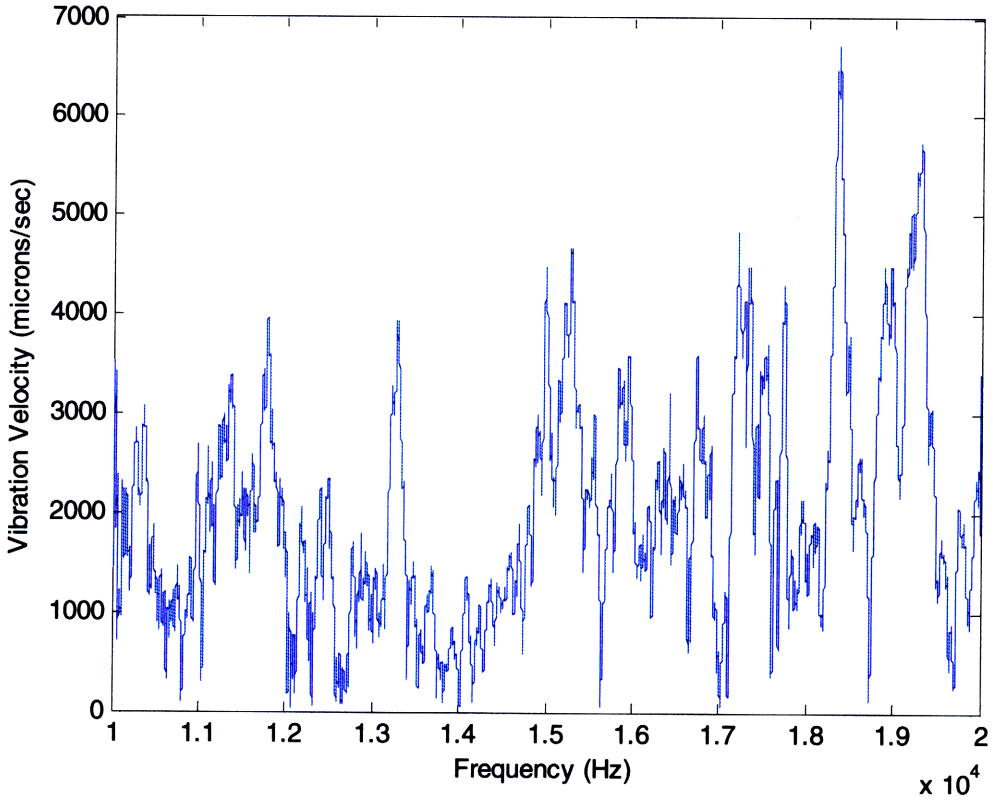
Crushed Soda Can



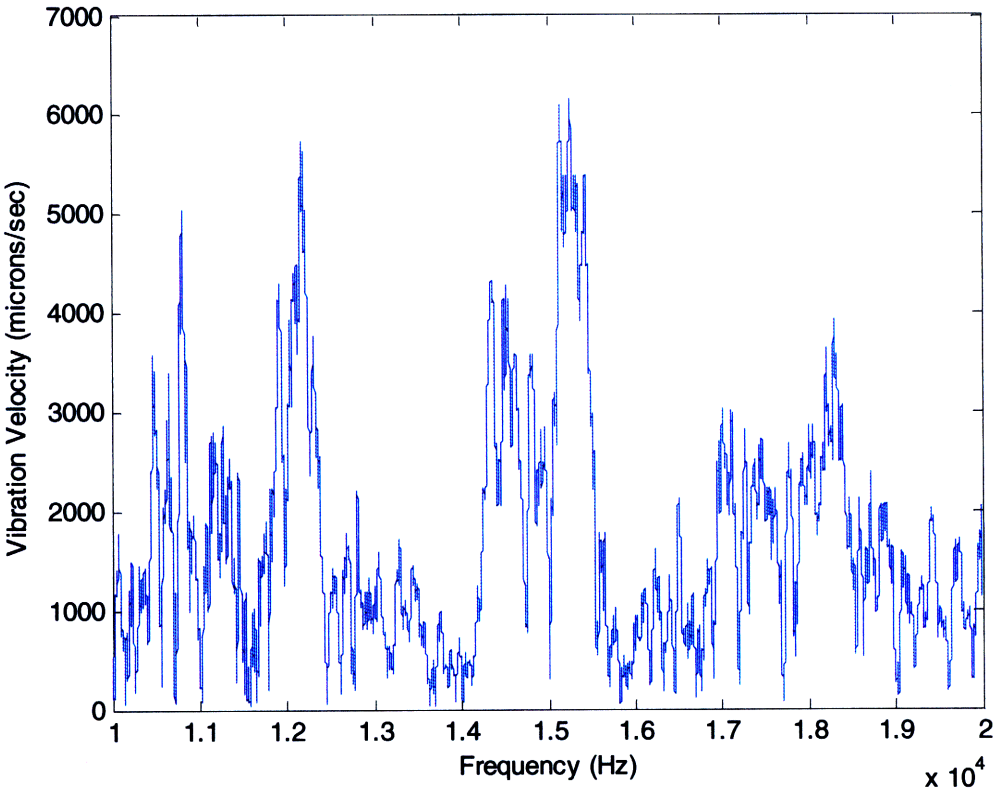


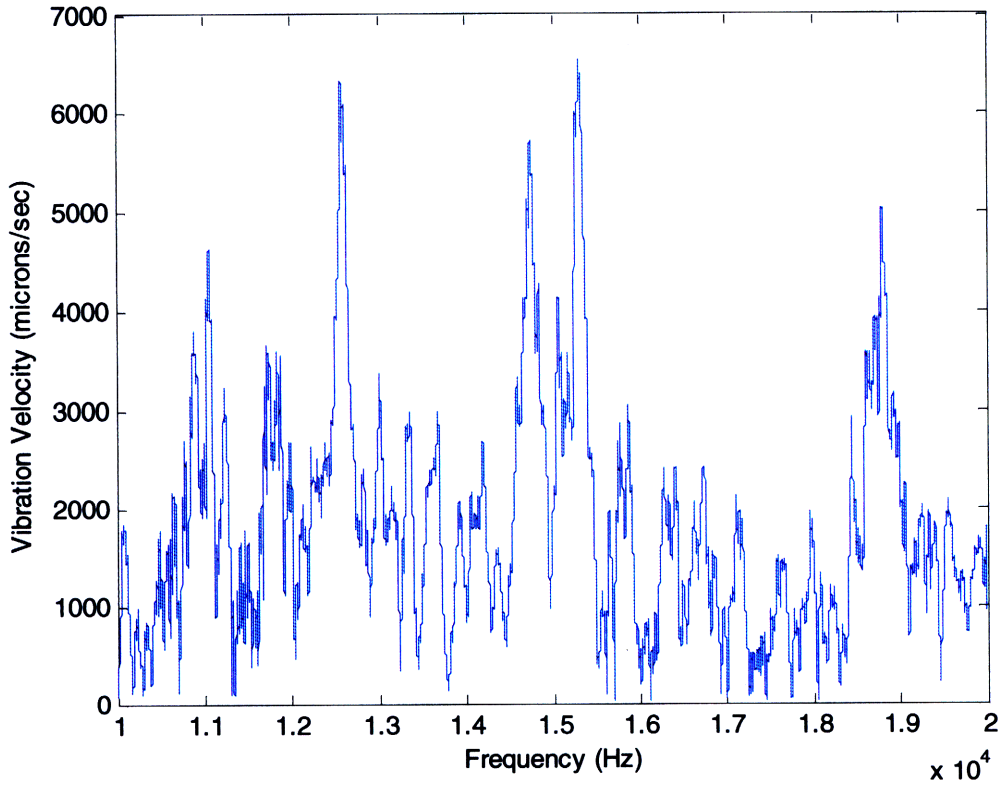

Vibration Velocity (microns/sec)
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

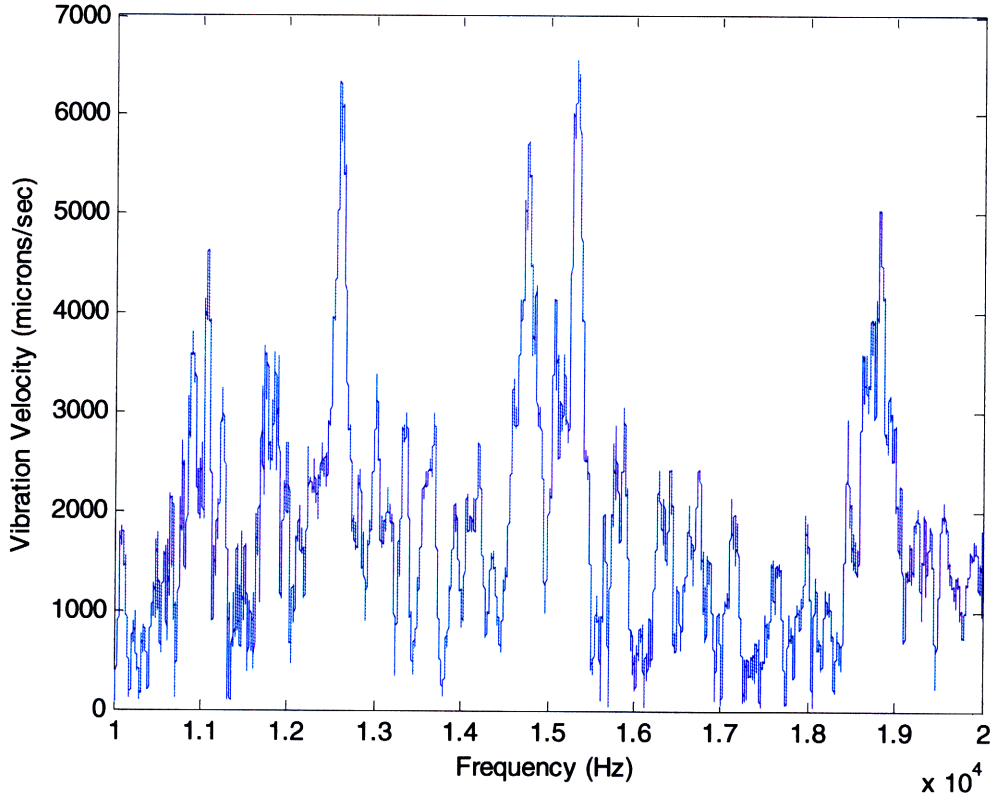

Vibration Velocity (microns/sec)
6000
5000
4000
3000
2000
1000
0
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

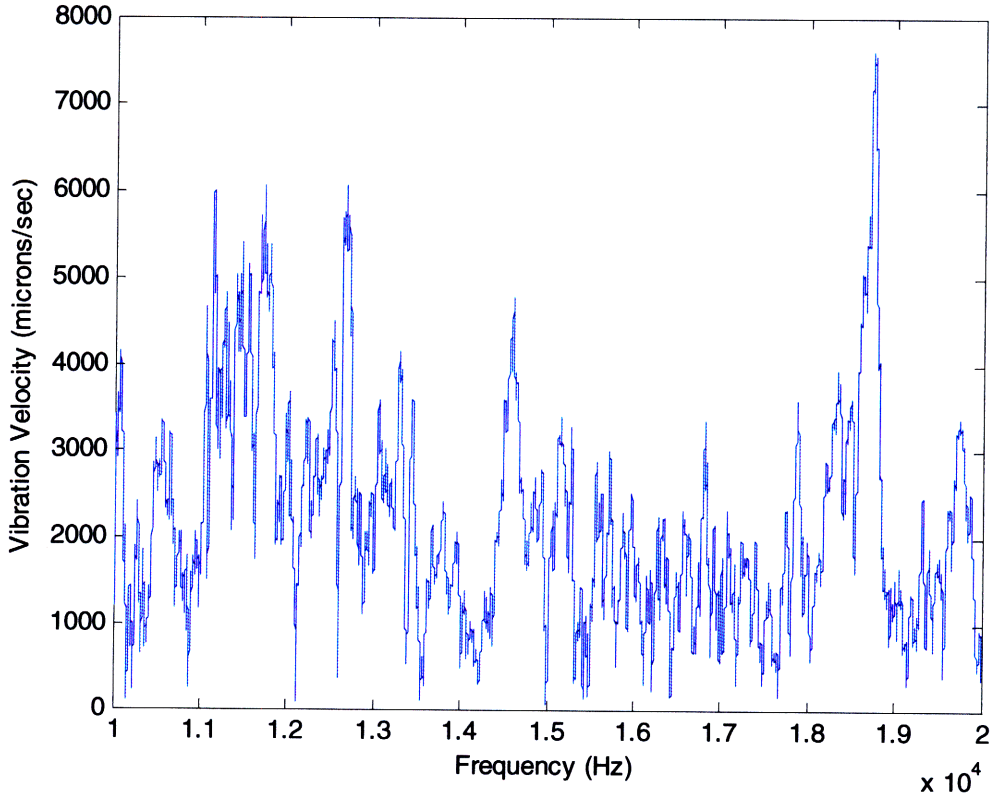

7000
6000
5000
4000
3000
2000
1000
0
Vibration Velocity (microns/sec)
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

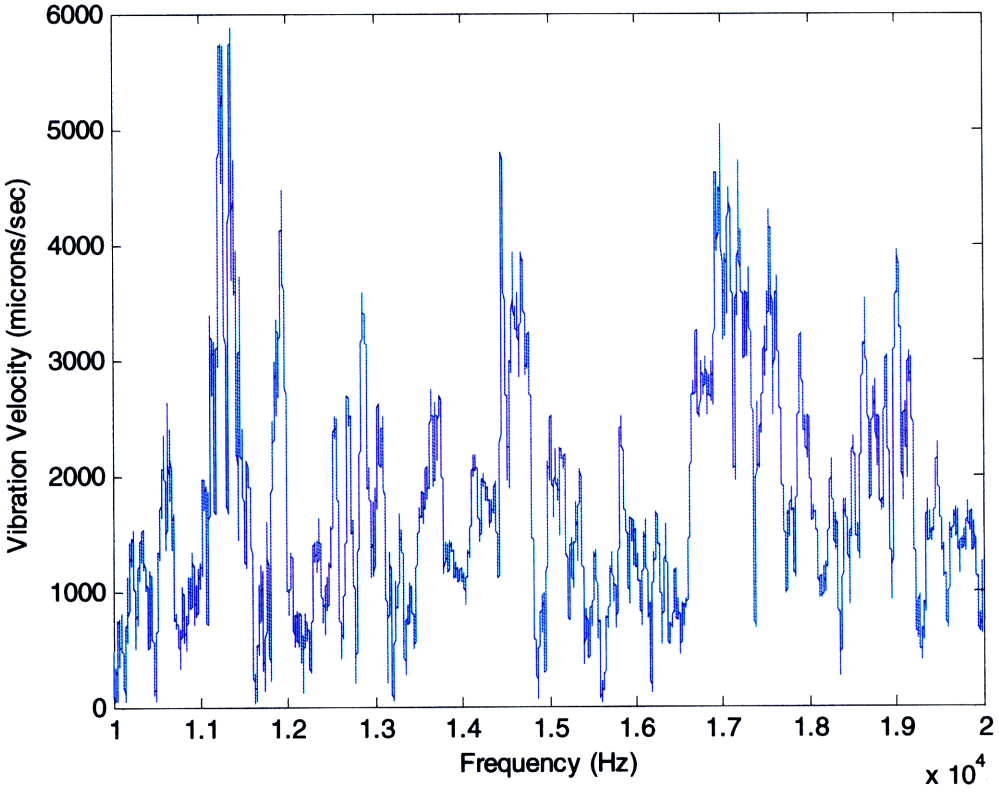
Foamcup


Vibration Velocity (microns/sec)
7000
6000
5000
4000
3000
2000
1000
0
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

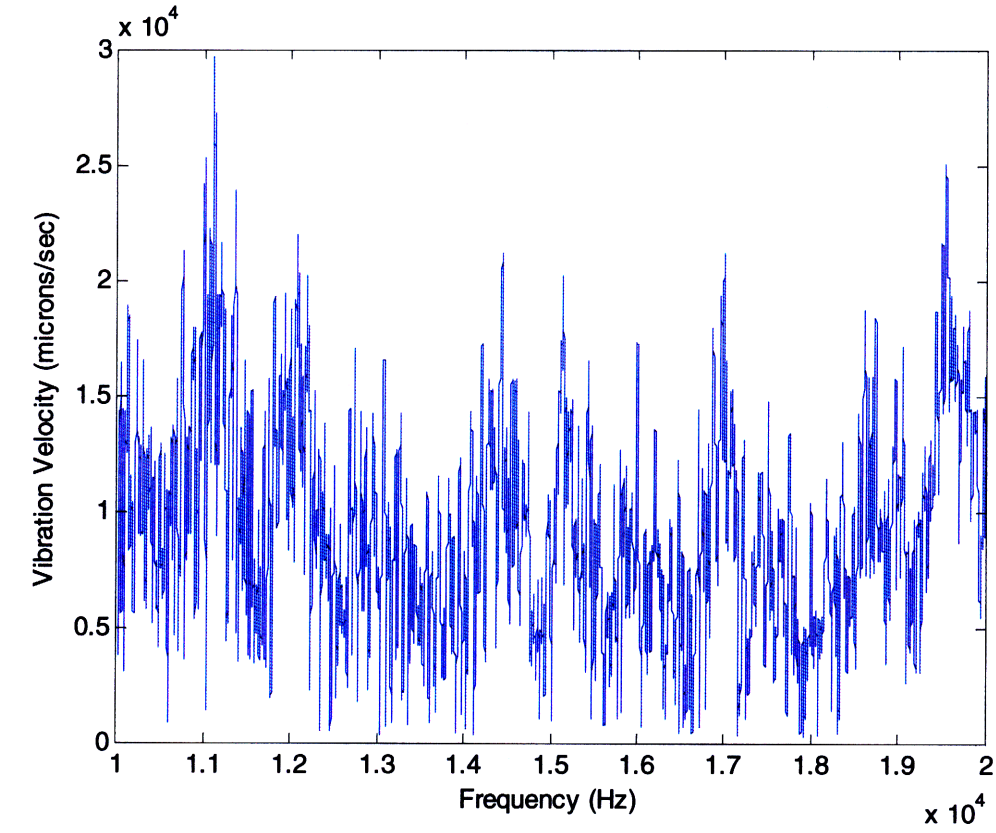

Vibration Velocity (microns/sec)
7000
6000
5000
4000
3000
2000
1000
0
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

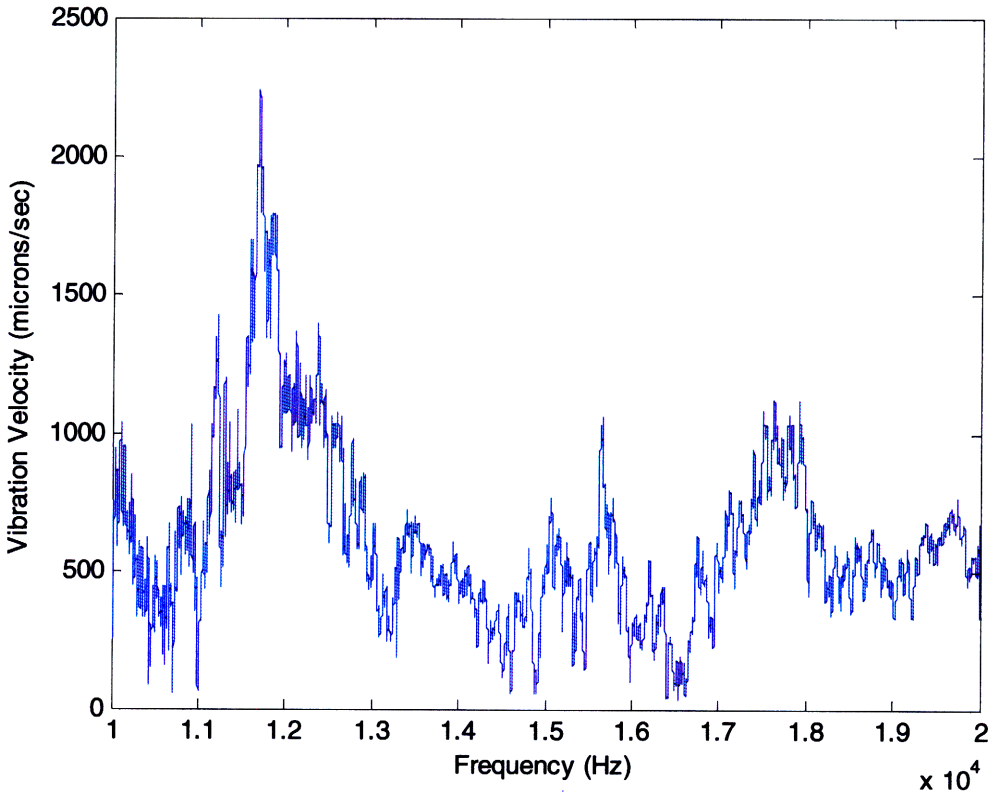

Vibration Velocity (microns/sec)
7000
6000
5000
4000
3000
2000
1000
0
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

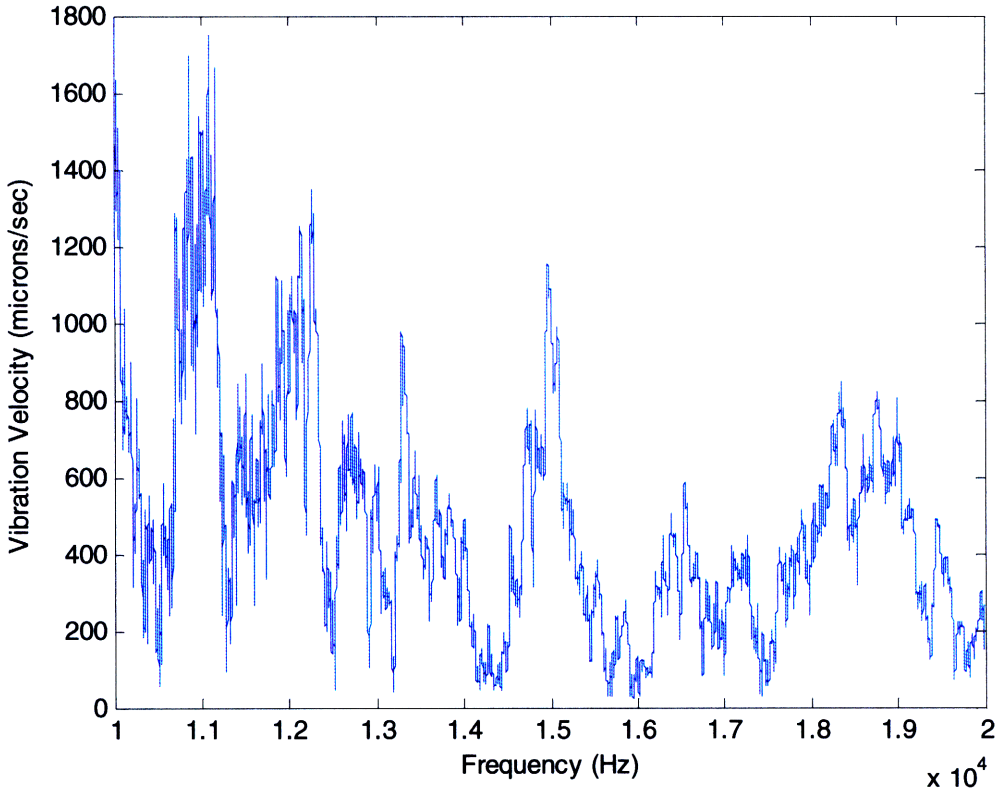

Vibration Velocity (microns/sec)
8000
7000
6000
5000
4000
3000
2000
1000
0
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

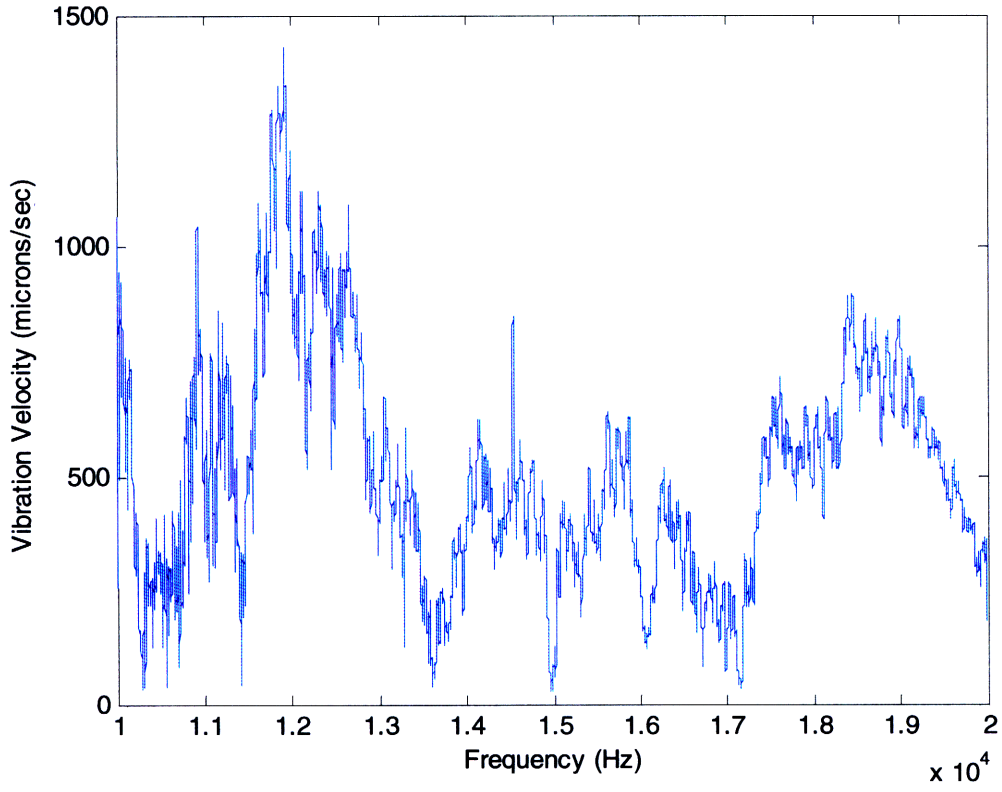

6000
5000
4000
3000
2000
1000
0
Vibration Velocity (microns/sec)
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

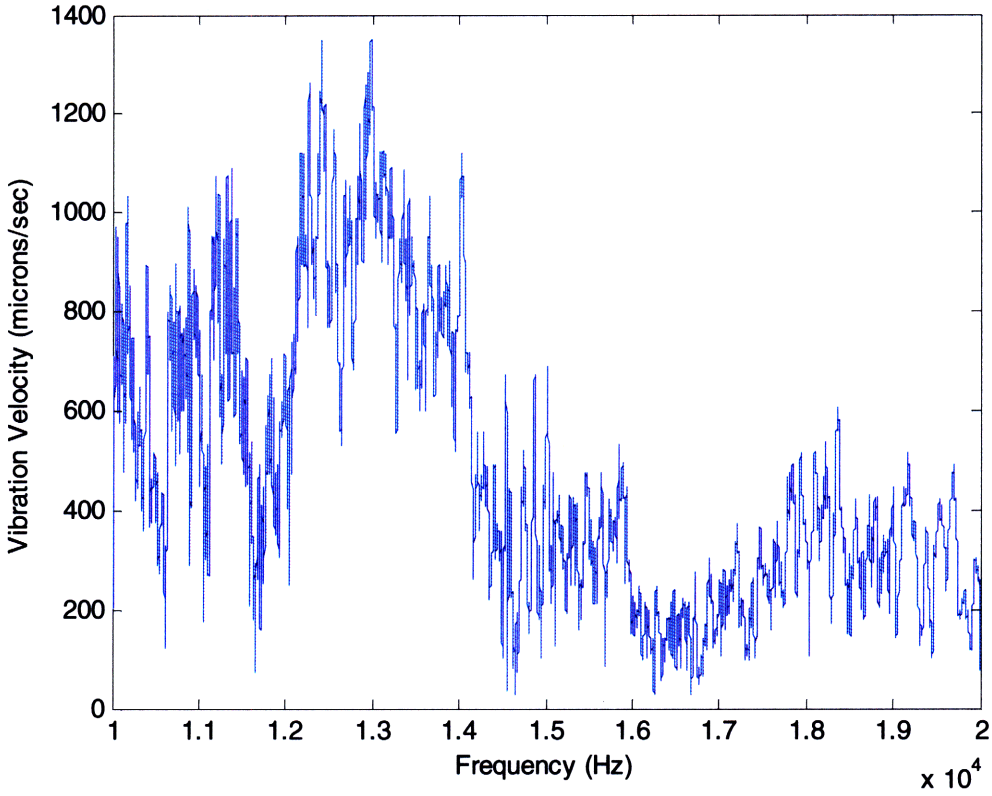
Waterbottle



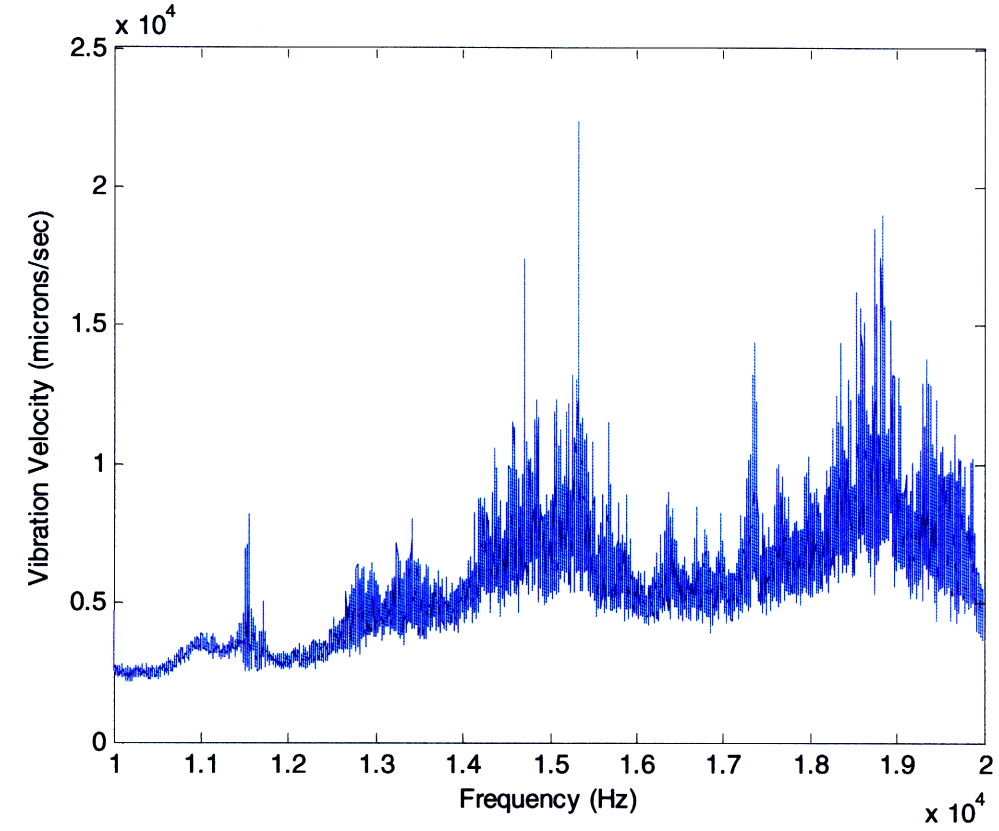


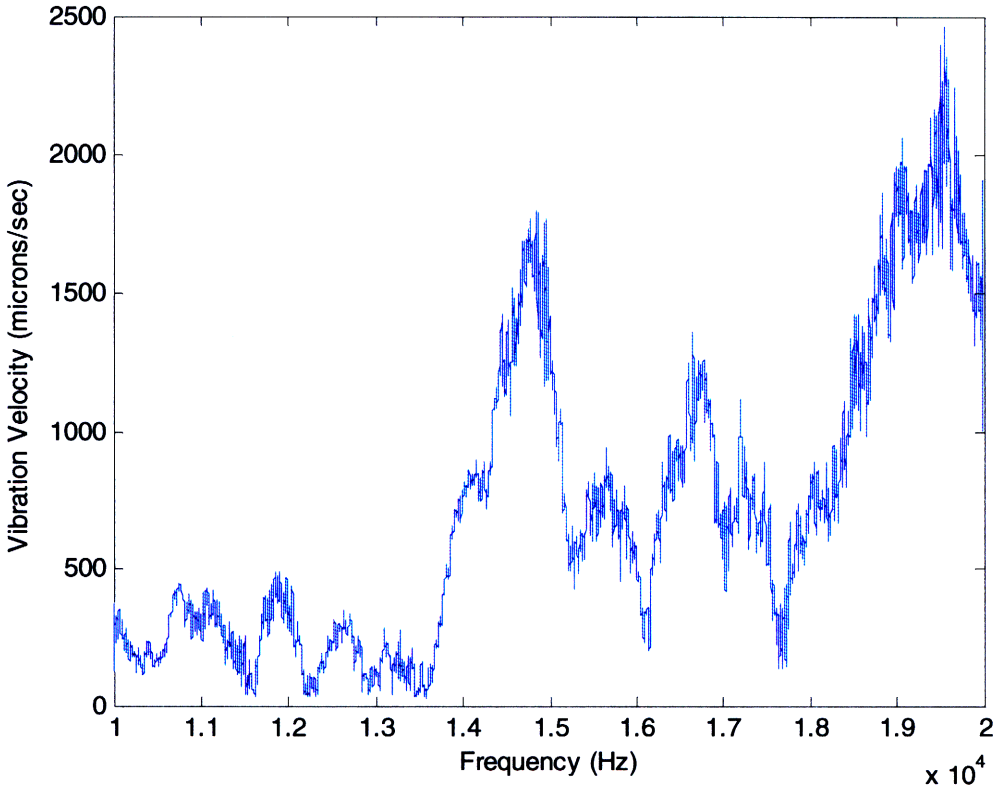

Vibration Velocity (microns/sec)
1800
1600
1400
1200
1000
800
600
400
200
0
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

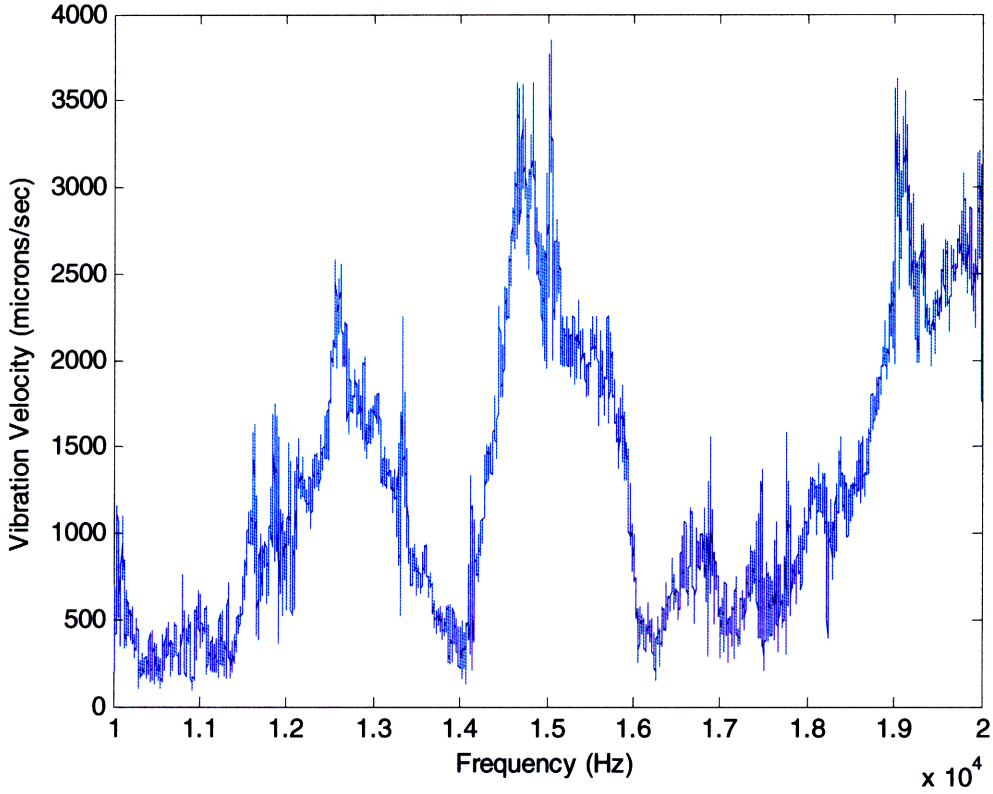

Vibration Velocity (microns/sec)
1500
1000
500
0
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

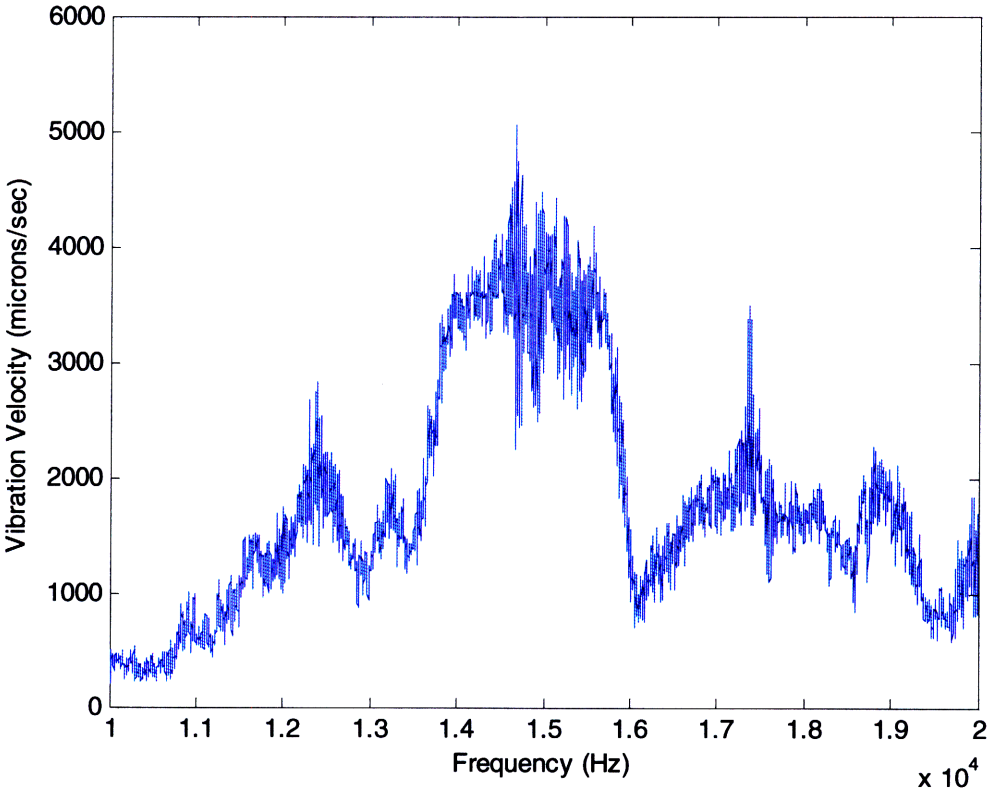

1400
1200
1000
800
600
400
200
0
Vibration Velocity (microns/sec)
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

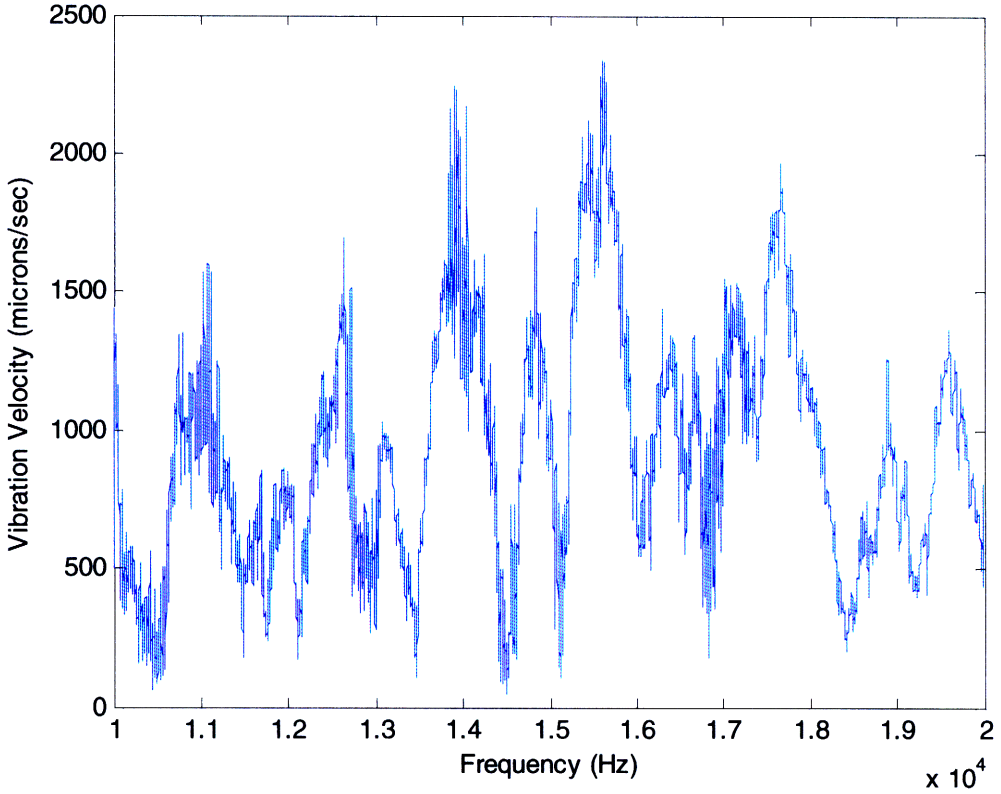
Trashbag

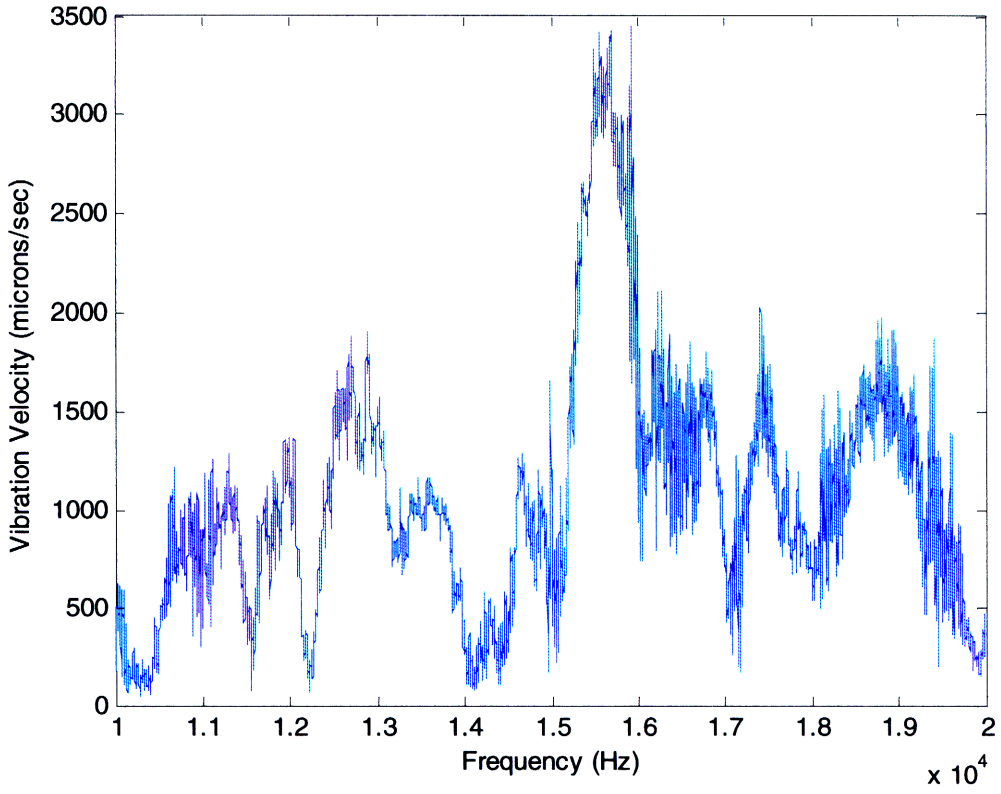

x 10^4
2.5
2
1.5
1
0.5
0
Vibration Velocity (microns/sec)
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4


2500
2000
1500
1000
500
0
Vibration Velocity (microns/sec)
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

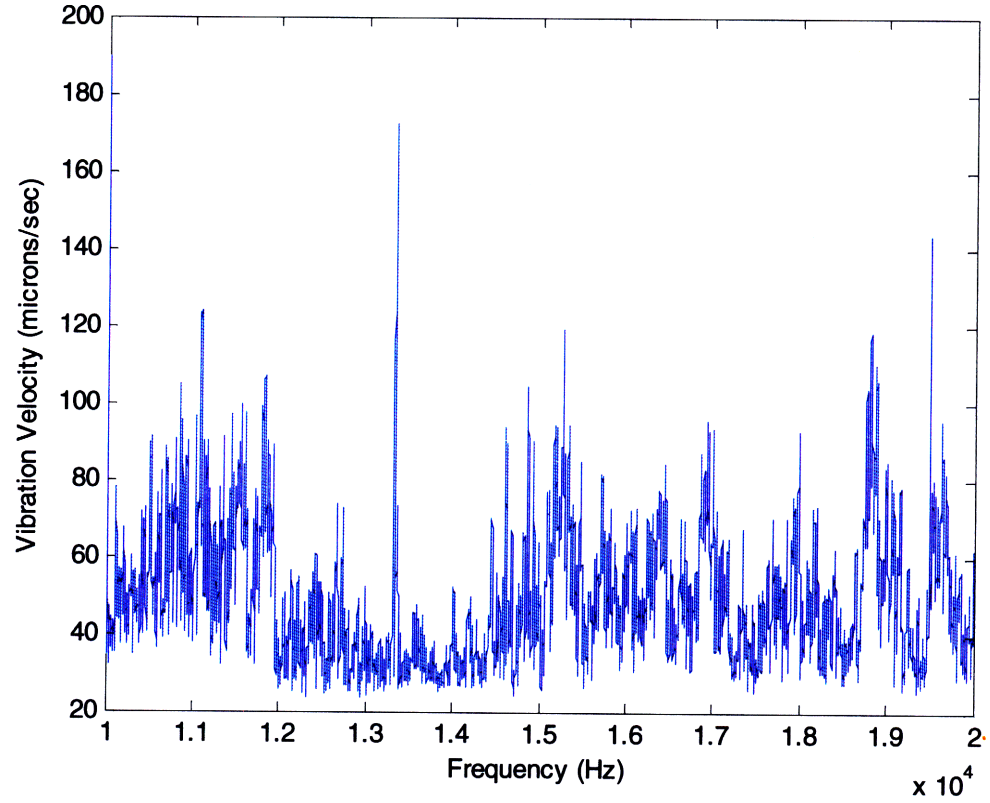

Vibration Velocity (microns/sec)
0
500
1000
1500
2000
2500
3000
3500
4000
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

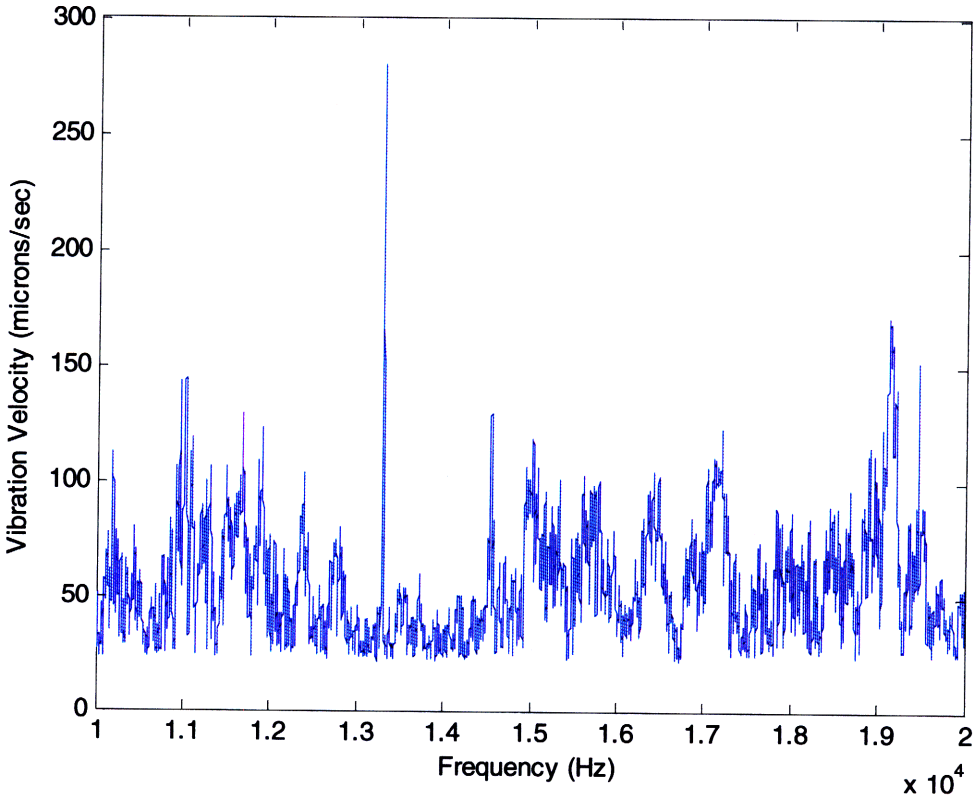

Vibration Velocity (microns/sec)
0
1000
2000
3000
4000
5000
6000
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

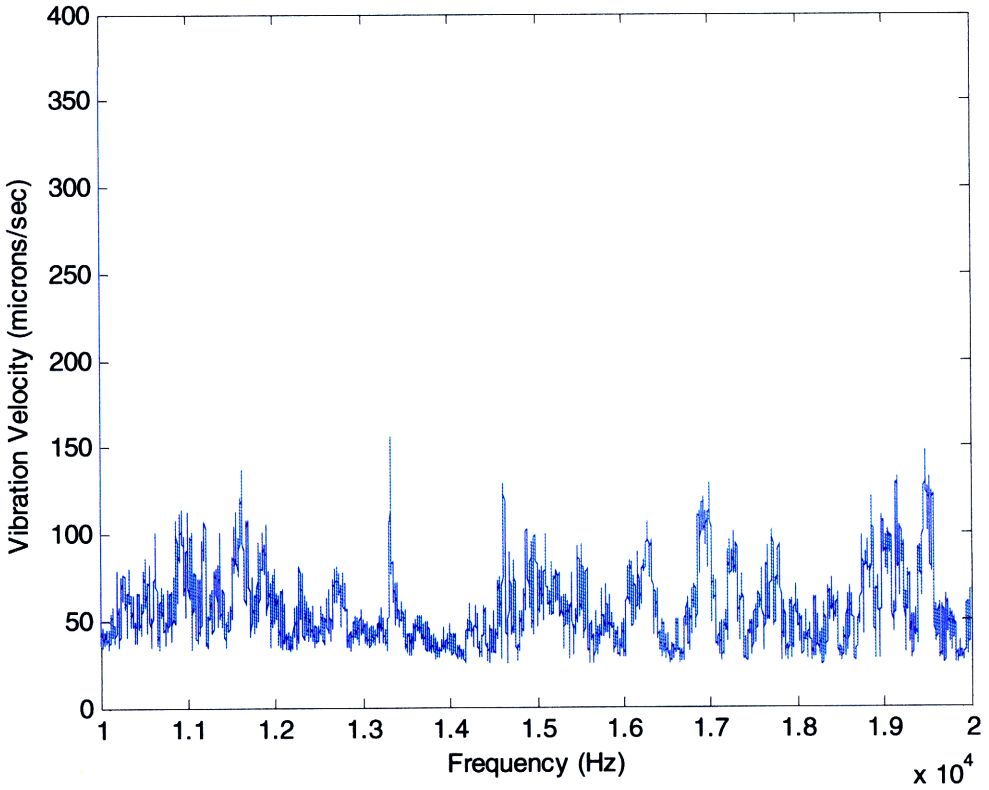

Vibration Velocity (microns/sec)
2500
2000
1500
1000
500
0
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

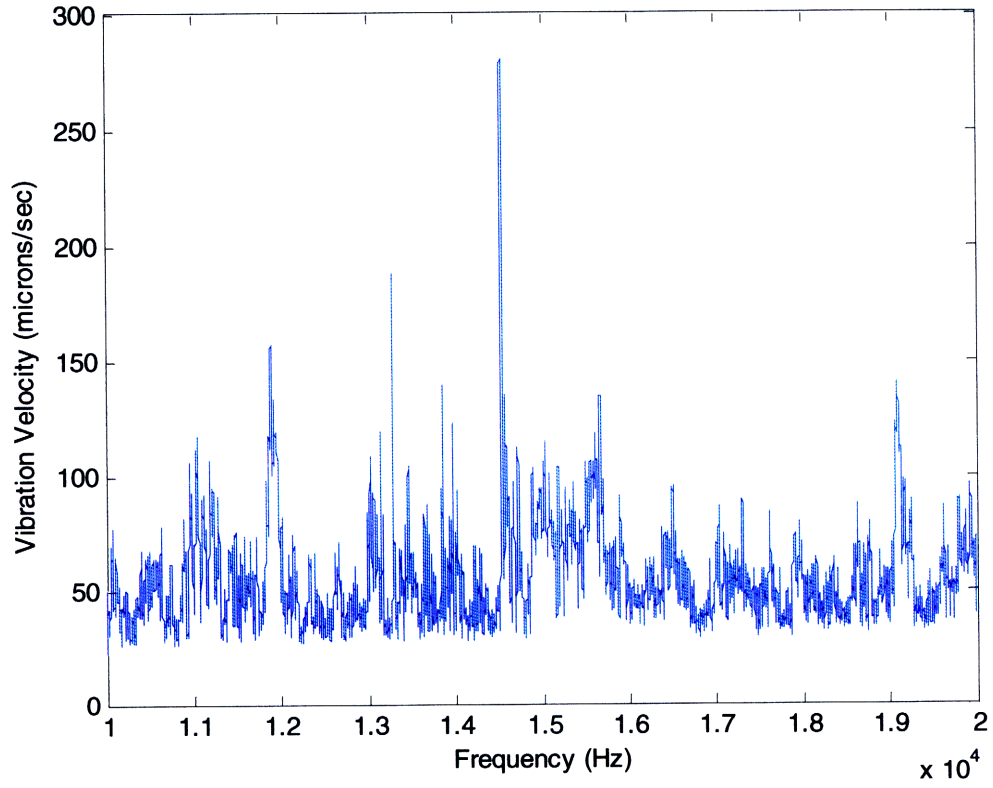

Vibration Velocity (microns/sec)
3500
3000
2500
2000
1500
1000
500
0
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4

Solid Aluminum Cylinder






Vibration Velocity (microns/sec)
400
350
300
250
200
150
100
50
0
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4


Vibration Velocity (microns/sec)
300
250
200
150
100
50
0
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
Frequency (Hz)
x 10^4