Non-Intrusive Water Utility Monitoring and Free-Space Load Monitoring

by

Sabrina M. Neuman

S.B., Massachusetts Institute of Technology (2009)

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2010

[February 2011]

Author ______________________________
Department of Electrical Engineering and Computer Science
September 7, 2010

Certified by ______________________________
Steven B. Leeb
Professor of Electrical and Mechanical Engineering
Thesis Supervisor

Certified by ______________________________
Christopher J. Schantz
Doctoral Candidate, Mechanical Engineering
Thesis Supervisor

Accepted by ______________________________
Christopher J. Terman
Chairman, Department Committee on Graduate Theses

Non-Intrusive Water Utility Monitoring and Free-Space Load Monitoring

by

Sabrina M. Neuman

Submitted to the Department of Electrical Engineering and Computer Science on September 7, 2010, in partial fulfillment of the requirements for the degree of Master of Engineering in Electrical Engineering and Computer Science

Abstract

This work presents a non-intrusive, single-point sensing scheme to monitor the water usage for various loads on a water utility pipe network through the vibration of a pipe near the water intake source. Experiments with the water utility sensor provided data sufficient to identify individual loads on the water distribution network both alone and during operation of multiple loads. This sensor setup is useful for smart-metering applications to promote water conservation by keeping track of the operational schedule of individual loads on the local water network.

This work also presents the development of a free-space sensor to provide information about the operation and location of electrical loads: an electroquasistatic (EQS) sensor to detect voltage-mode events. The free-space sensor was able to detect events in a room, such as the activation of a line upon turning on a power strip or switching a light switch. This sensor could supplement a power monitoring system by helping to localize the activation of loads.

Thesis Supervisor: Steven B. Leeb
Title: Professor of Electrical and Mechanical Engineering

Thesis Supervisor: Christopher J. Schantz
Title: Doctoral Candidate, Mechanical Engineering

Acknowledgements

I would like to thank everyone who helped me over the course of this research: My colleague Sean Muller and his family, who were kind enough to tolerate my repeated intrusion on their home to collect field data for the non-intrusive water utility monitoring system. My colleague Sarah Page, who assisted me with so many of those field experiments. My colleague John Cooley, whose advice and supervision guided the development of the free-space electroquasistatic (EQS) sensor. My colleague Zachary Remscrim, who has been my invaluable laboratory partner for all of my experiments, problem-solving, and analysis. My thesis co-supervisor Christopher Schantz, who was my expert guide and resource for all things mechanical. Finally, my thesis supervisor Professor Steven B. Leeb, who has been my role-model and mentor. I have learned so much from Professor Leeb over the years, as his student in the classroom and in the laboratory. His advice on all matters has shaped my academic career, and I am extremely grateful to have worked for him.

I thank the Laboratory of Electromagnetic and Electronic Systems (LEES) within the Research Laboratory of Electronics (RLE) for making this work possible.

I thank my family and my friends for supporting me during my work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Obtaining detailed information about individual loads on a large utility network (sometimes called smart-metering) is a necessary step in promoting efficient use of resources. Utility distribution systems are plagued by inefficient loads on the system with unexpectedly high resource usage, which may indicate that the loads are faulty or that perhaps too many loads in concurrent operation are putting a strain on the system. Obtaining load information with traditional multi-sensor networks is difficult and expensive due to the challenge of installation and maintenance of the many distributed sensors.

1.2 Non-Intrusive Load Monitoring

The Non-Intrusive Load Monitor (NILM) [1], [2], [3] is an effective proposed solution to the smart-metering problem. For example, an electrical NILM system takes aggregate current and voltage measurements from a single power intake point, such as the breaker box in a house, that feeds all the loads on that local network. From this one access point, and with a single monitoring unit taking aggregate measurements, it is possible to individually determine the power consumption of each load on the network.

NILM-inspired approaches can be extended to utilities besides electrical power, such as water distribution networks. From a single monitoring point at the source of water intake, it is possible to collect information about the operation schedule and water usage of each of the individual water-drawing loads on the system.

One limitation of existing electrical NILM systems lies in the fact that they cannot discern the difference between two identical loads operating at different physical locations (such as two similar light bulbs operating in different rooms of a house). To address this limitation, the development of free-space local sensors to supplement the NILM system is desirable.

1.3 Non-Intrusive Water Utility Monitoring

This work presents a non-intrusive monitoring approach that makes water utility distribution networks function as "dual use" systems. These "dual use" systems provide water usage and diagnostic information about loads on the system while delivering water to them.

A single non-intrusive vibration sensor is attached to a section of pipe near the point of aggregate water intake to a local water pipe network, such as the plumbing system of a building. This sensor provides data on vibrations induced by the operation of water fixtures and appliances on the network. The data can then be processed by software to identify individual loads on the water distribution network whether operated alone or simultaneously, as well as provide diagnostic information such as water flow rate. This sensor setup is useful for smart-metering applications to promote water conservation by keeping track and informing the consumer of the operational schedule and water draw of individual loads on their local water network.

This work describes some of the basic mechanical principles employed by the non-intrusive water utility monitor to to glean water delivery information from a single-point vibration sensor. Also described are experimental laboratory setups that demonstrate the relationship between data from the vibration sensor, and the physical phenomena that cause the vibration characteristics for loads on the pipe system. Experimental data both from the laboratory setups and from deployment of the prototype system in a single-family home is presented, and the analysis techniques performed on the data are described. The results from this data demonstrate that individual loads on the system are identifiable from a single monitoring point. They also demonstrate that individual loads remain identifiable and distinguishable even when multiple loads are in use.

1.4 Free Space Load Monitoring

This work also presents the preliminary development of a free-space sensor to provide information about the operation and location of electrical loads: an electroquasistatic (EQS) sensor to detect voltage-mode events. The free-space EQS sensor was able to detect events in a room, such as the activation of a line upon turning on a power strip or switching a light switch. This sensor could supplement a power monitoring system by helping to localize the activation of loads. Design considerations and preliminary experimental data for the EQS sensor are presented.

1.5 Thesis Organization

The work in this thesis is presented in five chapters and a conclusion. Chapter 2 discusses the water physics theory behind the signals observed by the non-intrusive water utility monitoring system. Chapter 3 discusses the design of the sensor system as well as the design of the experimental laboratory setup created to test the system. Chapter 4 presents the data collected from experiments in the laboratory and in the field with the non-intrusive water utility monitor and the analysis performed on the data. Chapter 5 documents the development of the free-space EQS sensor, and presents preliminary data gathered with the sensor. Chapter 6 concludes the thesis, detailing further work to be done on the projects presented and summarizing the contributions of the thesis work. Appendix A contains the MATLAB code used to process data collected by the non-intrusive water utility monitor. Appendix B contains schematics of the free-space EQS sensor.

Chapter 2

Water Physics

The non-intrusive water utility monitoring system observes information about the loads operating on a water pipe network through single-point vibration sensing. This requires an explanation of the physical phenomena that create local pipe vibrations from the flow of water delivered through the network to various loads. This chapter explains the mechanism for lateral vibration in a pipe segment, which produces high-energy peaks at discrete vibration frequency bands characteristic of the particular pipe segment and mounting. This chapter also presents theories about how the magnitudes of peak frequencies in a pipe segment are shaped by properties of the load(s) on the system in operation, why multiple loads in operation sometimes demonstrate approximate superposition, and how the vibration of a pipe segment is affected by water flow rate through the pipe network.

2.1 Lateral Vibration in a Pipe Segment

There are two pipe vibration types considered here. Axial vibrations in the fluid travel along the direction of a pipe segment, while beam-like lateral vibrations are motion normal to the axial direction of the pipe. Axial waves traveling through a fluid in a pipe segment couple to the lateral vibration modes of the pipe segment. Axial waves in the fluid interact with the inner pipe wall through various mechanisms such as gravity sag, in horizontal pipes, imperfections in the shape or thickness of the pipe wall, and material property variations [4]. Even pipes mounted vertically to eliminate gravity sag demonstrate coupling between axial waves and lateral vibration from pipe imperfections alone [5]. These various forces against the pipe wall cause an excitation of the lateral pipe vibration modes.

This lateral movement excites the the frequency modes of the pipe described by classical beam theory. The beam-like vibration modes amplify this motion, and the vibration data will display a distinctive set of frequency peaks for any particular pipe based on physical characteristics of the pipe, such as its dimensions and Young's modulus elasticity constant, as well as the mounting of the pipe ends (fixed-fixed, pinned-pinned, etc.).

Because the peak vibration modes can be predicted by analysis of the pipe as a beam, the same peaks observed as a result of water flow through the pipe should also be observed as a result of striking a blow to the pipe full of stationary water, so long as the mounting is the same.

The vibration mode peak locations can shift based on differences in the flow rate of the water [6], however they only shift by very small fractions of Hz at low flow rates, like the flow rate expected in a house or typical building. For most residential and commercial applications this effect is minimal and can be safely ignored.

Classical beam theory is well-developed, so the full analysis will not be reproduced here. For analysis of the experiments described in Chapter 4, it is necessary to derive the relative locations of nodes of the first several harmonic beam-like pipe modes along the length of a given pipe segment mounted with fixed-fixed ends [7]. Standard household and commercial pipe mounting is not classical fixed-fixed mounting, but the fixed-fixed classical model is used here to demonstrate that the calculation of beam-like mode nodes given the mounting conditions is possible.

The general governing equation of small lateral motions of a pipe carrying fluid is:

$$E^v I \frac{\partial^5 y}{\partial t \partial x^4} + EI \frac{\partial^4 y}{\partial x^4} + \left[\rho A_i U^2 + \overline{p} A_p (1 - 2v\delta) - \overline{T} - \left\{ (\rho A_i + m) g - \rho A_i \frac{\partial U}{\partial t} \right\} (L - x) \right] \frac{\partial^2 y}{\partial x^2}$$

$$+ 2\rho A_i U \frac{\partial^2 y}{\partial t \partial x} + (\rho A_i + m) g \frac{\partial y}{\partial x} + c \frac{\partial y}{\partial t} + (\rho A_i + m) \frac{\partial^2 y}{\partial t^2} = 0$$

The pipe properties involved in that equation are:

Length	L
Fluid Density	ρ
Internal Cross-Sectional Area	A_i
Pipe Wall Area	A_p
Tension in Pipe	$\overline{T}$
Pipe Area Moment of Inertia	I
Pipe Mass Per Length	m
Pipe Wall Young's Modulus	E
Pipe Wall Poisson's Ratio	v
Pressure of Water	$\overline{p}$
Flow Velocity	U
Viscoelastic Dissipation in Pipe	E^v
External Fluid Dissipation	c
Gravity	g

Table 2.1: Pipe Properties

It is necessary to make several assumptions about horizontal sections of household plumbing. It is reasonable to assume that the pressure in domestic water service is sufficiently small and that pipe supports in a home are sufficiently weak such that pressure-induced lengthwise tension from the expansion of the pipe wall cross-section under water pressure is negligible. It is also reasonable to assume that there is constant flow velocity in household plumbing. Finally, it can be assumed that the dissipation effects of the pipe wall material and the surrounding air are negligible as well. Under these assumptions the governing equation reduces to:

$$EI\frac{\partial^4 y}{\partial x^4} + \left[\rho A_i U^2 - (\rho A_i + m)g(L-x)\right]\frac{\partial^2 y}{\partial x^2}$$
$$+2\rho A_i U\frac{\partial^2 y}{\partial t \partial x} + (\rho A_i + m)g\frac{\partial y}{\partial x} + (\rho A_i + m)\frac{\partial^2 y}{\partial t^2} = 0$$

The previous equation can be non-dimensionalized with the following variables:

$$\xi = x/L \qquad \eta = y/L \qquad \beta = \frac{\rho A_i}{\rho A_i + m} \qquad \gamma = \frac{\rho A_i + m}{EI}L^3 g$$

$$u = \sqrt{\frac{\rho A_i}{EI}}UL \qquad \tau = \sqrt{\frac{EI}{m + \rho A_i}}\frac{t}{L^2}$$

Plugging in the above parameters into the governing equation, the equation becomes:

$$\frac{\partial^2 \eta}{\partial \tau^2} + 2\sqrt{\beta}u\frac{\partial^2 \eta}{\partial \tau \partial \xi} + \gamma\frac{\partial \eta}{\partial \xi} + [u^2 - \gamma(1-\xi)]\frac{\partial^2 \eta}{\partial \xi^2} + \frac{\partial^4 \eta}{\partial \xi^4} = 0$$

A Galerkin method can be used to find the modal frequencies. It can be assumed that the solutions can be represented by the following Galerkin expansion (where N is large enough for a good approximation):

$$\eta(\xi,\tau) = \sum_{r=1}^{N} \phi_r(\xi) q_r(\tau)$$

The functions $\phi_r(\xi)$ are the the dimensionless eigenfunctions of an Euler-Bernoulli beam that has the same boundary conditions as the pipe. There exist four classical boundary conditions: free, guided, pinned, and fixed. Calculations will be done with fixed-fixed boundary conditions, since those are closest to the boundary conditions of the experimental pipe setup used in Chapter 4. The form of the eigenfunctions is:

$$\phi_r(\xi) = C_1 \sin(\lambda_r \xi) + C_2 \cos(\lambda_r \xi) + C_3 \sinh(\lambda_r \xi) + C_4 \cosh(\lambda_r \xi)$$

The fixed-fixed boundary condition gives $\eta = \frac{\partial \eta}{\partial \xi}$ at $\xi = 0$ and $\xi = 1$. Our boundary conditions give:

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ \sin(\lambda) & \cos(\lambda) & \sinh(\lambda) & \cosh(\lambda) \\ \cos(\lambda) & -\sin(\lambda) & \cosh(\lambda) & \sinh(\lambda) \end{bmatrix} \quad \text{and} \quad C = \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{bmatrix}$$

For an arbitrary constant vector, the roots of the matrix equation $AC = 0$ are found via $\det(A)$ gives $1 - \cos(\lambda)\cosh(\lambda) = 0$ with roots:

$$\begin{aligned} \lambda_1 &= 4.730 \\ \lambda_2 &= 7.85475 \\ \lambda_3 &= 10.9955 \\ \lambda_4 &= 14.13716 \\ \lambda_5 &= 17.2787 \end{aligned}$$

In addition, we have that $C_1 = -C_3$, $C_2 = -C_4$, and $C_1 = C_2 \frac{\sin(\lambda)+\sinh(\lambda)}{\cos(\lambda)-\cosh(\lambda)}$. This leaves one arbitrary constant C_2 which we will set to 1 for now. Assume r is a positive integer. We get:

$$\phi_r(\xi) = \left[\frac{\sin(\lambda_r) + \sinh(\lambda_r)}{\cos(\lambda_r) - \cosh(\lambda_r)}(\sin(\lambda_r\xi) - \sinh(\lambda_r\xi)) + \cos(\lambda_r\xi) - \cosh(\lambda_r\xi)\right]$$

The first 5 clamped-clamped mode shapes are shown in Figure 2.1.

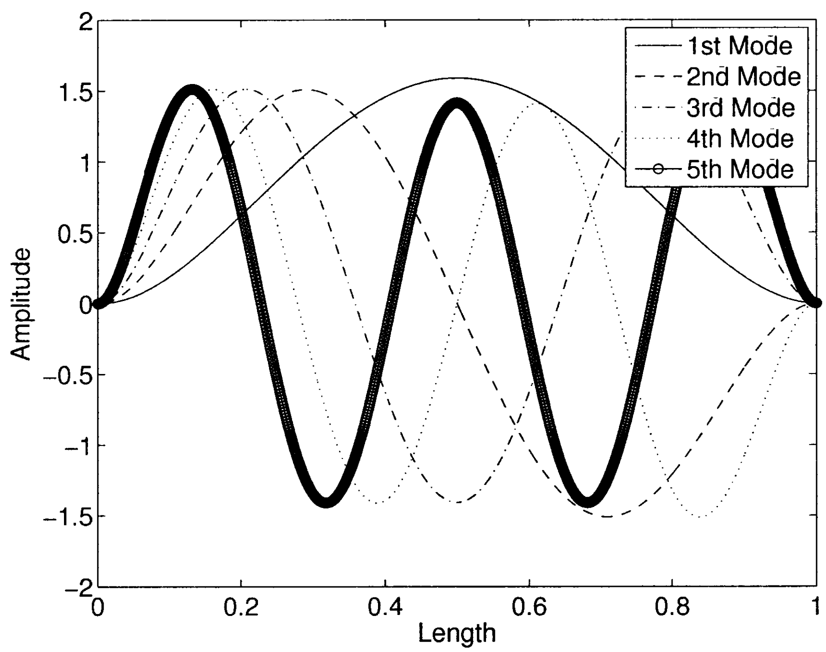


Figure 2.1: Locations of Nodes of Fixed-Fixed Harmonic Frequencies

2.2 Load Pressure Frequency Peak Coloring

The frequencies of the beam-like modes are fixed for all loads on the pipe, but the amplitudes depend on the particular load in operation. Different water loads such as garden sprayers with varied aperture sizes will demand different levels of water pressure and flow rate. Different water loads also empirically produce spray sounds of different audible frequencies. A bathroom shower head's spray sound is distinguishable from the spray sound of a kitchen sink faucet or a garden hose tap in the backyard.

These different sound, pressure, and flow rate characteristics of water loads adjust the relative beam-like mode peak magnitudes into reproducible and characteristic load frequency "fingerprints". These "fingerprints" can be used to distinguish the operation of different types of loads from one another.

2.3 Superposition of Loads

If characteristics of particular loads determine the relative ratio of the magnitudes of the beam-like modes produced by water flow, then the interaction of two different loads in operation at the same time can be explored. Based on experiments presented in Chapter 4, it appears that the characteristic peak ratios of two different loads in operation follow some rough, if nonlinear, superposition in shape. The exact reasons for this are unknown. Further analysis of this problem is not presented in this thesis, but is suggested in Chapter 6 as further work.

2.4 Flow Rate

Previous work [6] suggests that the flow rate through a pipe segment can be extracted from vibration information. The turbulence from very high flow rates shift can change the power of the noise floor in vibration measurements. However detection of the variations described would require very fine measurements at the relatively low flow rates found in typical residential and commercial plumbing.

There may be correlation between the vibration frequency floor level and rate of water flow, but it is difficult to measure this experimentally because of the practical difficulties in controlling flow rate and water pressure independently in our laboratory setup.

Chapter 3

Water Experimental Setup Design and Data Analysis

This chapter details the design of the vibration measurement setup for the non-intrusive water utility monitoring system and the signal processing techniques applied to the vibration data collected with the setup. The chapter also describes the construction of a laboratory setup used for the experiments presented in Chapter 4.

3.1 Vibration Measurement Setup

The non-intrusive water utility monitor initially used a microphone to take vibration data, but use of the microphone required excessive precaution of the contamination risk of ambient sound. An accelerometer was later used, which solved the ambient noise problem.

The accelerometer employed was reliably sensitive to frequencies up to 1 kHz. The accelerometer was seated on a small prototyping board (see Figure 3.1). That prototyping board was mounted on a custom harness to rigidly attach it to points along the pipe segment to be monitored (see Figure 3.2).

Output from the accelerometer was connected to a breadboard with a low-pass filter and several level shifting and gain stages (see Figure 3.3). The output signal from that breadboard was connected to an eighth-inch audio jack. A laptop running the audio recording software program Audacity was connected via USB interface to an external audio card with input and output ports. The external audio card device was a GWC USB 5.1 Channel Audio Adapter AA1500 5.1 Channels USB Interface Sound Card, purchased from www.newegg.com (see Figure 3.4). The eighth-inch audio jack from the breadboard was plugged into the "Line" input port of the external audio card connected to the laptop computer. Data from the external audio card was recorded on the laptop in Audacity and exported as ".WAV" format audio files for further processing (see Figure 3.5 and Figure 3.6). All settings on the Audacity recording software were set to their defaults, except the sampling rate, which was set to 48000 Hz for all data collection.

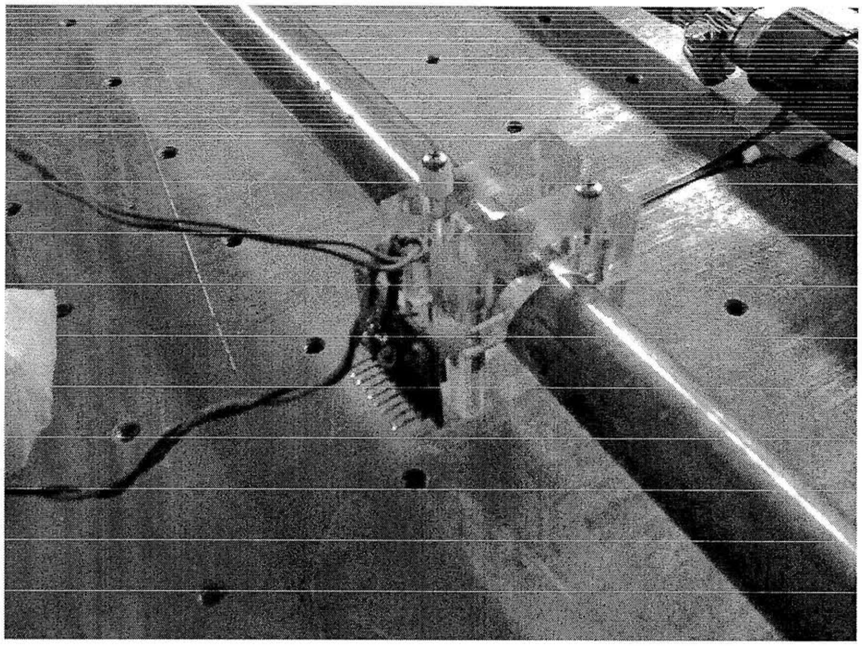


Figure 3.1: Accelerometer Protoboard

Field test vibration data was collected with a microphone (see Figure 3.7). For early field experiments, the microphone was seated in a block of foam, to dampen ambient sound (see Figure 3.8). The microphone and foam assembly was attached to the pipe segment to be monitored using cable ties (see Figure 3.9). For later field experiments, the foam block mounting was not used, and instead the microphone was rigidly attached to the pipe using a hose clamp to improve pipe vibration sensitivity.

Output from the microphone was connected directly into the external audio card's "Mic" input port via the eighth-inch audio jack on the microphone. Just as with the accelerometer, the external audio card was connected by a USB interface to a laptop computer running the Audacity audio recording software. The data was exported as .WAV audio files (see Figure 3.10).

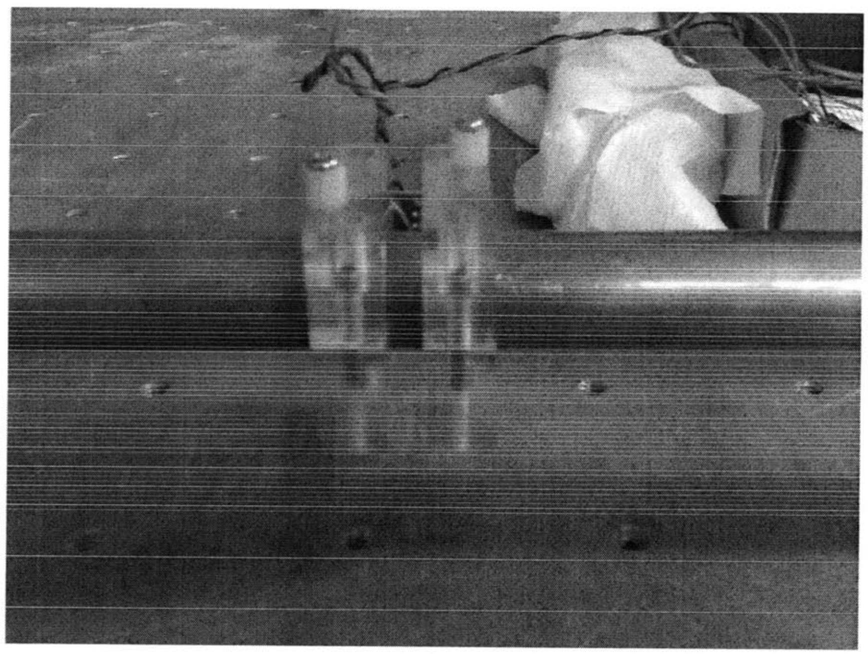


Figure 3.2: Accelerometer Protoboard Mounted on Custom Harness

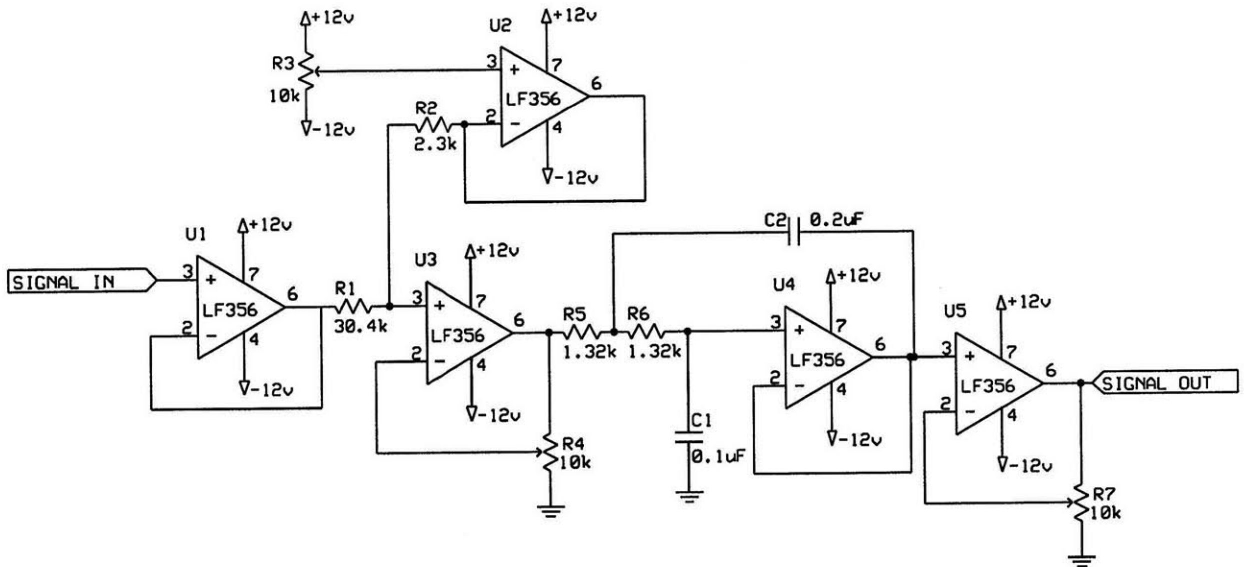


Figure3.3: Schematic of Breadboard Signal Processing Stages

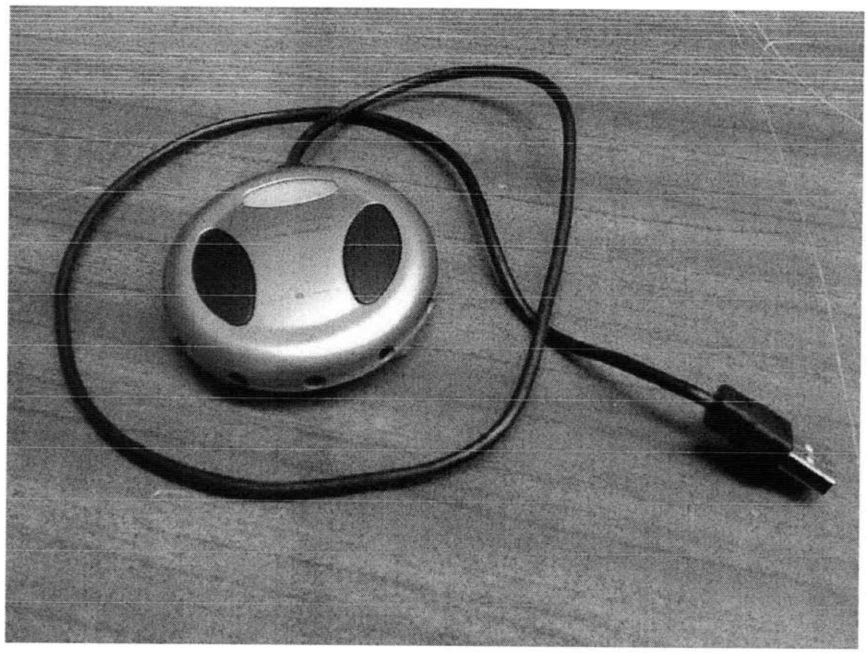


Figure 3.4: Photo of USB External Audio Card

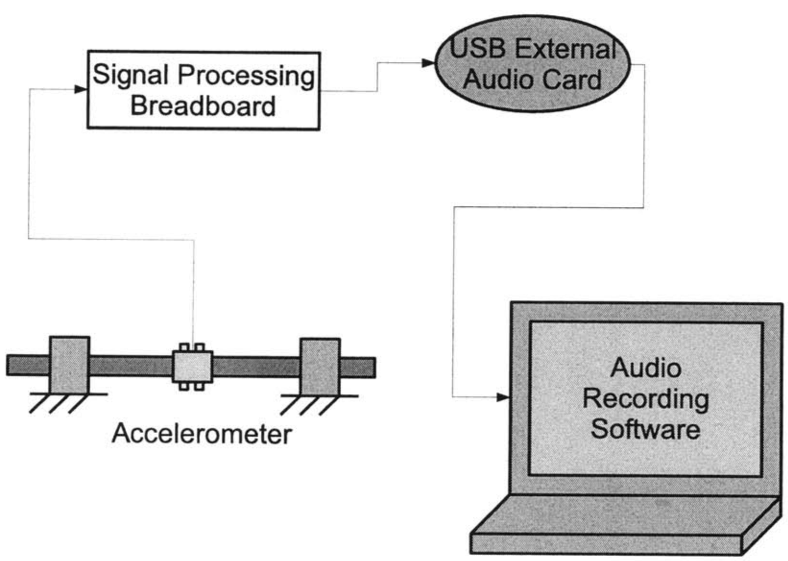


Figure 3.5: Block Diagram of Data Flow from Accelerometer to Laptop

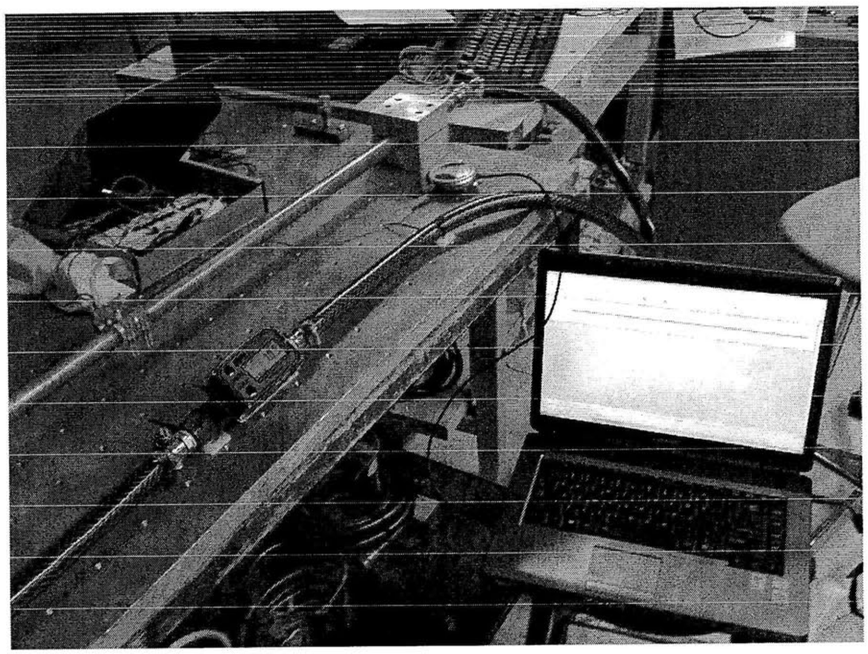


Figure 3.6: Photo of Data Flow from Accelerometer to Laptop

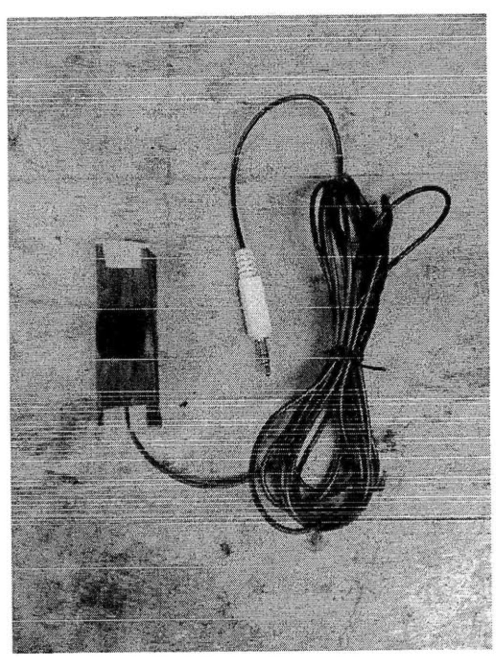


Figure 3.7: Microphone

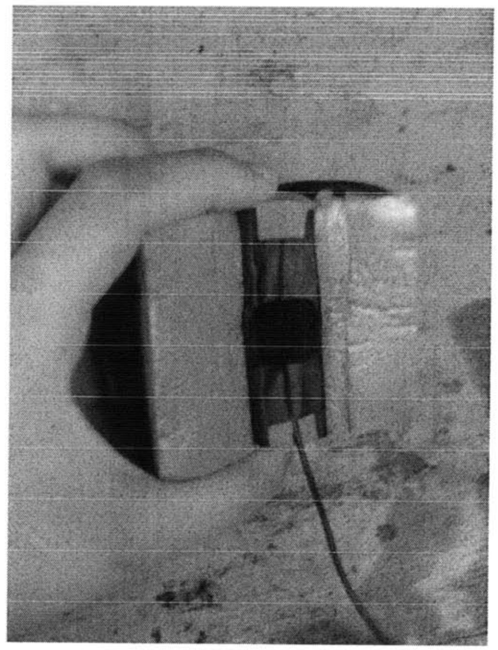


Figure 3.8: Microphone and Foam Block Assembly

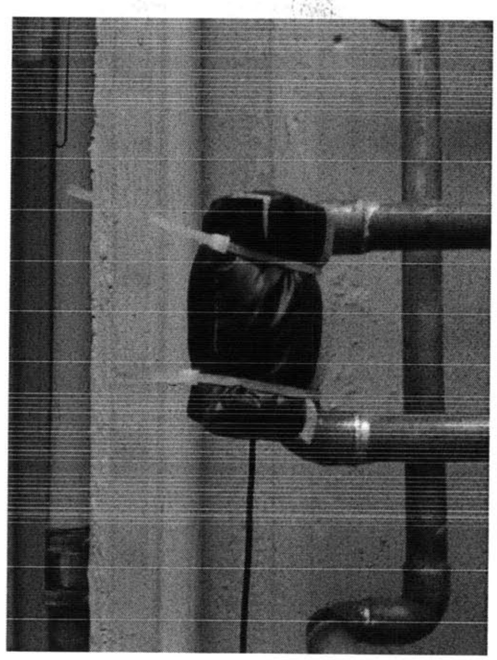


Figure 3.9: Microphone and Foam Block Assembly Mounted on Pipe

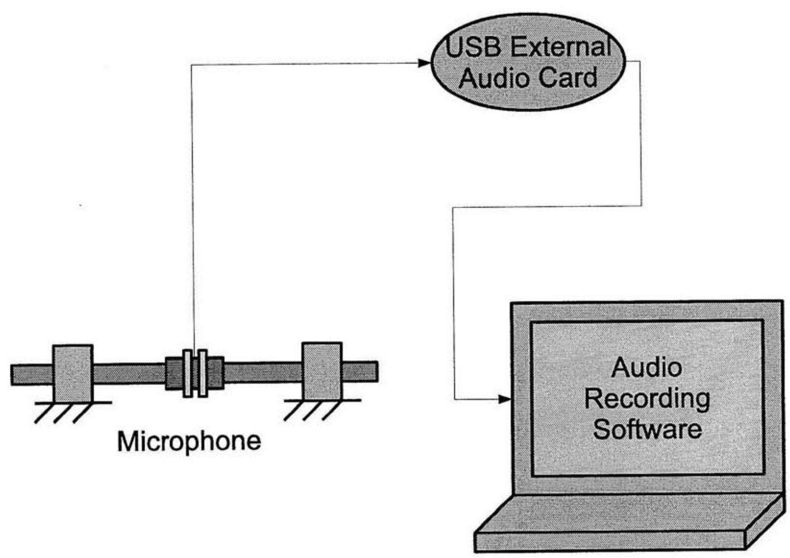


Figure 3.10: Block Diagram of Data Flow from Microphone to Laptop

3.2 Vibration Data Signal Processing

The .WAV data files taken from the vibration sensor in the non-intrusive water utility monitoring system were processed in MATLAB (see Appendix A) to create two types of graphs: Energy Spectral Density (ESD) versus Frequency and Spectral Envelope versus Time.

Energy Spectral Density versus Frequency graphs display the total energy at each vibration frequency summed over the total time of the data sample. This information identifies which vibration frequency bands carried the most energy over all time, and how much energy they carried relative to one another. This relates to both finding the vibration modes of a local pipe segment and for finding frequencies whose Spectral Envelope characteristics could be used identify different loads on the network. To find the ESD for a given data file, first a sliding time window FFT was taken. The magnitudes of all of the sliding time window transforms are then added together, giving the Energy Spectral Density for all time (see Figure 3.11).

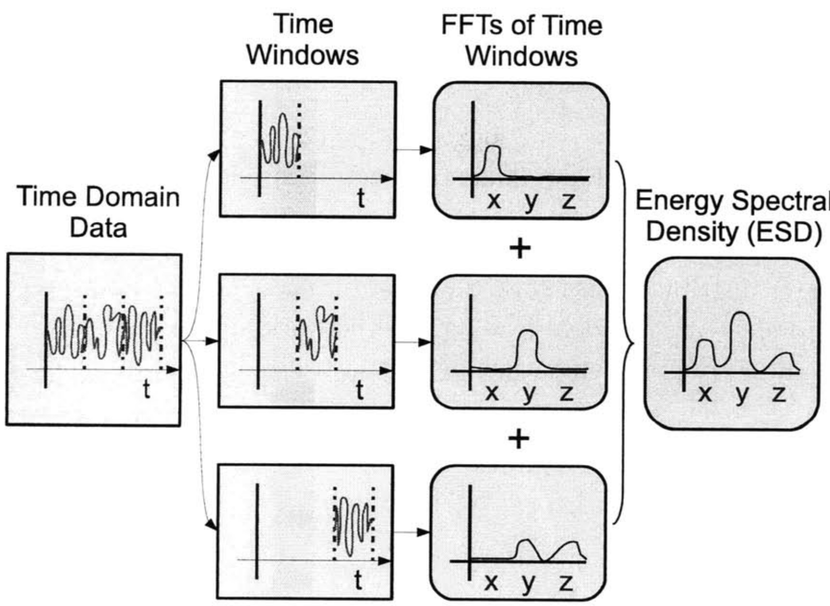


Figure 3.11: Flow Chart of ESD Calculation

Using the peaks of the Energy Spectral Density of loads as a guide, high magnitude peaks were selected. The development of the magnitudes of those peak frequencies over time create distinctive load signatures, or Spectral Envelopes. The Spectral Envelopes were assembled from singling out the frequency under observation from each of the time windowed FFT's taken above. These points are then time indexed to construct the pattern that the given frequency follows at different time values (see Figure 3.12). This information identifies changes in the energy at a given frequency which correspond to changes in the water flow through the pipe, such as the turn-on and turn-off transients of loads on the network.

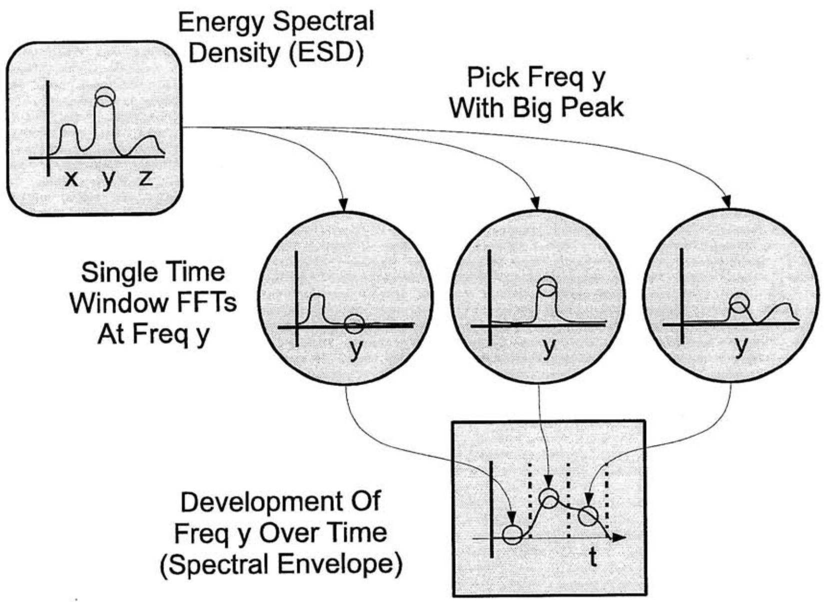


Figure 3.12: Flow Chart of Spectral Envelope Calculation

When applied to vibration data taken from a single point measurement on one pipe segment in a water distribution network, these two processing techniques provided sufficient information to identify individual loads on the network and to distinguish them from one another, even in overlapping operation, as demonstrated in the experiments in Chapter 4.

3.3 Experimental Setup

An experimental pipe fixture laboratory setup was constructed for performing experiments to verify the supporting water physics analysis for the non-intrusive water utility monitoring system. The main part of the setup consisted of a length of copper tube held in place by two rigid fixtures attached to an optical bench (see Figure 3.13).

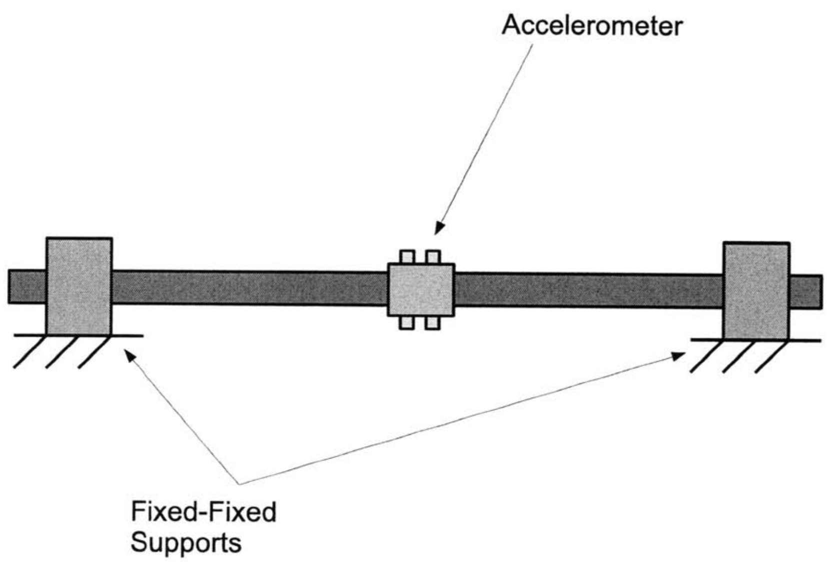


Figure 3.13: Diagram of Pipe with Supports

The copper tube segment was connected to a water source (a sink) and water loads in various ways for different experiments (see Figure 3.14 and Figure 3.15). One end of the copper tube segment terminated in a junction with attachment points for some different loads: a garden hose tap, a vegetable sprayer, and a shower head (see Figure 3.16 and Figure 3.17).

Two sensors besides the accelerometer were employed for the laboratory experiments: a water flow meter, GPI TM050-N/TM050-N-P, and a water pressure sensor, Measurement Specialties Inc. MSP-300-100-P-4-N-1. The water flow meter was an inline flow meter sensitive from 1 to 10 Gallons Per Minute (GPM), with a digital display output (see Figure 3.18). The pressure sensor was sensitive up to 100 pounds per square inch (PSI), with an analog voltage output. The water flow meter data was recorded by observation. The pressure sensor data was recorded electronically, using a setup similar to the accelerometer data collection. The pressure sensor output was connected to a breadboard with the same

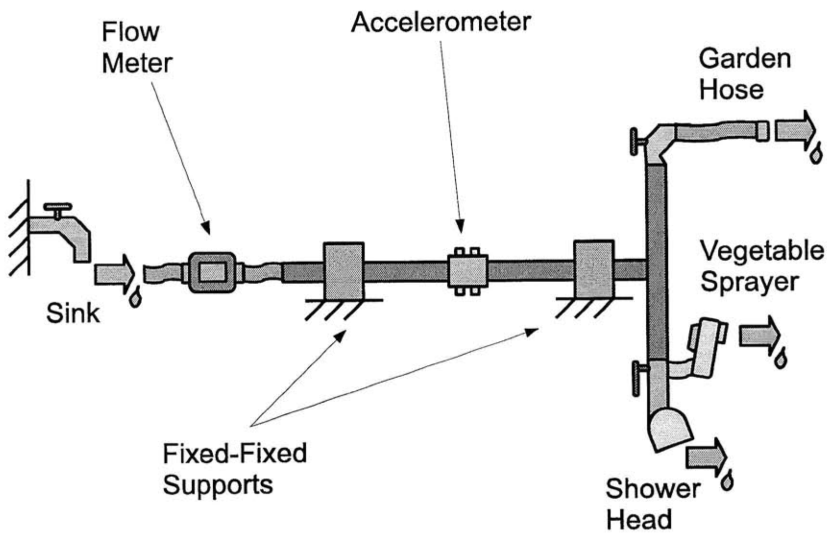


Figure 3.14: Diagram of Whole Setup

low-pass filter and several level shifting and gain stages as the accelerometer breadboard (see Figure 3.3). The output from the breadboard was again a cable terminating in an eighth-inch audio jack. The audio jack was plugged into the "Line" input port of the same external USB audio card. The audio card was connected to a laptop computer, and the data was recorded in the Audacity audio recording program (see Figure 3.19).

The pressure sensor data was exported as ".WAV" files and subjected to the same signal processing techniques in MATLAB as the vibration data.

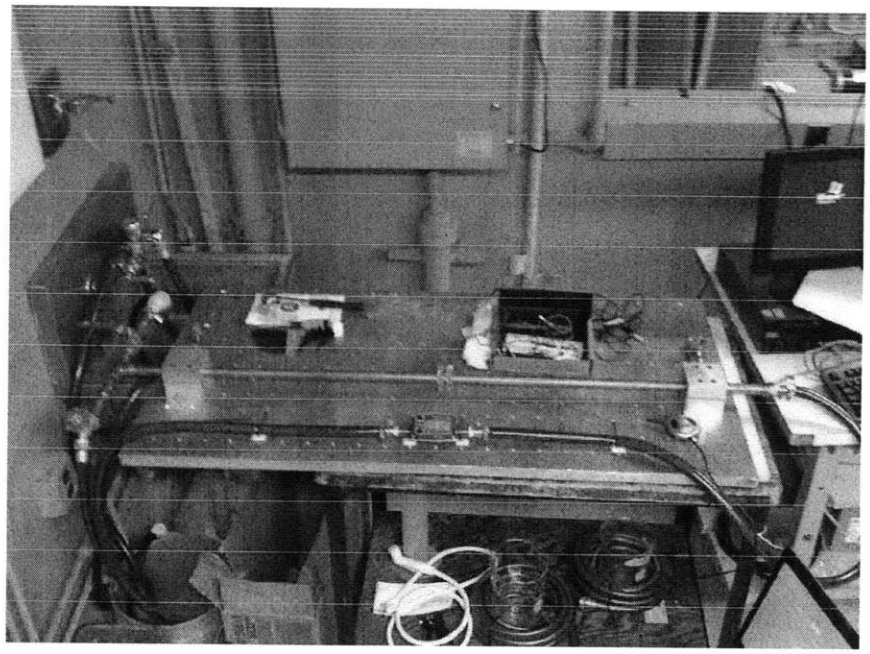


Figure 3.15: Photo of Whole Setup

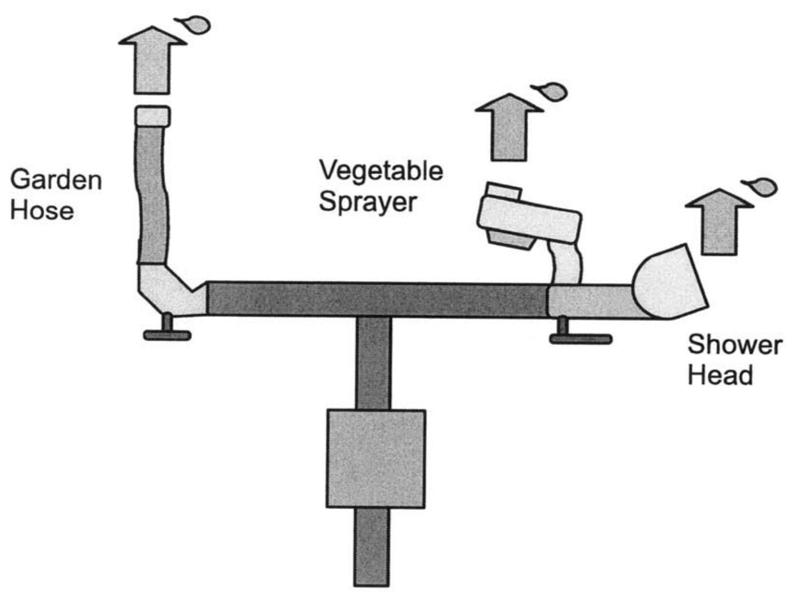


Figure 3.16: Diagram of Junction Loads

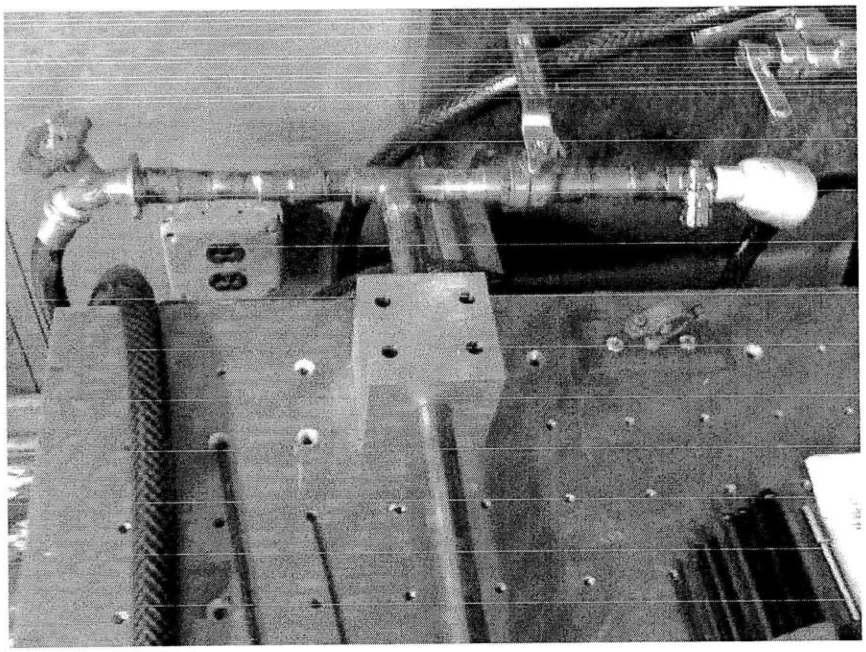


Figure 3.17: Photo of Junction Loads

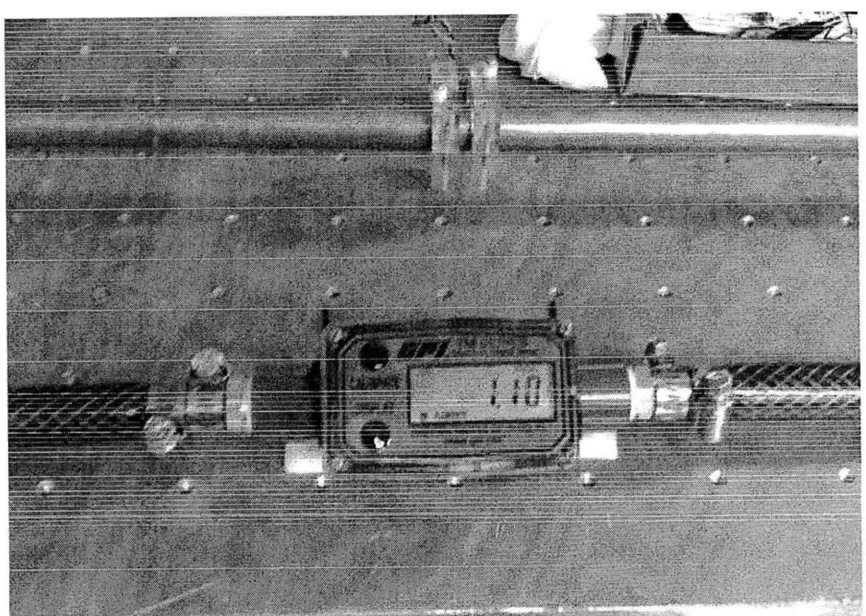


Figure 3.18: Photo of Flow Meter

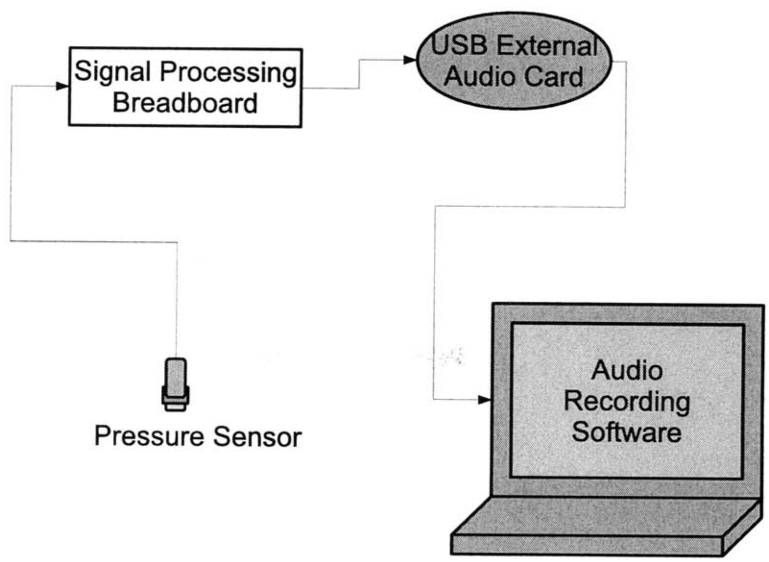


Figure 3.19: Block Diagram of Data Flow from Pressure Sensor to Laptop

Chapter 4

Water Experiments

Experiments were performed both on a laboratory pipe setup and on the plumbing network of a single-family home. This chapter presents the data collected from both sources, and the analysis performed on the data.

4.1 Lateral Vibration in a Pipe Segment Experiments

Lateral vibration experiments were conducted to find the resonant beam-like modes for an example pipe segment.

The pipe segment setup was configured with two fixed supports simulating the classical fixed-fixed boundary conditions spaced 39.125 in apart. The pipe segment was filled with water and then struck with a hammer while vibration data was recorded by an accelerometer in the middle of the pipe segment (see Figure 4.1). The ESD plot for the accelerometer data demonstrated three main beam-like modal peaks at 72 Hz, 384 Hz, and 924 Hz (see Figure 4.2). These correspond to the 1st, 3rd, and 5th natural modes of the beam. The 2nd and 4th modes are not present because the accelerometer sat in the middle of the pipe segment and could only measure odd pipe harmonics.

To confirm that water flow through the pipe segment drawn by a water load excites the beam-like modes of a pipe, the same pipe segment was connected using garden hose segments to a water source on one end and an open garden hose spout on the other end. The accelerometer was left in the same location as the hammer strike test (see Figure 4.3). Vibration data was recorded while the water flowed. The ESD plot for the accelerometer data generated by water flow demonstrates the same three modal peaks as the accelerometer data generated by the previous hammer strike test, at 72 Hz, 384 Hz, and 924 Hz respectively (see Figure 4.4).

These experiments confirm the beam-like vibration mechanism behind the observed water-flow induced frequency peaks. The frequency peaks in the vibration generated by water flow through the pipe are at the same frequencies as the peaks that represent the beam-like

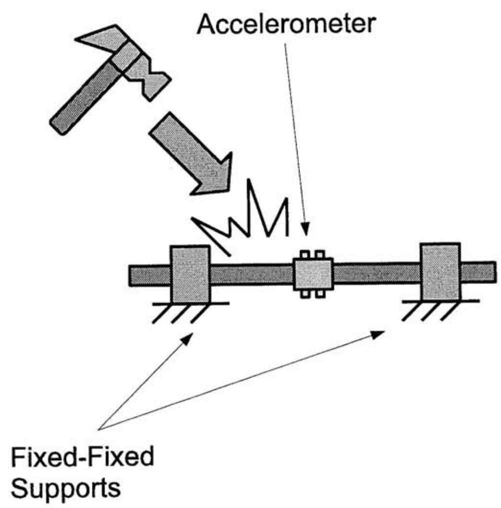


Figure 4.1: Diagram of Hammer Strike Experiment Setup

modes of the pipe. The peaks demonstrate different magnitude ratios but they are at the same frequencies.

To further demonstrate that the frequencies at which ESD peaks are observed are the beam-like modes of the pipe, experiments were performed in which the accelerometer was moved to different locations along the pipe to cancel out different beam-like mode harmonics. The beam-like harmonic frequencies of the water-filled pipe for fixed-fixed end conditions have nodes spaced at distances derived in Chapter 2 (see Figure 4.5). Again, the pipe segment was connected using garden hose segments to the sink on one end and an open garden hose spout on the other end. The accelerometer was placed at two different positions along the pipe. One position was one-half of the way along the pipe, and the other position was one-third of the way along the pipe.

Based on the harmonic node locations, when the accelerometer was one-half of the way along the pipe, only odd harmonic modes should be detected in the vibration. When water was run through the pipe, only odd harmonic modes (the 1st, 3rd, and 5th harmonics in this case) were visible in the ESD graph (see Figure 4.6).

When the accelerometer was one-third of the way along the pipe, only the 2nd, 3rd, and 4th harmonic modes should be detected in the vibration. When water was run through the pipe, only the 2nd, 3rd, and 4th modes were visible in the ESD graph (see Figure 4.7).

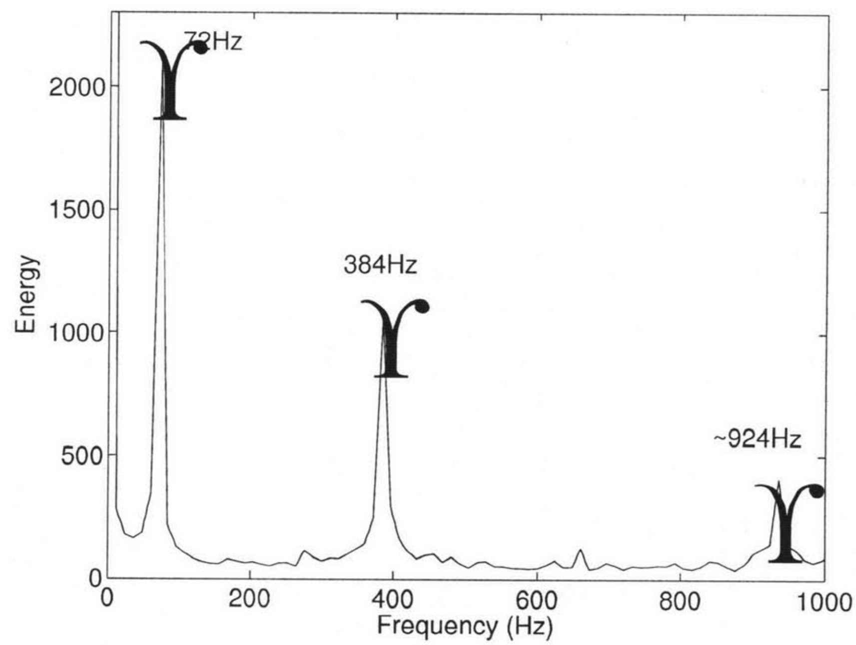


Figure 4.2: ESD for Hammer Experiment

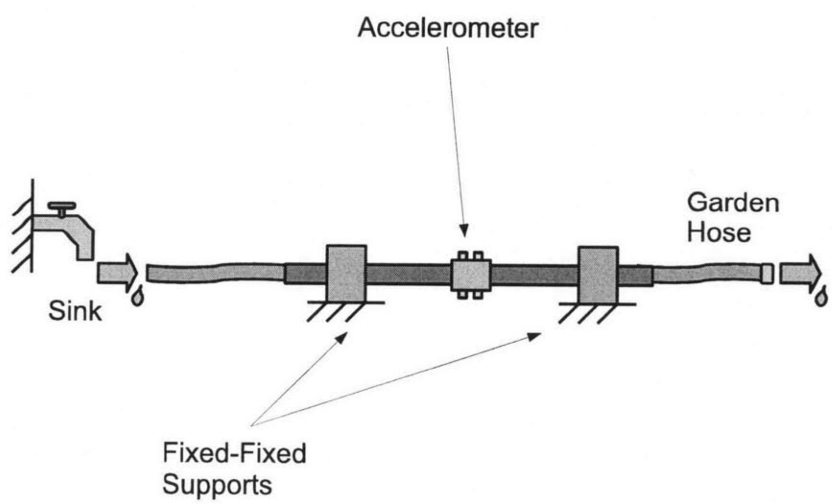


Figure 4.3: Diagram of Water Flow Experiment Setup

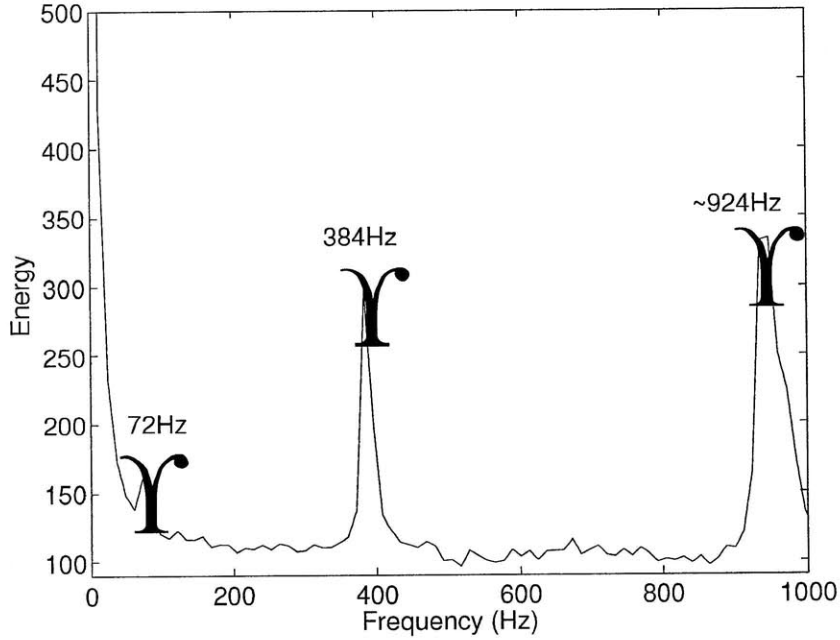


Figure 4.4: ESD for Water Flow Experiment

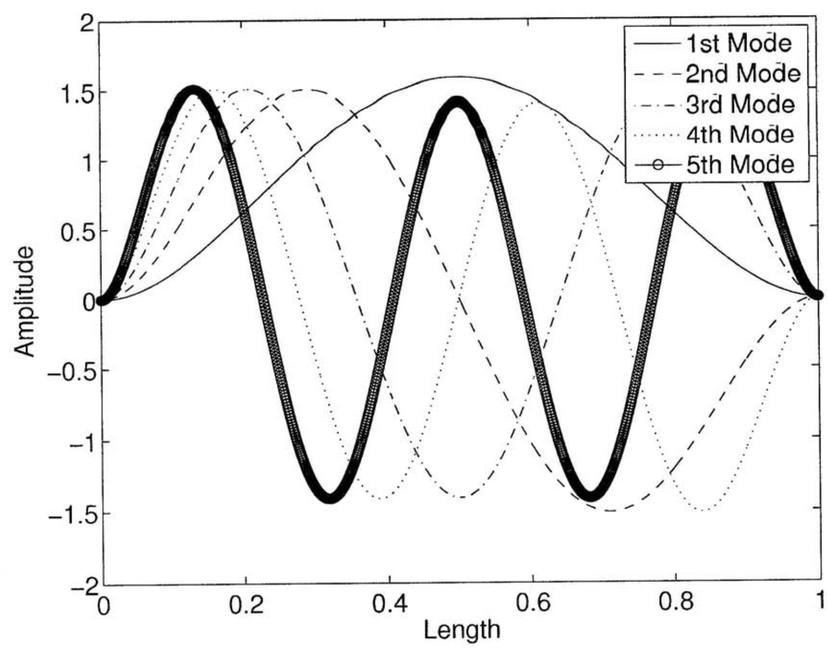


Figure 4.5: Locations of Nodes of Fixed-Fixed Harmonic Frequencies

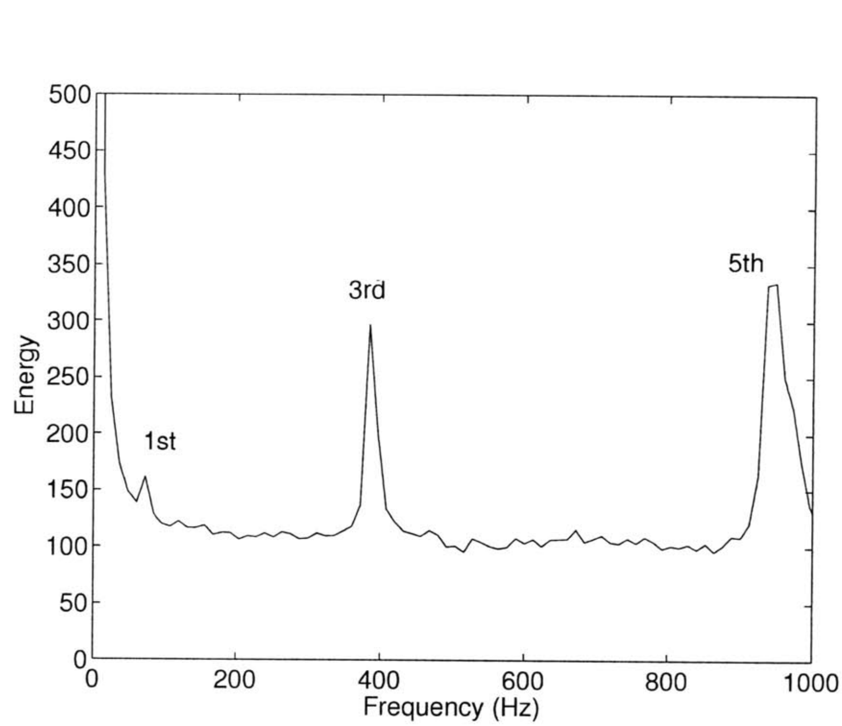


Figure 4.6: ESD of Water Flow with Accelerometer at One-Half of the Pipe Length

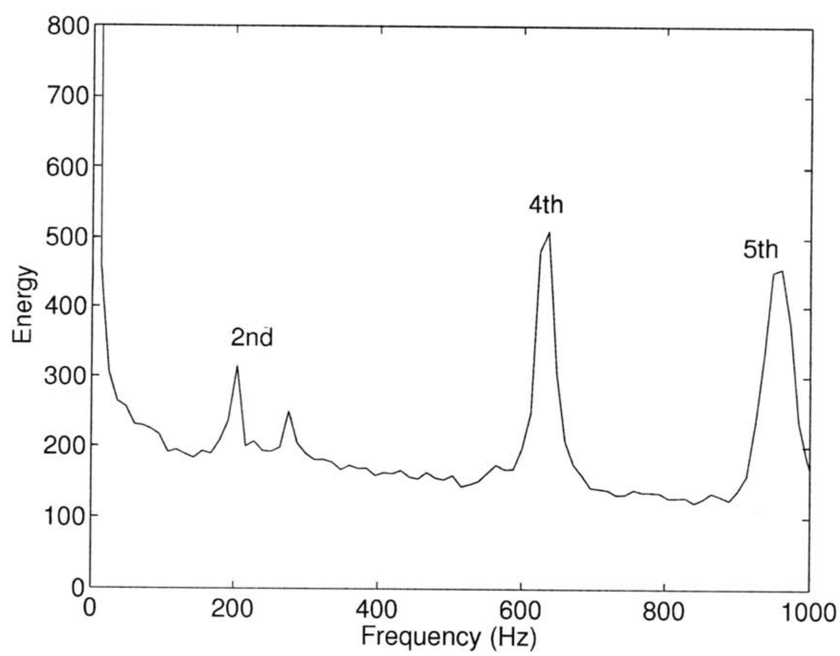


Figure 4.7: ESD of Water Flow with Accelerometer at One-Third of the Pipe Length

4.2 Load Pressure Frequency Coloring Experiments

To determine the effect of water pressure on the relative magnitudes of the beam-like frequency mode peaks, several experiments were performed. The first experiment was to characterize the background frequency content of water source used in the experiments: an institute utility sink. A pressure sensor was attached as a cap on the end of the sink spout. To minimize air bubbles, the sensor was attached while water was flowing, cutting off the flow as it was screwed in place. After the sensor cap was in place and exposed to the utility pressure by fully opening the sink spigot, background pressure data was recorded. This procedure was also repeated for two different lengths of garden hose attached to the sink, to see if the flexible hose wall significantly affected the background pressure variations (see Figure 4.8). The ESD plots of the data revealed that the frequency peaks were at the same locations and were of nearly identical magnitudes for all three cases (see Figure 4.9).

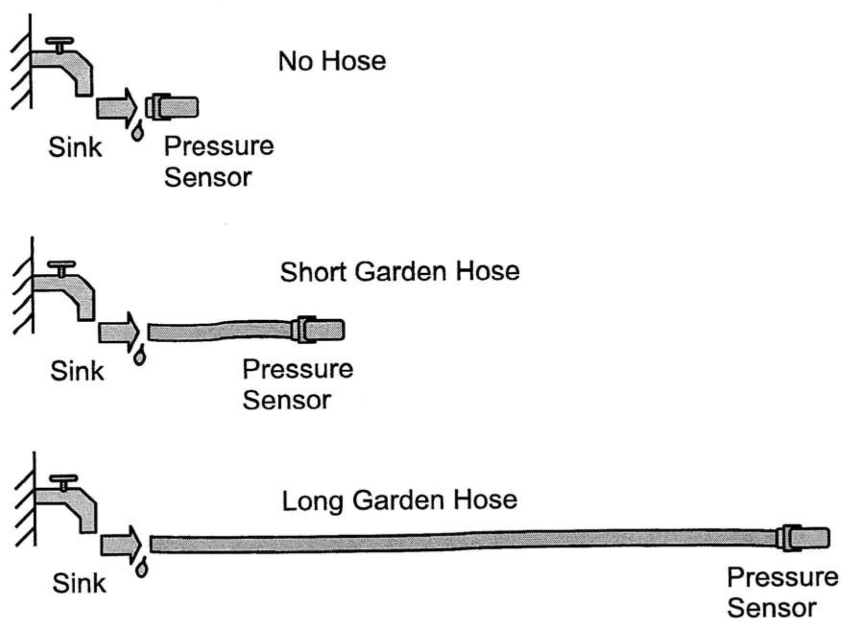


Figure 4.8: Diagram of Background Pressure Experiment

The second pressure experiment performed was to examine the frequency content of water flowing through an opening of different sizes. The opening used was a garden sprayer hose attachment that had a rotating wheel on the front to select different spraying aperture sizes (see Figure 4.10). The variable aperture garden sprayer was attached to the end of a short length of garden hose connected to the sink source (see Figure 4.11). The short length of hose was chosen for convenience, since the previous experiment's results (see Figure 4.9) indicated that hose length did not affect the pressure frequency content. Water was

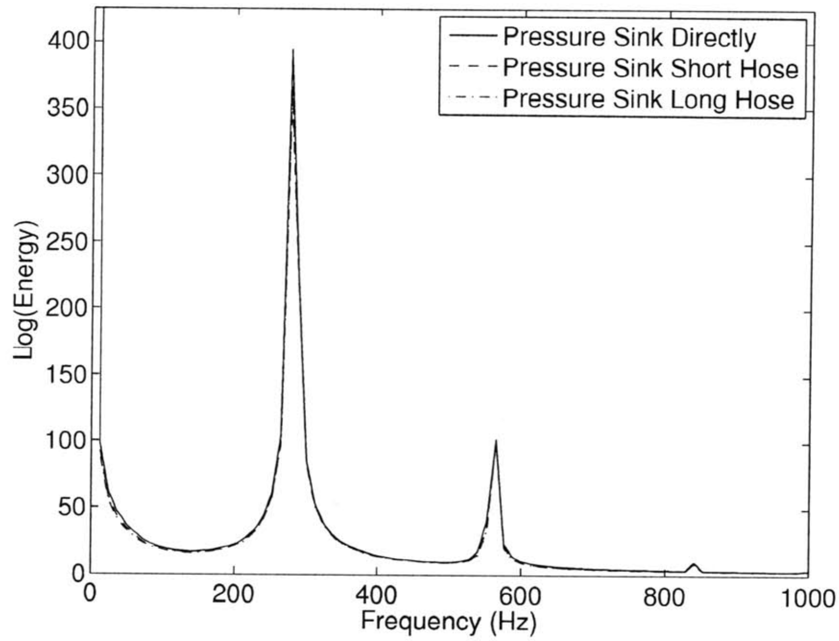


Figure 4.9: ESDs for Background Pressure Experiment

run through the four different aperture options. The ESDs of the pressure data showed a clear pattern at lower frequencies, but no clear pattern at higher frequencies (see Figure 4.12). The energy decreases with decreasing flow rate achieved by selecting small apertures, including no flow.

The third experiment was a variation on the second experiment, to collect both vibration and pressure data. In this experiment, the garden sprayer and pressure sensor T-junction were not directly connected to the sink with hose. Instead, the garden sprayer and pressure sensor T-junction connected to the fixed-fixed pipe segment with hose, and the pipe segment was connected to the sink with hose. The accelerometer was mounted on the middle of the pipe segment (see Figure 4.13). Water was run through the four different aperture sizes. The ESD of the pressure sensor data followed a similar trend to the previous experiment with garden hose only. There was a similar pattern in the lower frequencies, and no clear pattern in the higher frequencies (see Figure 4.14). The ESD of the accelerometer data displayed no clear trend, except around 900 Hz, where there was the most energy when the biggest aperture was used, decreasing gradually to the least energy when no aperture was used (see Figure 4.15).

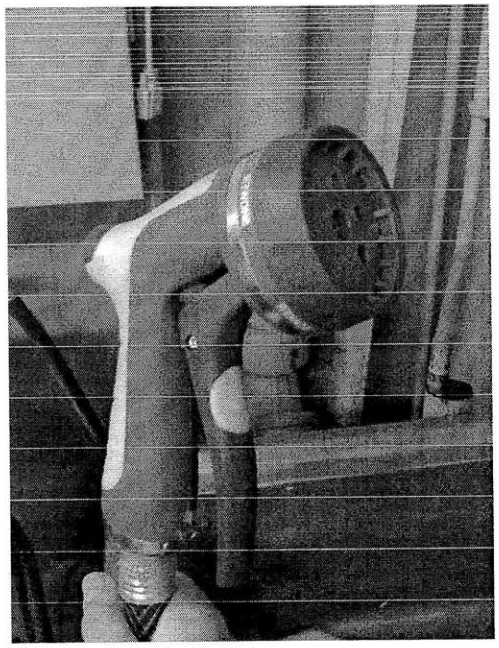


Figure 4.10: Photo of Variable Aperture Garden Sprayer

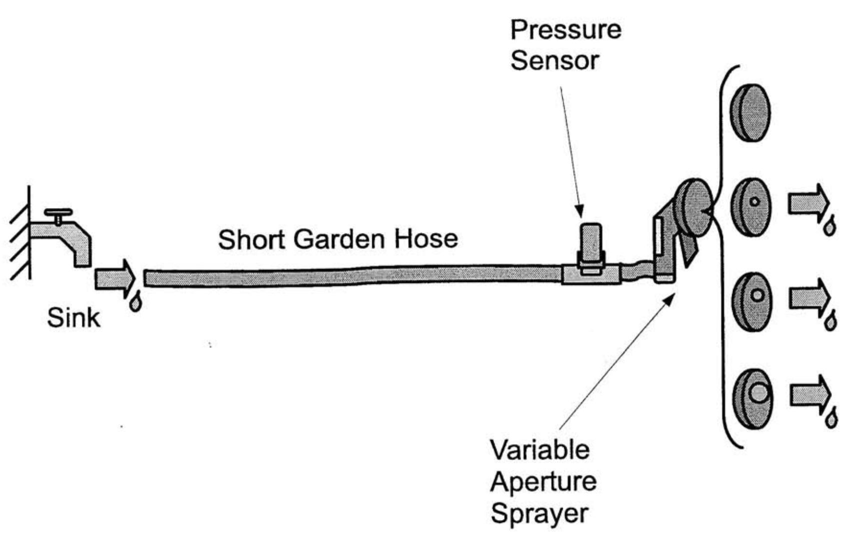


Figure 4.11: Diagram of Hose Background Pressure Experiment

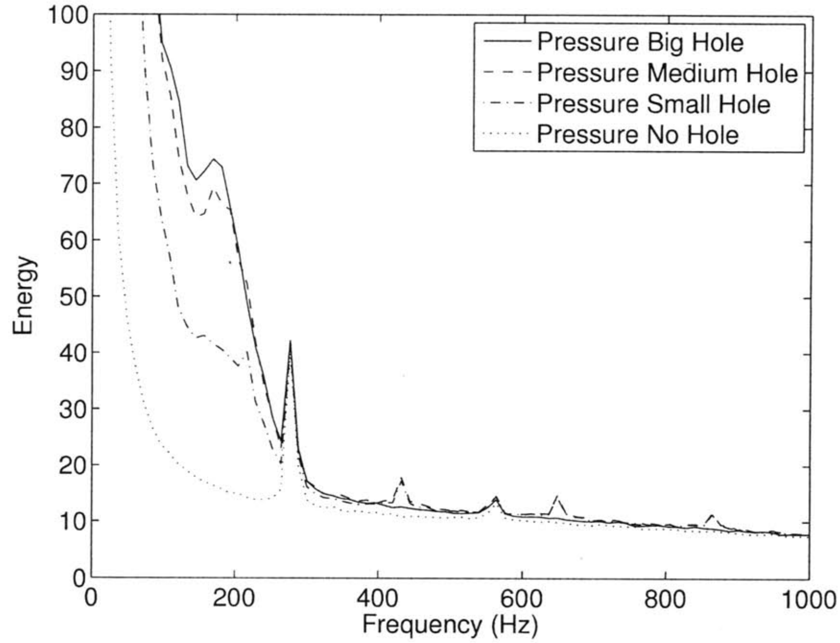


Figure 4.12: ESD for Hose Background Pressure Experiment

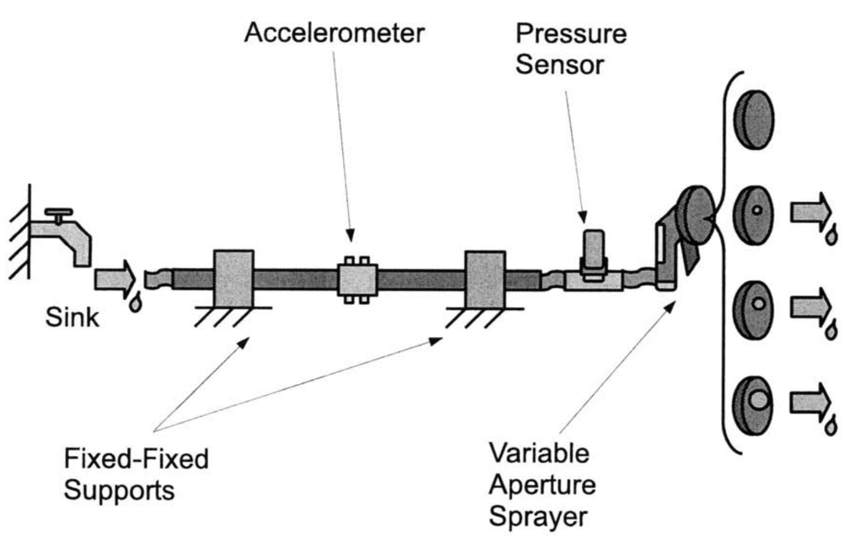


Figure 4.13: Diagram of Pipe Background Pressure Experiment

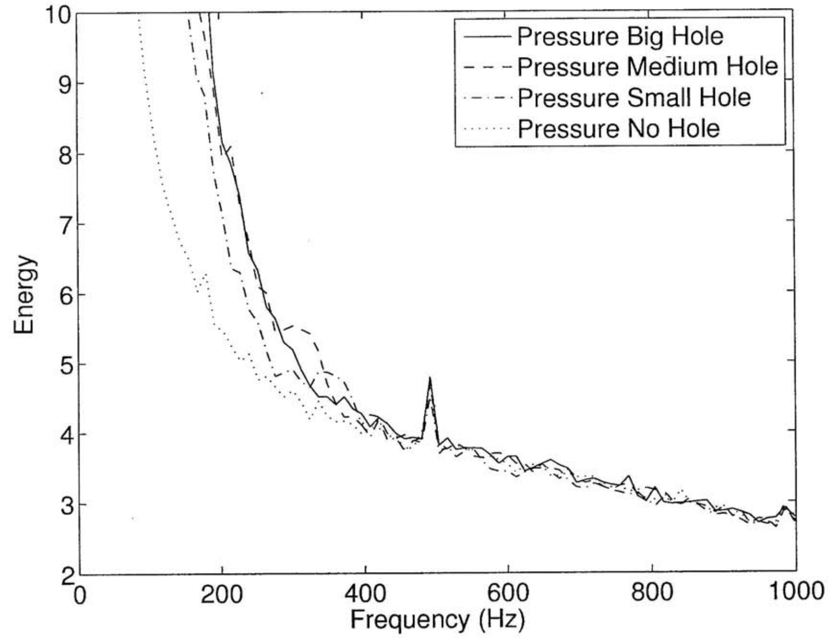


Figure 4.14: ESD for Pipe Background Pressure Experiment Pressure Sensor Data

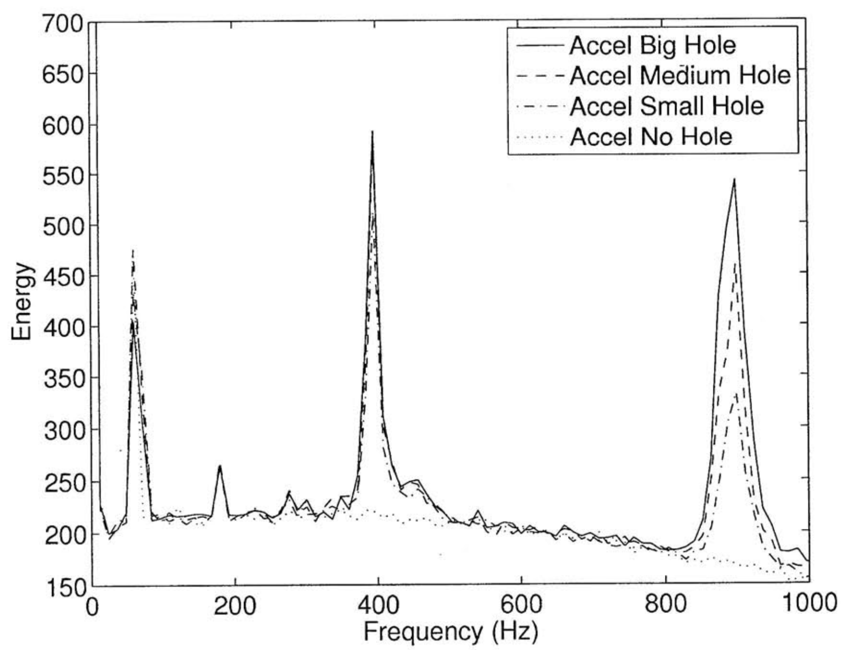


Figure 4.15: ESD for Pipe Background Pressure Experiment Accelerometer Data

4.3 Individual Load Transients Experiments

Experiments were performed to demonstrate that individual loads on the same water distribution network could be identified from a single non-intrusive vibration sensor mounted on a pipe near the source of water intake. To do these experiments, the sink water source was connected by hose to one end of the pipe segment mounted on fixed-fixed supports. The other end of the pipe segment terminated in a junction with three different loads attached to it: a garden hose, a vegetable sprayer, and a shower head (see Figure 4.16). These loads could each be operated individually. The accelerometer was mounted on the middle of the pipe segment.

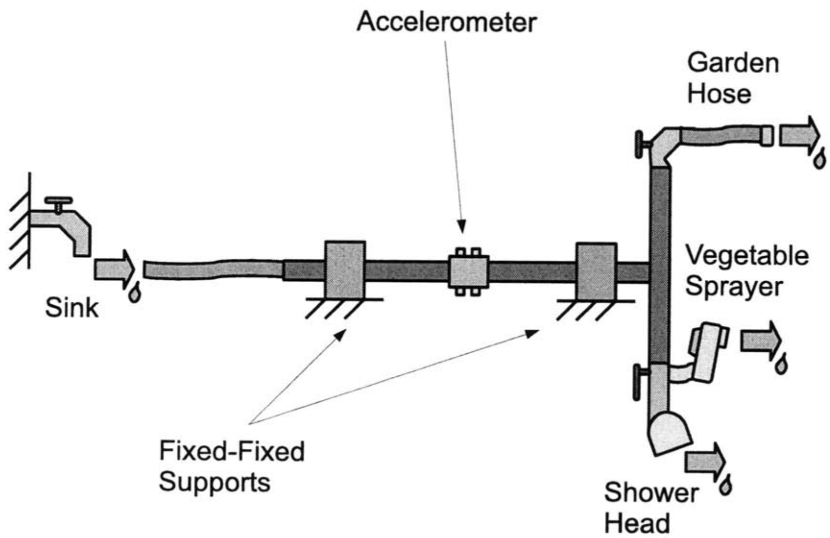


Figure 4.16: Diagram of General Individual Load Transient Experiment Setup

The garden hose load started at the junction of the pipe end with a spigot (see Figure 4.17). The garden hose screwed onto the spigot, and water to the load was turned on and off at the spigot.

The garden hose load was operated alone. The ESD of the accelerometer data (see Figure 4.18) has peaks at the three beam-like modes of the pipe segment found in the lateral vibration of a pipe segment experiments (see Figure 4.4): at 72 Hz, 384 Hz, and 924 Hz. The 72 Hz peak is the smallest in magnitude of the three. The 384 Hz peak is the largest, and the 924 Hz peak is in between.

The Spectral Envelopes were calculated (see Chapter 2) and plotted for the garden hose load

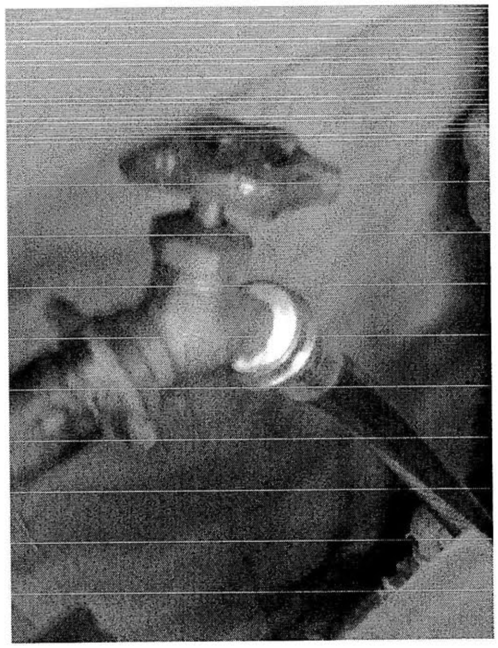


Figure 4.17: Photo of Garden Hose Load

at the three peak frequencies. The magnitudes of the three Spectral Envelopes maintain the same rank order as the corresponding peak magnitudes in the ESD plot (see Figure 4.18). The Spectral Envelope of the development of the 72 Hz component of the vibration over time is the smallest in magnitude (see Figure 4.19). The Spectral Envelope of the development of the 384 Hz component of the vibration over time is the largest in magnitude (see Figure 4.20), and the Spectral Envelope of the 924 Hz component is in between (see Figure 4.21). The shape of the Spectral Envelope of the load at all three frequencies is roughly rectangular, starting when the load turns on and ending when the load turns off.

The shower head load was attached to the junction of the pipe end (see Figure 4.22). The shower head was turned on and off by a ball valve with a handle.

The shower head load was operated alone. The ESD of the accelerometer data (see Figure 4.23) also has peaks at the three beam-like modes of the pipe segment found in the lateral vibration of a pipe segment experiments. The 72 Hz peak is the smallest in magnitude of the three. The 384 Hz peak is larger, and the 924 Hz peak is the largest. This magnitude ordering is different from that of the garden hose load.

The Spectral Envelopes were calculated and plotted for the shower head load at the three peak frequencies. The magnitudes of the three shower head Spectral Envelopes again preserve the rank order of the peak magnitudes in the ESD plot of the shower head (see Figure

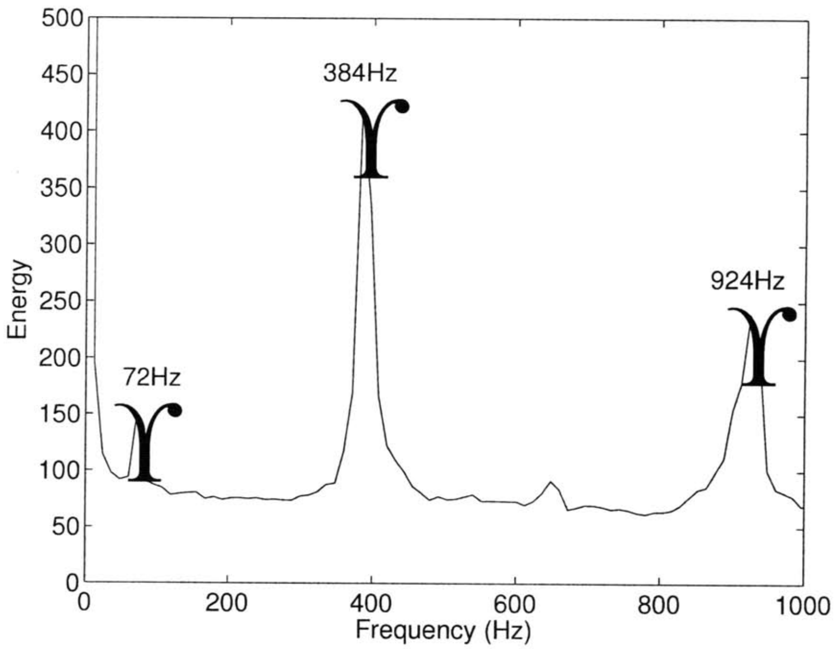


Figure 4.18: ESD for Garden Hose

4.23). The Spectral Envelope of the development of the 72 Hz component of the vibration over time is the smallest in magnitude (see Figure 4.24). The Spectral Envelope of the development of the 384 Hz component of the vibration over time is larger (see Figure 4.25), and the Spectral Envelope of the 924 Hz component is the largest (see Figure 4.26). The shape of the Spectral Envelope of the load at all three frequencies is also roughly rectangular, starting when the load turns on and ending when the load turns off.

The vegetable sprayer load was attached to the junction of the pipe end, next to the shower head load (see Figure 4.27). The vegetable sprayer was also turned on and off by the same ball valve with a handle as the shower head. The vegetable sprayer shared a fixture with the shower head such that only one or the other could operate, but not both at the same time.

The vegetable sprayer load was operated alone. The ESD of the accelerometer data (see Figure 4.28) also has peaks at the three beam-like modes of the pipe segment found in the lateral vibration of a pipe segment experiments. The 72 Hz peak is the largest in magnitude of the three. The 384 Hz peak is smaller, and the 924 Hz peak is in between. This is different from the magnitude ordering of the garden hose and the shower head.

The Spectral Envelopes were calculated and plotted for the vegetable sprayer load at the three peak frequencies. The magnitudes of the three vegetable sprayer Spectral Envelopes

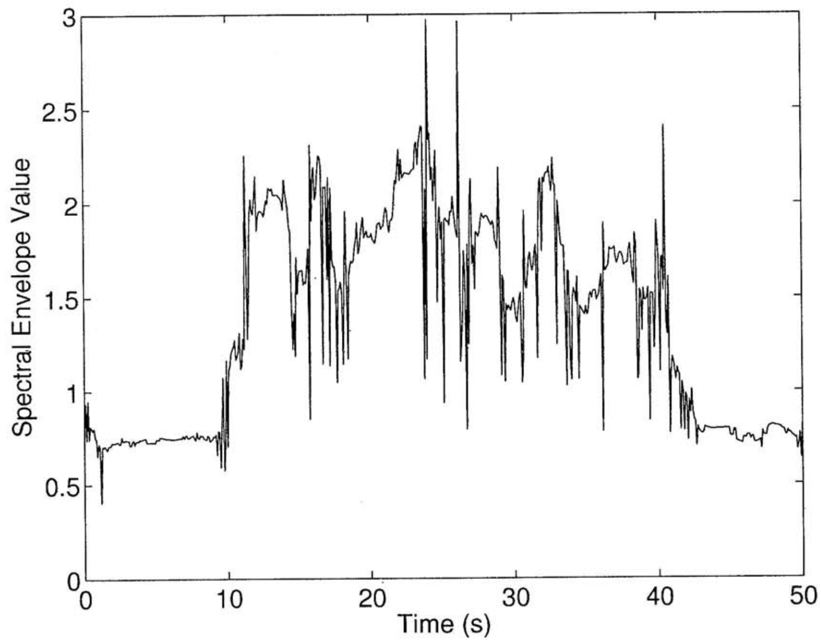


Figure 4.19: Spectral Envelope for Garden Hose at Frequency 72 Hz

once again preserve the rank order of the peak magnitudes in the vegetable sprayer ESD plot (see Figure 4.28), but in an unusual way because of the shapes of the Spectral Envelopes. The Spectral Envelope of the development of the 72 Hz component of the vibration over time is the largest in magnitude because of high spikes in the turn on and shut off transients, but it is much smaller in magnitude in steady state operation (see Figure 4.29). It has a distinct shape that is rectangular in the middle, with spikes at either end. The Spectral Envelope of the development of the 384 Hz component of the vibration over time is the smallest, and rectangular in shape (see Figure 4.30). The Spectral Envelope of the 924 Hz component is in between, and rectangular in shape (see Figure 4.31).

As seen in the above experiments, all of the individual loads excite the same three modes of the pipe (see Figure 4.18, Figure 4.23, and Figure 4.28). However, they excite these same modes in different magnitude rank orders (see Table 4.1). So, even if two different loads have Spectral Envelope transients that look similar in shape at the same frequency, they can be distinguished from one another and correctly identified by also observing them at a different frequency. For example, the garden hose load and the shower head load Spectral Envelopes are both roughly rectangular at 924 Hz (see Figure 4.21 and Figure 4.26). These loads can be identified correctly by observing their Spectral Envelopes at both 924 Hz and 384 Hz. According to the magnitude rank order of the peaks in the ESD for the garden hose, its Spectral Envelope at 384 Hz is larger than at 924 Hz. The ESD for the shower head, on the other hand, indicates that its Spectral Envelope at 384 Hz is smaller than

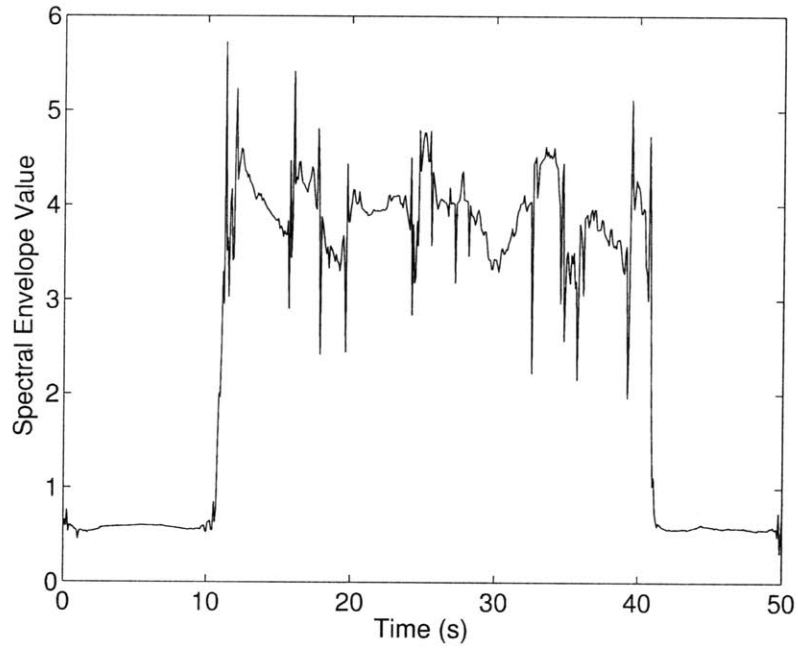


Figure 4.20: Spectral Envelope for Garden Hose at Frequency 384 Hz

at 924 Hz. These relationships can be seen by comparing the Spectral Envelope plots at 384 Hz (see Figure 4.20 and Figure 4.25) to the Spectral Envelopes at 924 Hz (see Figure 4.21 and Figure 4.26). The loads are identifiable from one another because of their ESD peak magnitude rank order, even when their Spectral Envelope shapes are similar at most frequencies.

	72 Hz	384 Hz	924 Hz
Garden Hose	Smallest	Largest	Middle
Shower Head	Smallest	Middle	Largest
Vegetable Sprayer	Largest	Smallest	Middle

Table 4.1: Summary of ESD Peak Magnitude Rank Orders for Laboratory Loads

Some loads are readily identifiable from one another because while their Spectral Envelopes are similar in shape at one frequency, they are very different in shape at another frequency (see Table 4.2). For example, the shower head load and the vegetable sprayer load Spectral Envelope shapes are both roughly rectangular at 924 Hz (see Figure 4.26 and Figure 4.31). However, at 72 Hz, the Spectral Envelope for the shower head is still roughly rectangular throughout (see Figure 4.24), while the Spectral Envelope for the vegetable sprayer load looks different because of the sharp spikes at its turn on and shut off transients. The loads are identifiable because of differences in their Spectral Envelope shapes at some frequencies, even while their Spectral Envelope shapes are similar in other frequencies.

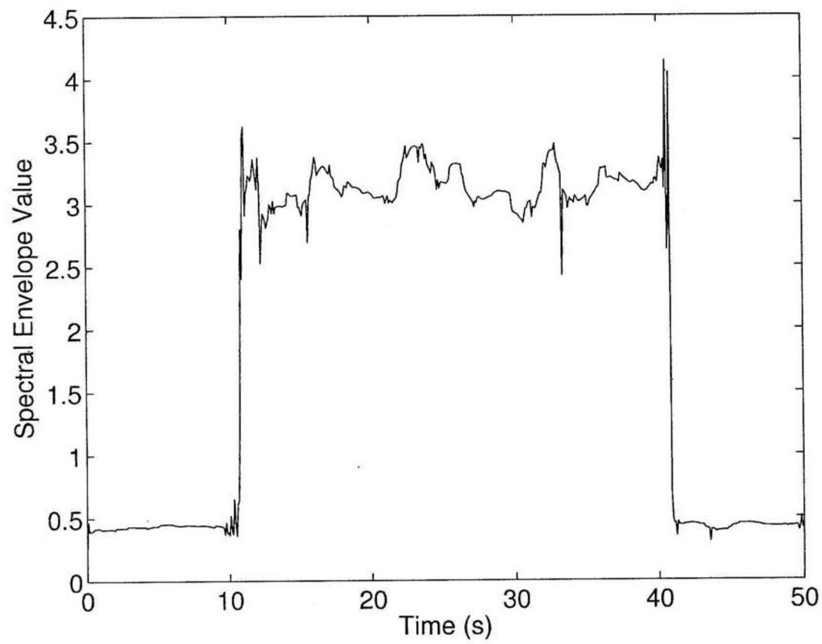


Figure 4.21: Spectral Envelope for Garden Hose at Frequency 924 Hz

	72 Hz	384 Hz	924 Hz
Garden Hose	Rectangular	Rectangular	Rectangular
Shower Head	Rectangular	Rectangular	Rectangular
Vegetable Sprayer	Start and Stop Spikes	Rectangular	Rectangular

Table 4.2: Summary of Spectral Envelope Shapes for Laboratory Loads

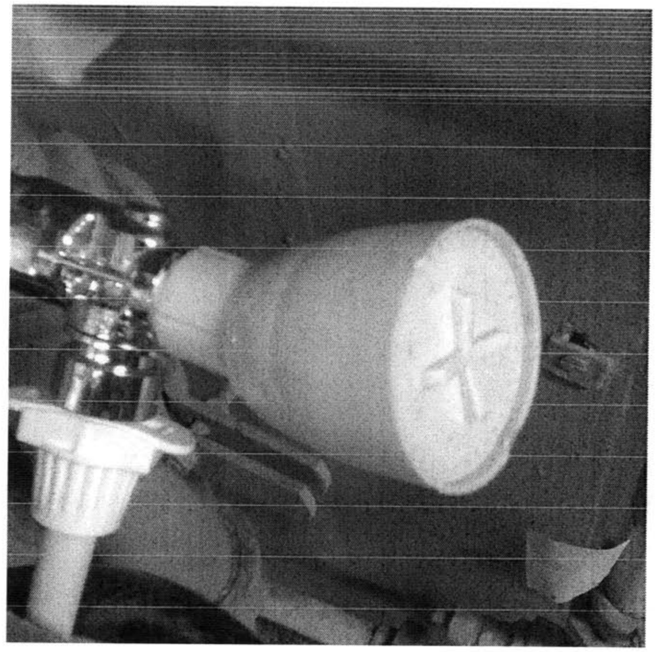


Figure 4.22: Photo of Shower Head Load

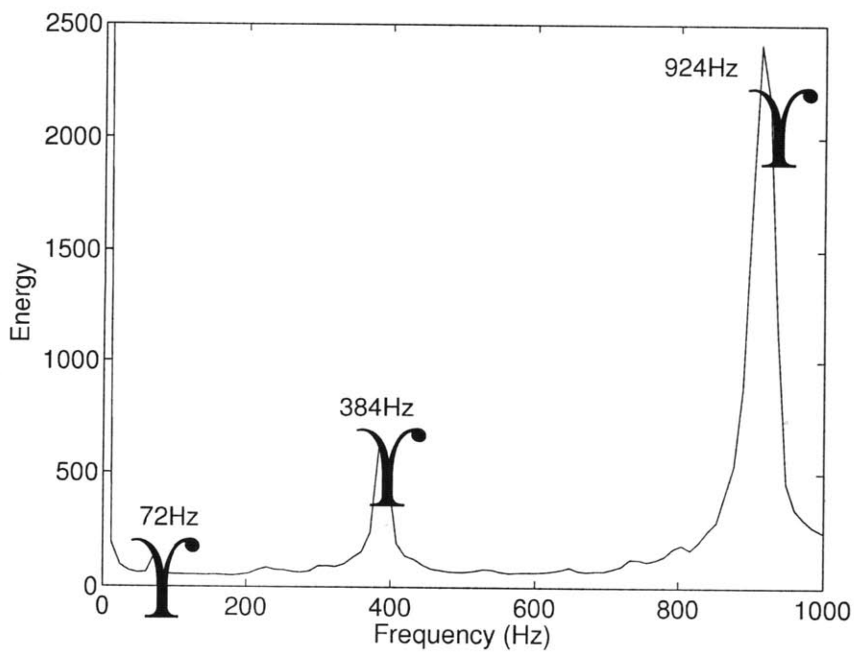


Figure 4.23: ESD for Shower Head

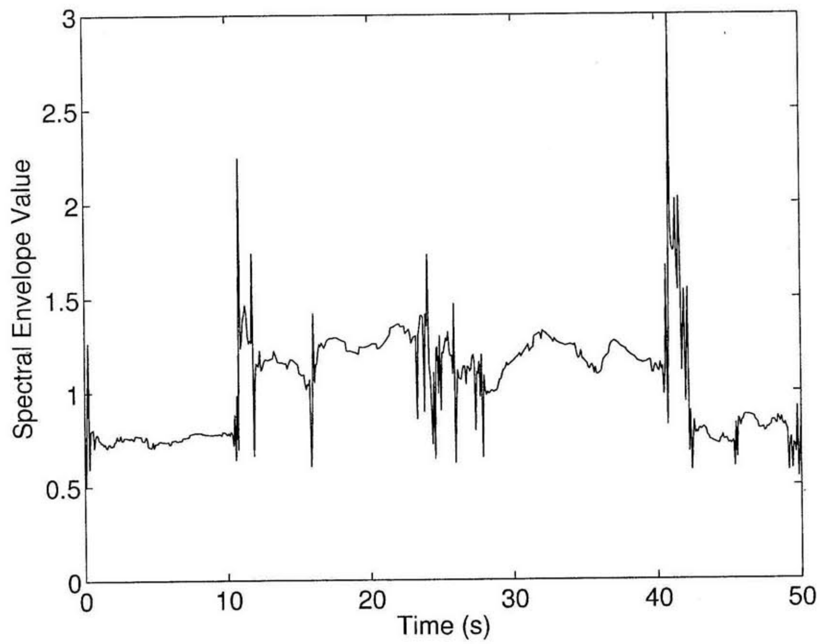


Figure 4.24: Spectral Envelope for Shower Head at Frequency 72 Hz

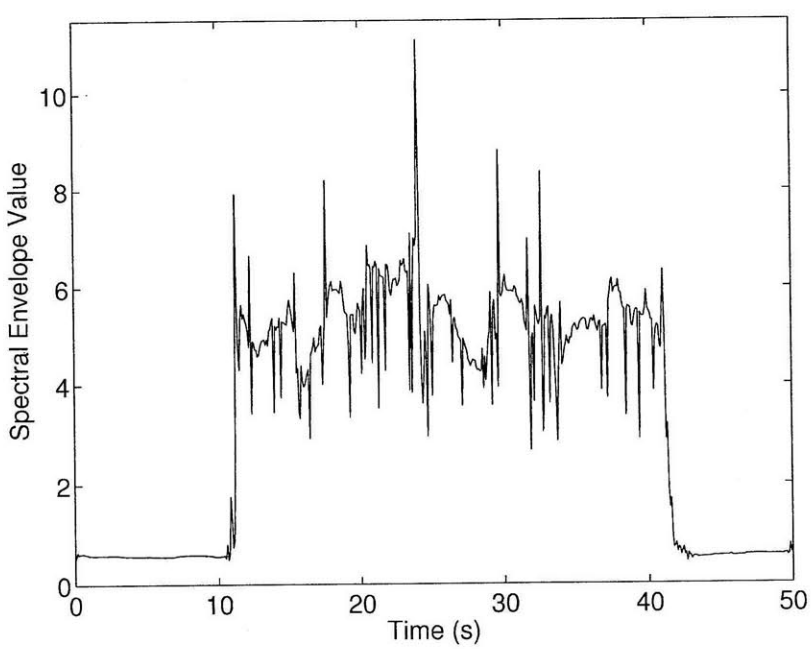


Figure 4.25: Spectral Envelope for Shower Head at Frequency 384 Hz

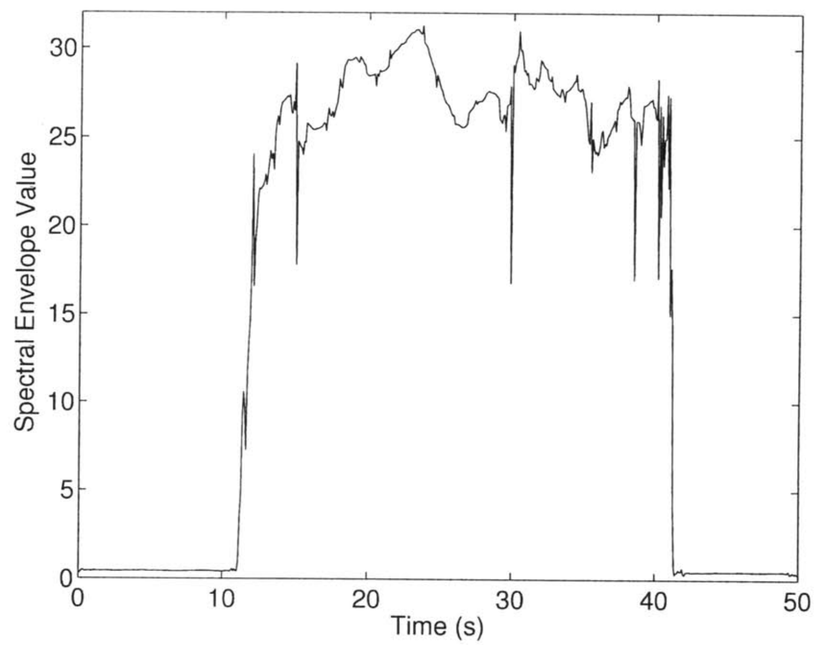


Figure 4.26: Spectral Envelope for Shower Head at Frequency 924 Hz

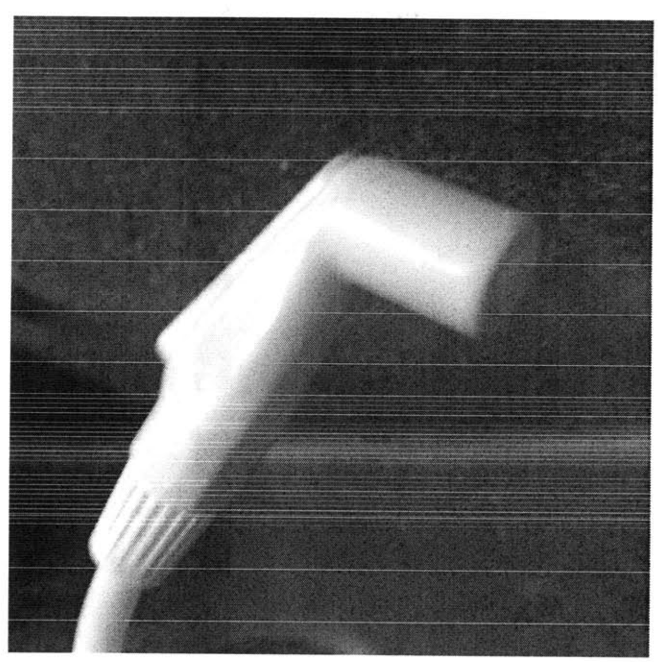


Figure 4.27: Photo of Vegetable Sprayer

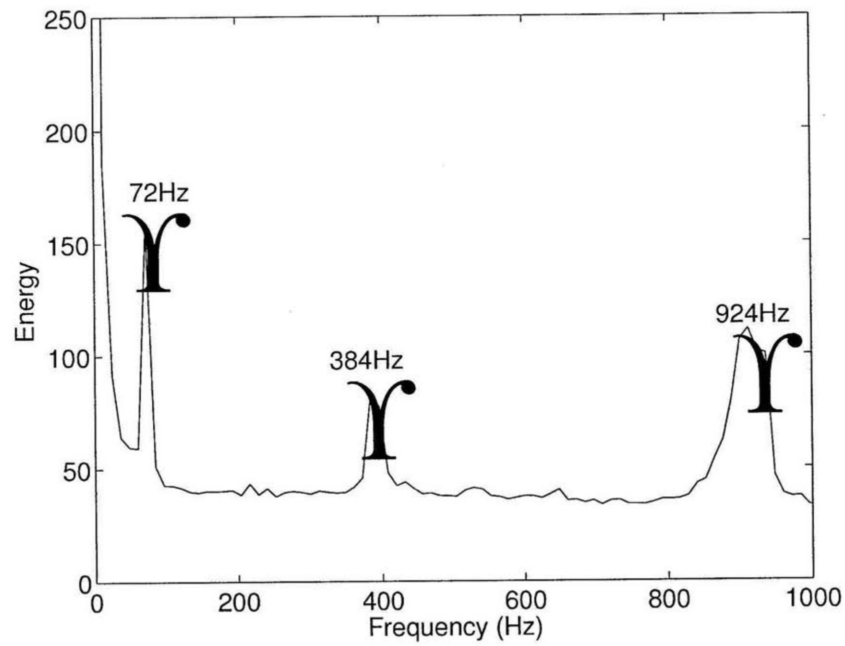


Figure 4.28: ESD for Vegetable Sprayer

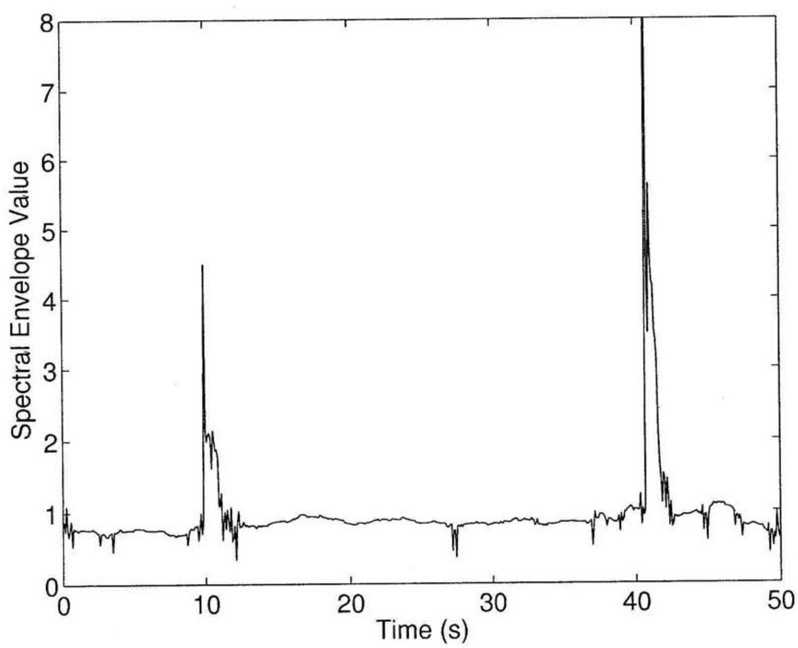


Figure 4.29: Spectral Envelope for Vegetable Sprayer at Frequency 72 Hz

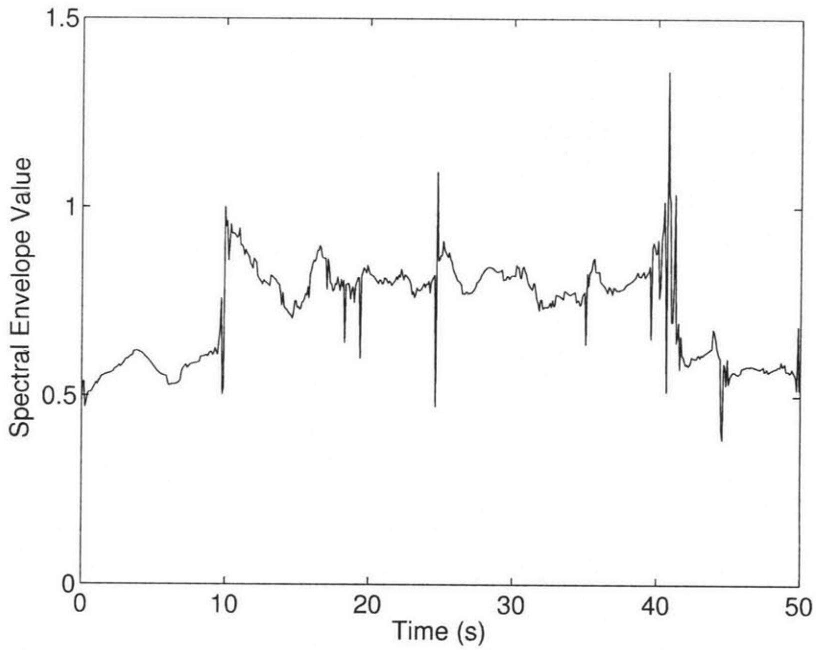


Figure 4.30: Spectral Envelope for Vegetable Sprayer at Frequency 384 Hz

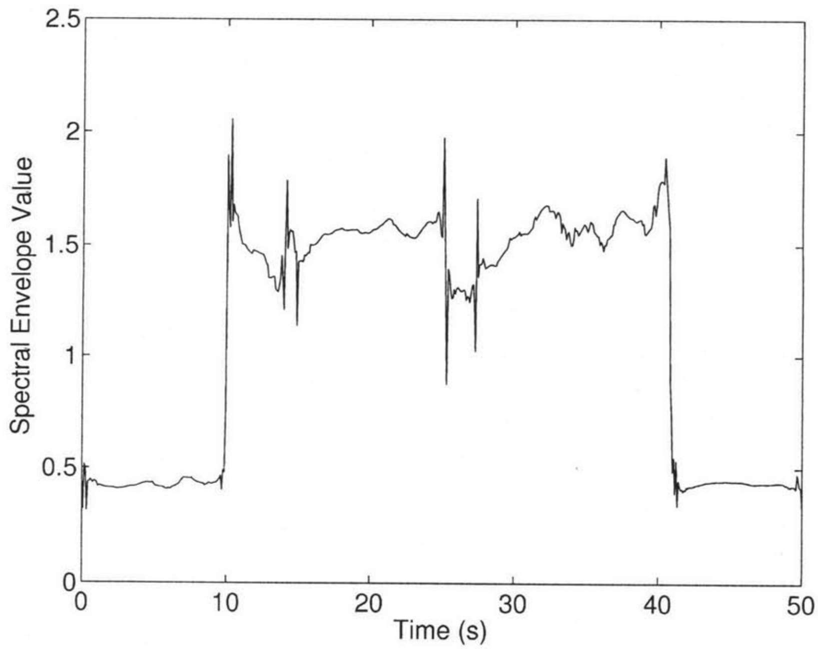


Figure 4.31: Spectral Envelope for Vegetable Sprayer at Frequency 924 Hz

4.4 Multiple Load Transients Experiments

For demonstrating that the turn on and turn off events of multiple loads in overlapping operation can be detected, and that multiple loads in overlapping operation can still be identified, the same experimental setup as that for individual load transient experiments was used (see Figure 4.32).

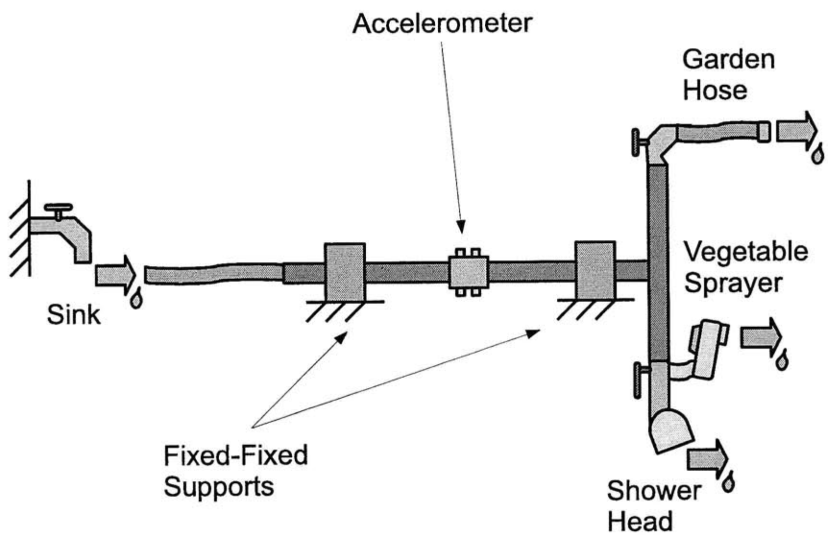


Figure 4.32: Diagram of General Multiple Load Transient Experiment Setup

First, the garden hose and the shower head were run in overlapping operation. The garden hose was turned on, and about 10 seconds into its operation, the shower head was also turned on. After another 10 s, the shower head was turned off. After 10 s more, the garden hose was also turned off. The ESD for the whole experiment demonstrates the same three modal frequencies as before (see Figure 4.33).

The Spectral Envelopes at 72 Hz, 384 Hz, and 924 Hz (see Figure 4.34, Figure 4.35, and Figure 4.36) have dashed lines drawn on them to indicate the start and the end of the shower head load, which is the period when both loads were run in overlapping operation. On either side of the dashed lines is the garden hose operating alone.

It is difficult to detect the turn on and turn off events of the shower head in the Spectral Envelope at 72 Hz (see Figure 4.34). However, it is very easy to detect the turn on and turn off events of the shower head in the Spectral Envelope at 924 Hz (see Figure 4.36). This is because, according to the ESD peak magnitude ratios for the garden hose alone and the

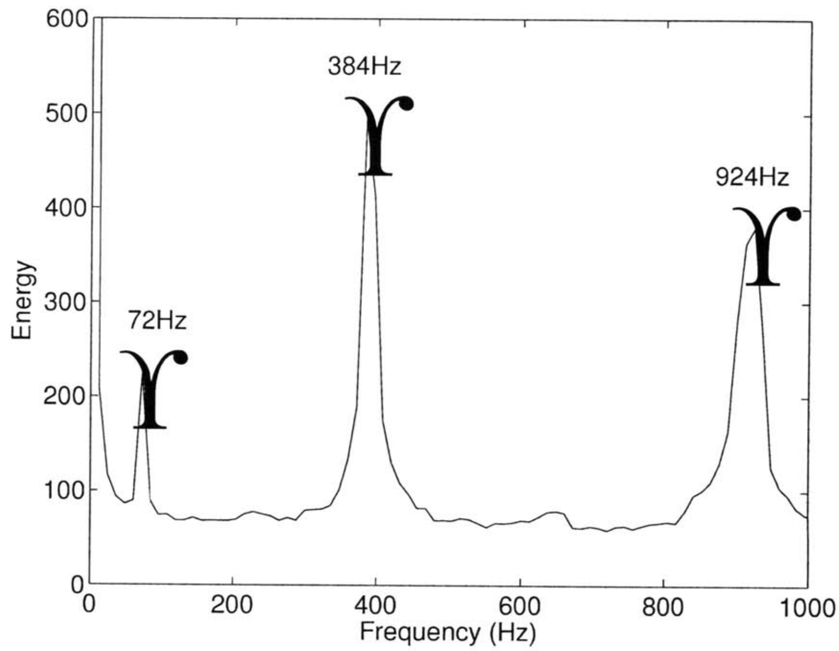


Figure 4.33: ESD for Garden Hose and Shower Head in Overlapping Operation

shower head alone (see Figure 4.18 and Figure 4.23), the garden hose characteristically has a little more energy at the 924 Hz band than at the 72 Hz band, whereas the shower head has significantly more energy at the 924 Hz band than at the 72 Hz band. The characteristic ratios of ESD frequency band peaks between different loads makes the shower head Spectral Envelope content more prominent than the garden hose Spectral Envelope content at 924 Hz, so the turn on and turn off events of the shower head are easy to detect at that frequency.

The garden hose and the vegetable sprayer were also run in overlapping operation. The garden hose was turned on, and about 10 seconds into its operation, the vegetable sprayer was also turned on. After another 10 s, the vegetable sprayer was turned off. After 10 s more, the garden hose was also turned off. The ESD for the whole experiment again demonstrates the same three modal frequencies from all previous loads (see Figure 4.37).

Again, the Spectral Envelopes at 72 Hz, 384 Hz, and 924 Hz (see Figure 4.38, Figure 4.39, and Figure 4.40) have dashed lines drawn on them to indicate the start and the end of the shower head load, which is the period when both loads were run in overlapping operation. On either side of the dashed lines is the garden hose operating alone.

It is difficult to detect the turn on and turn off events of the vegetable sprayer in the Spectral Envelope at 72 Hz (see Figure 4.38). However, it is easy to detect the turn on and turn off events of the vegetable sprayer in the Spectral Envelope at 924 Hz (see Figure 4.40).

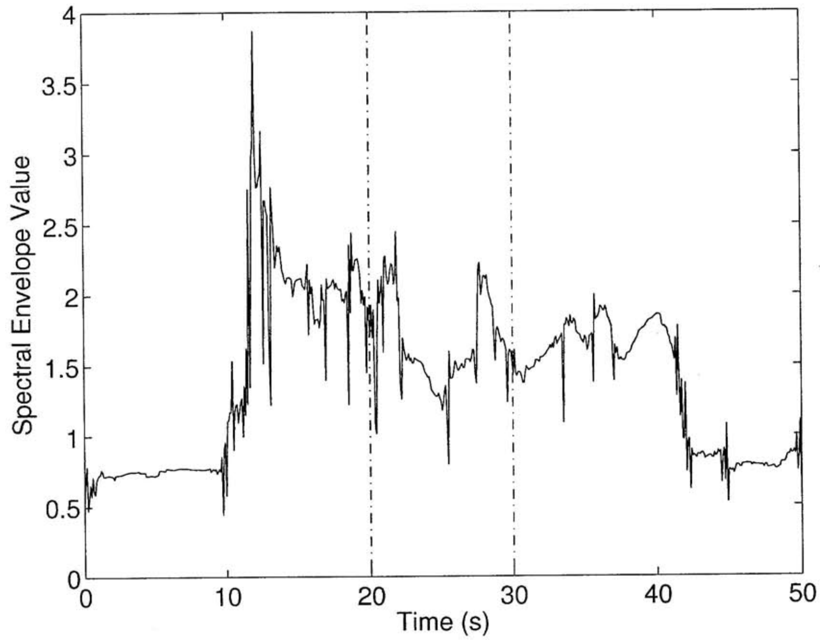


Figure 4.34: Spectral Envelope for Garden Hose and Shower Head in Overlapping Operation at 72 Hz

According to the ESD peak magnitude ratios for the garden hose alone and the vegetable sprayer alone (see Figure 4.18 and Figure 4.28), the garden hose characteristically has a little more energy at the 924 Hz band than at the 72 Hz band. The vegetable sprayer has less at the 924 Hz band than at the 72 Hz band. The characteristic ratios of ESD frequency band peaks between different loads create a dip in the combined Spectral Envelope content of the garden hose and vegetable sprayer at 924 Hz(see Figure 4.40), so the turn on and turn off events of the vegetable sprayer are easy to detect at that frequency.

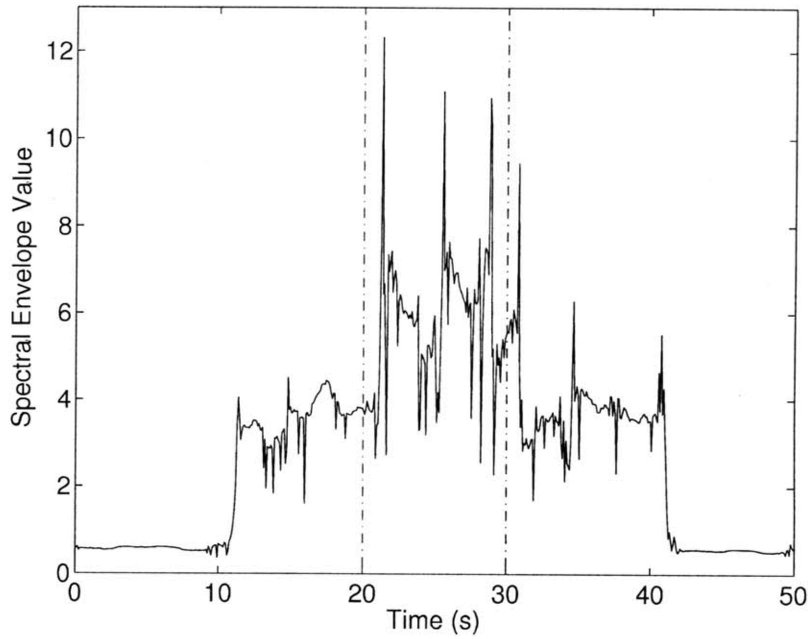


Figure 4.35: Spectral Envelope for Garden Hose and Shower Head in Overlapping Operation at 384 Hz

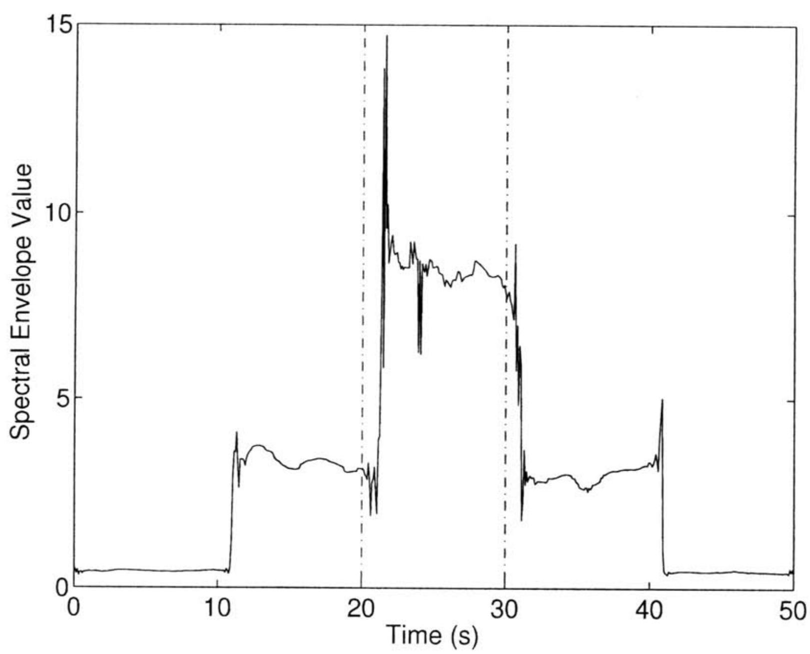


Figure 4.36: Spectral Envelope for Garden Hose and Shower Head in Overlapping Operation at 924 Hz

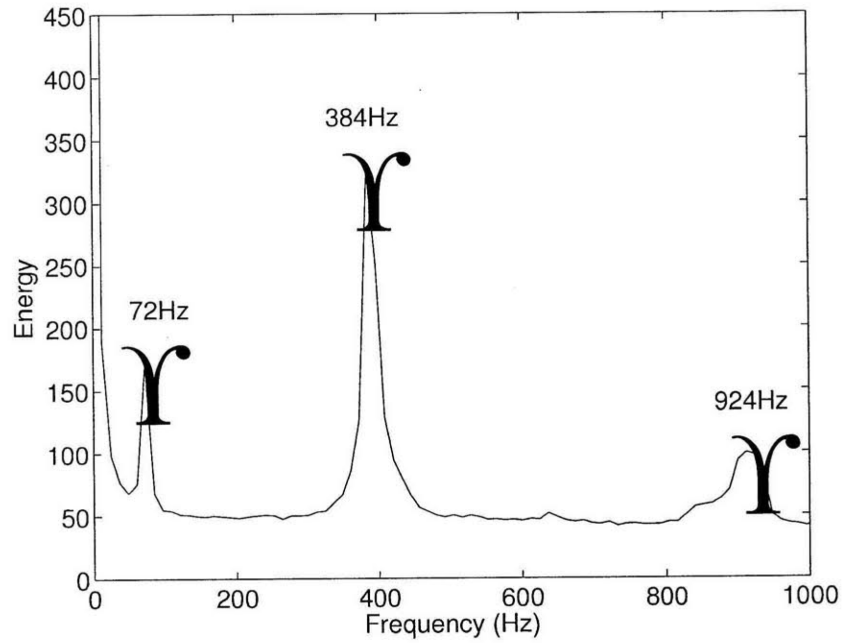


Figure 4.37: ESD for Garden Hose and Vegetable Sprayer in Overlapping Operation

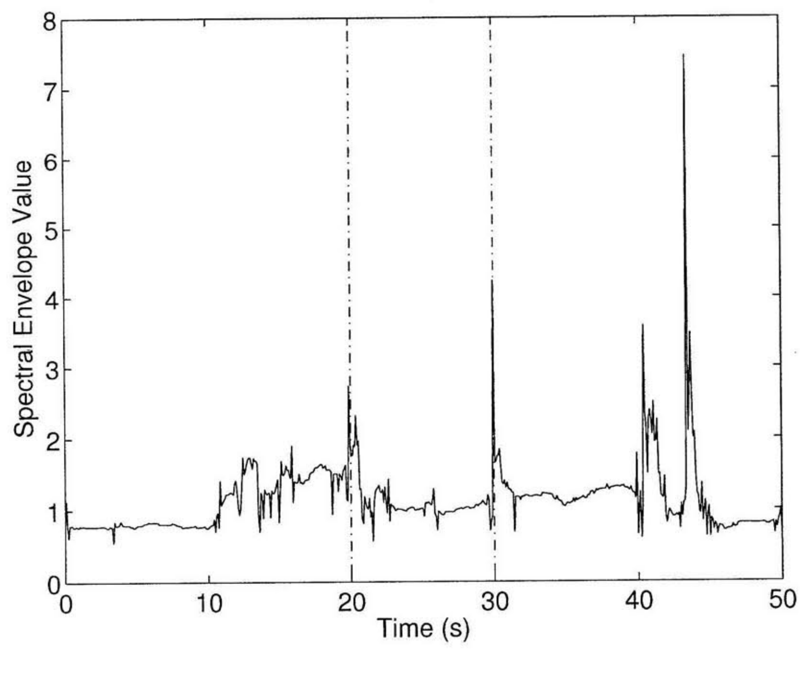


Figure 4.38: Spectral Envelope for Garden Hose and Vegetable Sprayer in Overlapping Operation at 72 Hz

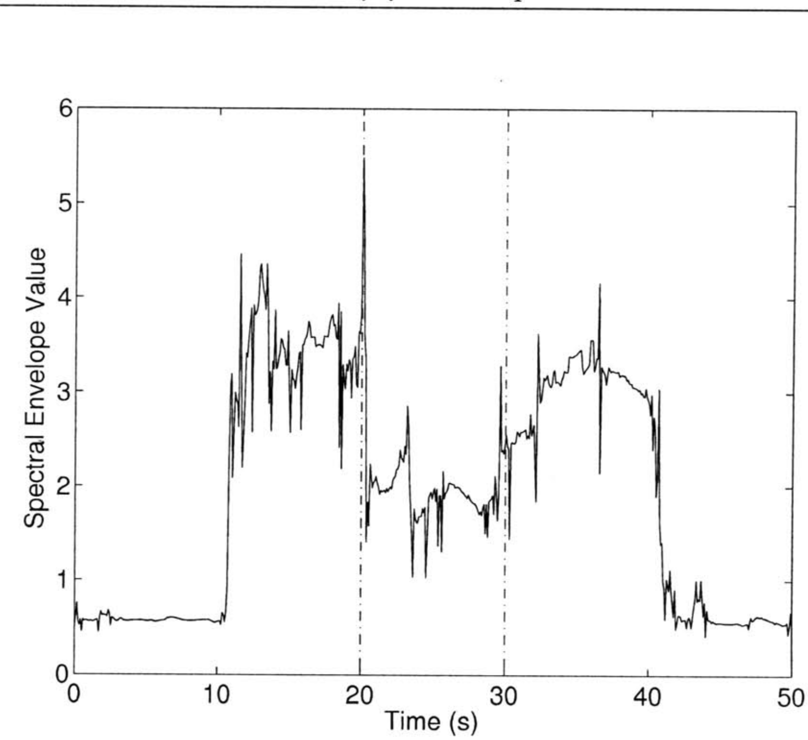


Figure 4.39: Spectral Envelope for Garden Hose and Vegetable Sprayer in Overlapping Operation at 384 Hz

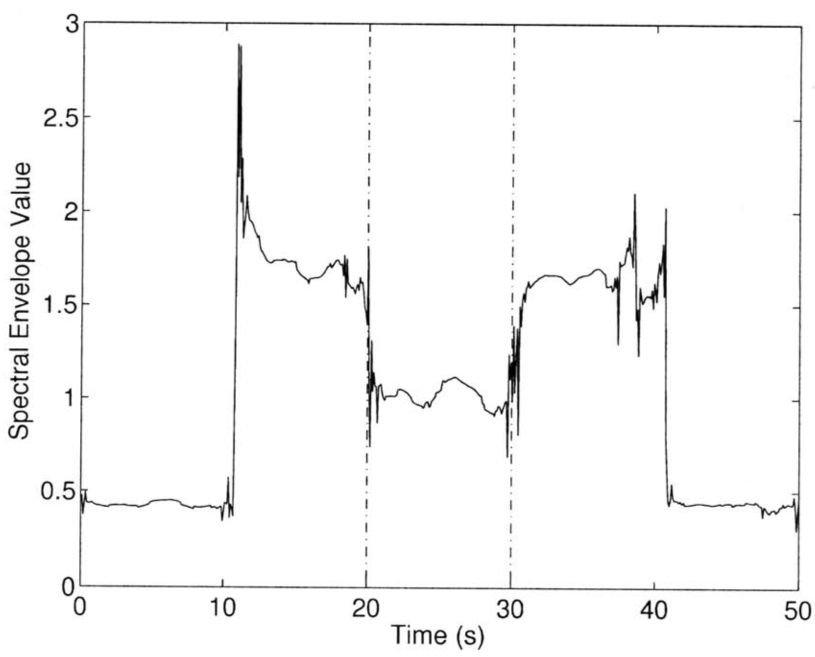


Figure 4.40: Spectral Envelope for Garden Hose and Vegetable Sprayer in Overlapping Operation at 924 Hz

4.5 Flow Rate Experiments

Flow rate experiments were performed to observe the effect of water flow rate through a pipe segment on its lateral vibration. To perform these experiments, a hose connected the sink water source to an inline flow meter. The inline flow meter connected with hose to the pipe segment mounted with fixed-fixed supports. At the end of the pipe segment, the adjustable garden hose spigot was used to vary the flow of water out through the garden hose load. The accelerometer was mounted on the middle of the pipe segment (see Figure 4.41).

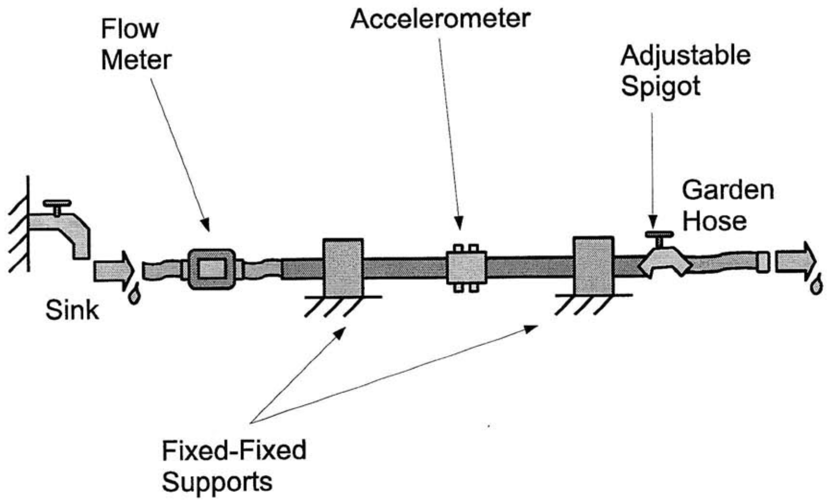


Figure 4.41: Flow Rate Experiment Setup

Water was run through the system at six different flow rates using the adjustable spigot and the display on the inline flow meter to set the flow rate. The six rates were 0.0, 1.1, 2.0, 3.0, 4.0, and 4.8 Gallons Per Minute (GPM). The ESD graphs of the six flow rates are plotted together (see Figure 4.42). The data does not appear to display any consistent trend.

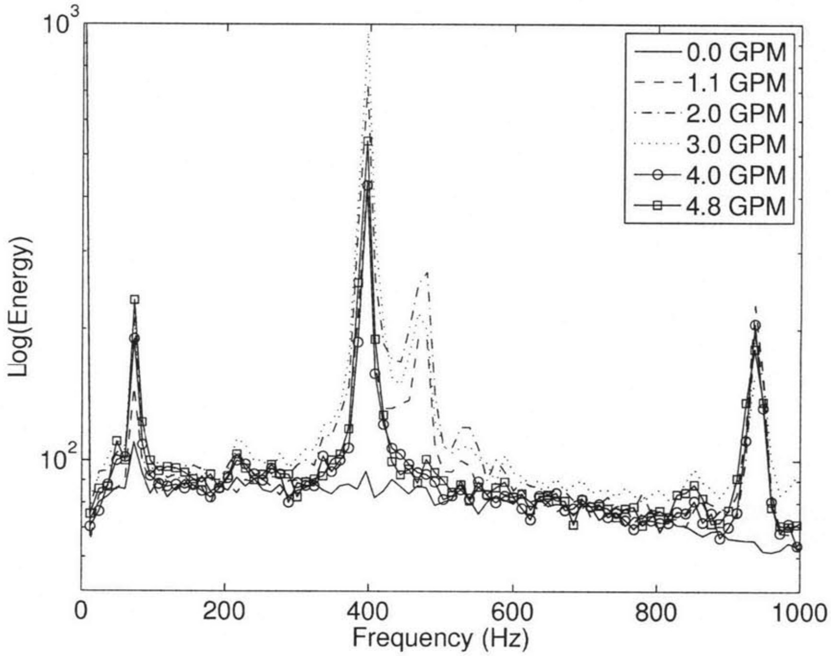


Figure 4.42: ESD for Different Flow Rates

4.6 Field Experiments

Field experiments were conducted in the basement of a single-family house in New Hampshire. The vibration sensor used for the field experiments was a microphone attached to a pipe segment near the main water intake to the house, in the basement. Loads on the plumbing network included a shower, a kitchen sink, and a bathroom sink (see Figure 4.43). These loads were operated, both alone and overlapping. Data was collected for individual and overlapping load operation, and processed in MATLAB.

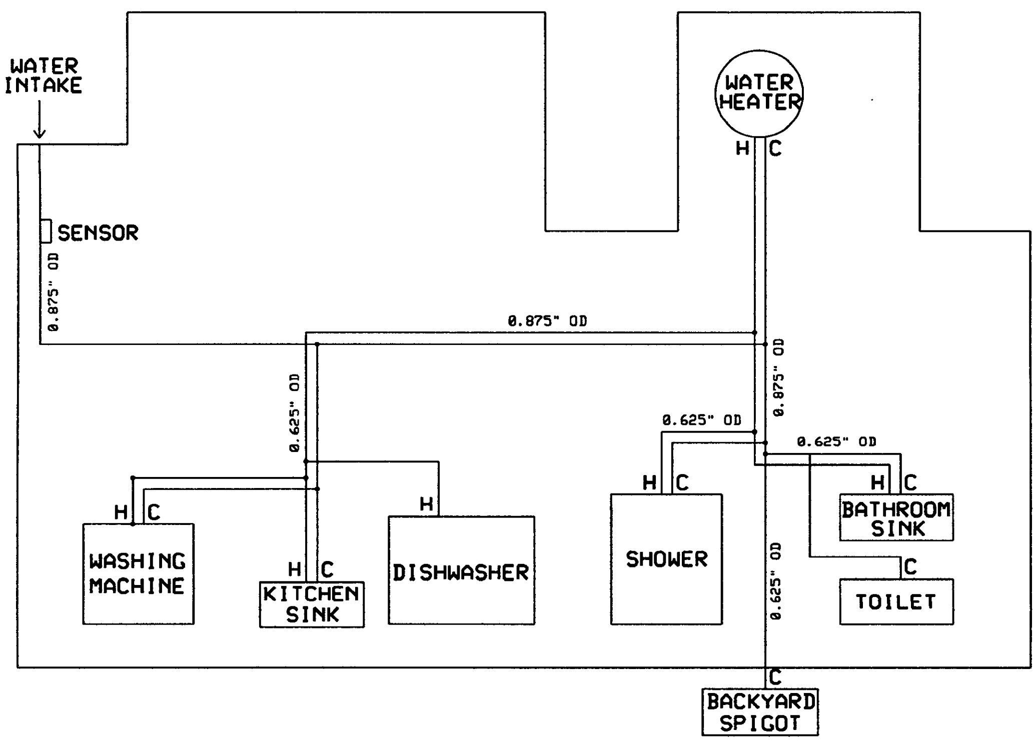


Figure 4.43: Diagram of Field Test Site Pipe Network

One of the loads operated was a shower in the bathroom of the house (see Figure 4.44). The ESD of the shower demonstrated several frequency peaks including ones at 324 Hz, 576 Hz, and 1380 Hz (see Figure 4.45). Of these three peaks, 576 Hz is the largest in magnitude, 1380 Hz is the smallest in magnitude, and 324 Hz is in between.

The Spectral Envelopes of the shower load maintained the magnitude rank order of the shower ESD peaks. The magnitude of the Spectral Envelope at 576 Hz is the largest, the Spectral Envelope at 1380 Hz is the smallest, and the Spectral Envelope at 324 Hz is in

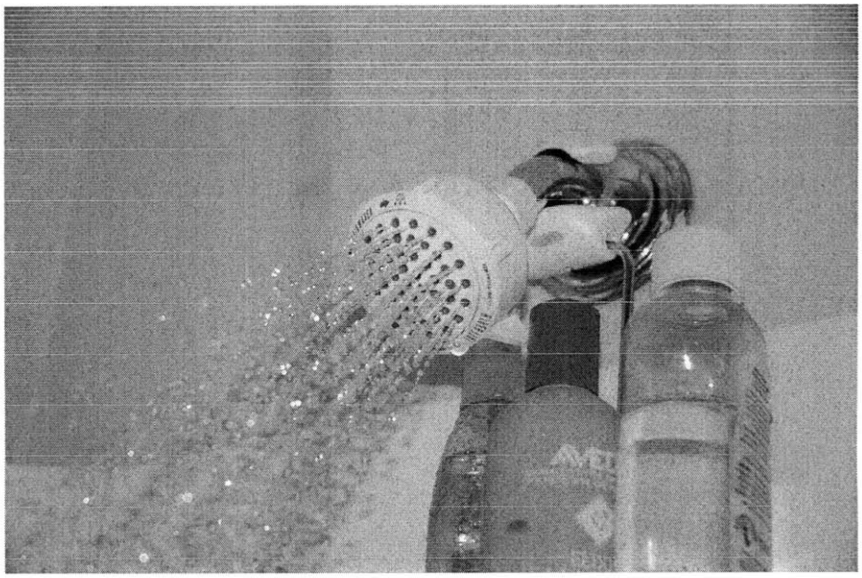


Figure 4.44: Photo of Shower

between (see Figure 4.46, Figure 4.47, and Figure 4.48). The Spectral Envelope shape is distinctive at all three frequencies. It starts with a sharp spike at the turn on transient, before leveling off at steady state operation.

Another load operated at the field test site was the sink in the kitchen (see Figure 4.49). The ESD of the kitchen sink load also contained peaks at 324 Hz, 576 Hz, and 1380 Hz (see Figure 4.50). These are the same frequencies where the shower load exhibited peaks. Of these three peaks, 576 Hz is the largest in magnitude, 1380 Hz is the smallest in magnitude, and 324 Hz is in between.

The Spectral Envelopes of the kitchen sink load also preserved the magnitude rank order of the kitchen sink ESD peaks (see Figure 4.51, Figure 4.52, and Figure 4.53). The Spectral Envelope shape is roughly rectangular at all three peak frequencies.

A third load operated at the field test site was the sink in the bathroom (see Figure 4.54). The ESD of the bathroom sink load again contained peaks at 324 Hz, 576 Hz, and 1380 Hz (see Figure 4.55). These are the same frequencies where the shower load and the kitchen sink load also exhibited peaks. Of these three peaks, 324 Hz is the largest in magnitude, 576 Hz is the smallest in magnitude, and 1380 Hz is in between.

The Spectral Envelopes of the bathroom sink load again preserved the magnitude rank

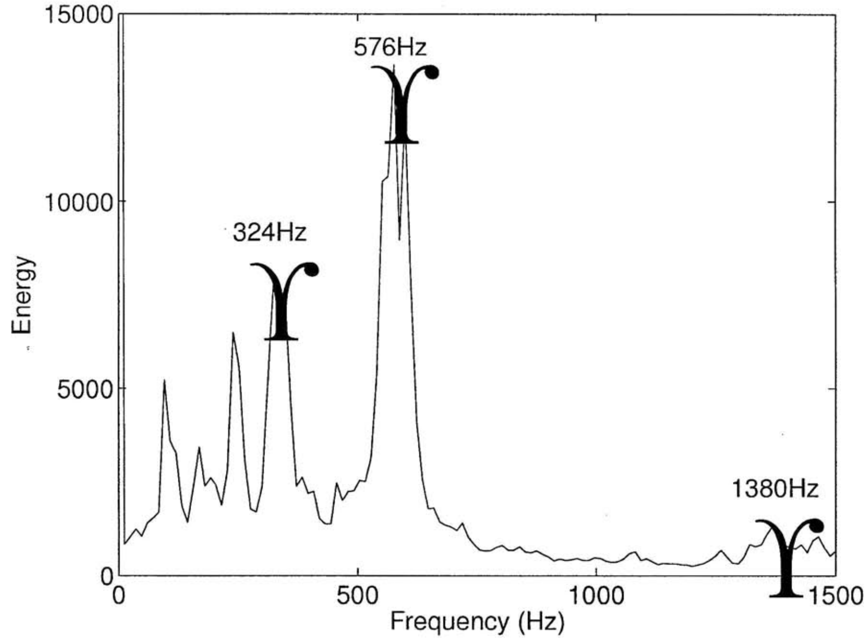


Figure 4.45: ESD for Shower

order of the bathroom sink ESD peaks (see Figure 4.56, Figure 4.57, and Figure 4.58). The Spectral Envelope shape is roughly rectangular at all three peak frequencies.

In the above field experiments, as in the laboratory experiments, all of the individual loads excite the same three modes of the pipe (see Figure 4.45, Figure 4.50, and Figure 4.55). However, they excite these same modes in different magnitude rank orders (see Table 4.3). So, even if two different loads have Spectral Envelope transients that look similar in shape at the same frequency, they can be distinguished from one another and correctly identified by also observing them at a different frequency. For example, the kitchen sink load and the bathroom sink load Spectral Envelopes are both roughly rectangular at 324 Hz (see Figure 4.51 and Figure 4.56). These loads can be identified correctly by observing their Spectral Envelopes at both 324 Hz and 576 Hz. According to the magnitude rank order of the peaks in the ESD for the kitchen sink, its Spectral Envelope at 576 Hz is smaller than at 324 Hz. The ESD for the bathroom sink, on the other hand, indicates that its Spectral Envelope at 576 Hz is larger than at 324 Hz. These relationships can be seen by comparing the Spectral Envelope plots at 576 Hz (see Figure 4.52 and Figure 4.57) to the Spectral Envelopes at 324 Hz (see Figure 4.51 and Figure 4.56). Just like the laboratory loads, the field loads are identifiable from one another because of their ESD peak magnitude rank order, even when their Spectral Envelope shapes are similar at most frequencies.

Also like in the laboratory experiments, some field loads are readily identifiable from one

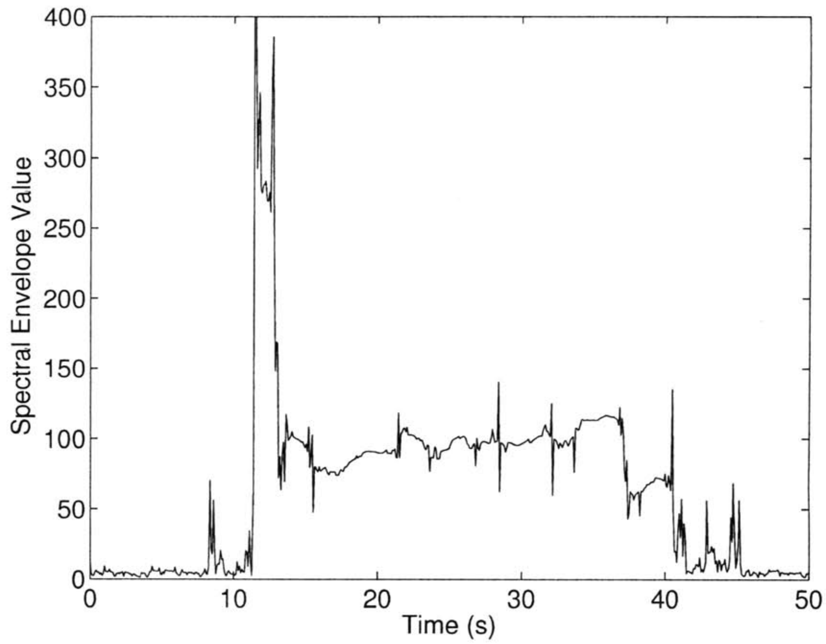


Figure 4.46: Spectral Envelope for Shower at 324 Hz

	324 Hz	576 Hz	1380 Hz
Shower	Middle	Largest	Smallest
Kitchen Sink	Largest	Smallest	Middle
Bathroom Sink	Middle	Largest	Smallest

Table 4.3: Summary of ESD Peak Magnitude Rank Orders for Field Loads

another because their Spectral Envelopes are very different in shape at some frequencies (see Table 4.4). For example, the shower load and the kitchen sink load Spectral Envelope shapes are very different at 324 Hz (see Figure 4.25 and Figure 4.51). At 324 Hz, the Spectral Envelope for the shower is roughly rectangular but starts with a large spike at its turn on transient (see Figure 4.46), while the Spectral Envelope for the vegetable sprayer load looks different because it is roughly rectangular throughout, with no transient spikes (see Figure 4.51). Like the laboratory loads, the field loads are identifiable because of differences in their Spectral Envelope shapes at some frequencies.

	324 Hz	576 Hz	1380 Hz
Shower	Start Spike	Start Spike	Start Spike
Kitchen Sink	Rectangular	Rectangular	Rectangular
Bathroom Sink	Rectangular	Rectangular	Rectangular

Table 4.4: Summary of Spectral Envelope Shapes for Field Loads

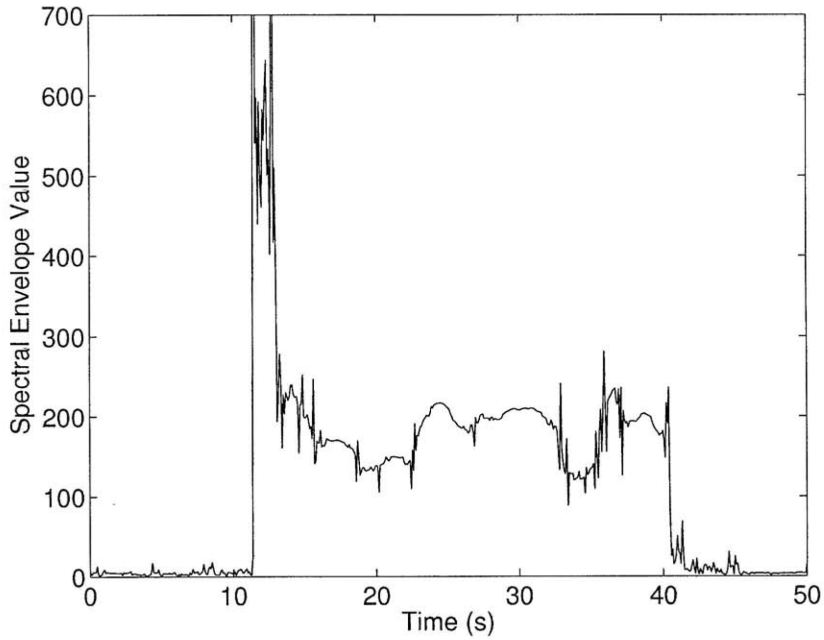


Figure 4.47: Spectral Envelope for Shower at 576 Hz

The field loads were also run in overlapping operation, like the laboratory loads. In one experiment, the shower and the kitchen sink were run in overlapping operation. The shower was turned on, and about 30 seconds into its operation, the kitchen sink was also turned on. After another 30 s, the kitchen sink was turned off. After 30 s more, the shower was also turned off. The ESD for the whole experiment demonstrates the same three modal frequencies as the individual field loads (see Figure 4.59).

The Spectral Envelopes at 324 Hz, 576 Hz, and 1380 Hz (see Figure 4.60, Figure 4.61, and Figure 4.62) have dashed lines drawn on them to indicate the start and the end of the kitchen sink load, which is the period when both loads were run in overlapping operation. On either side of the dashed lines is the shower operating alone.

The turn on and turn off events of the kitchen sink load are easiest to detect in the Spectral Envelope content at 324 Hz (see Figure 4.60. At 324 Hz, the kitchen sink load creates a sharp rectangular bump in the Spectral Envelope content. According to the table of ESD peak rank order (see Table 4.3), the kitchen sink load has its largest ESD peak at 324 Hz. The shower load has only its middling ESD peak at 324 Hz.

The shower and the bathroom sink were also run in overlapping operation. The shower was turned on, and about 30 seconds into its operation, the bathroom sink was also turned on. After another 30 s, the bathroom sink was turned off. After 30 s more, the shower was

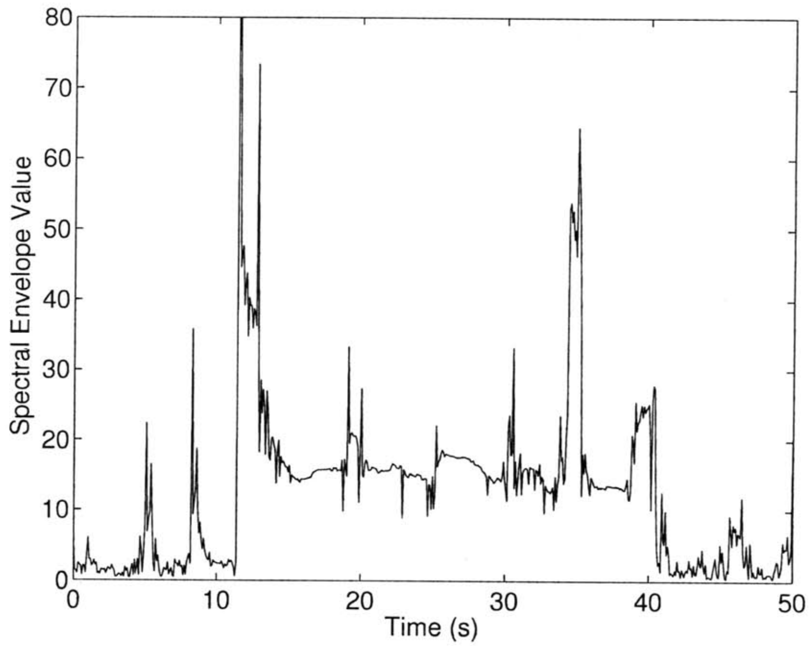


Figure 4.48: Spectral Envelope for Shower at 1380 Hz

also turned off. The ESD for the whole experiment demonstrates the same three modal frequencies as the individual field loads (see Figure 4.63).

The Spectral Envelopes at 324 Hz, 576 Hz, and 1380 Hz (see Figure 4.64, Figure 4.65, and Figure 4.66) have dashed lines drawn on them to indicate the start and the end of the bathroom sink load, which is the period when both loads were run in overlapping operation. On either side of the dashed lines is the shower operating alone.

The turn on and turn off events of the bathroom sink load are difficult to detect in the Spectral Envelope content at any of the three frequencies shown (see Figure 4.64, Figure 4.65, and Figure 4.66). At all three frequencies, the ESD rank order (see Table 4.3) for the shower load and the bathroom sink load match. The middling peaks are at 324 Hz, the largest peaks are at 576 Hz, and the smallest peaks are at 1380 Hz.

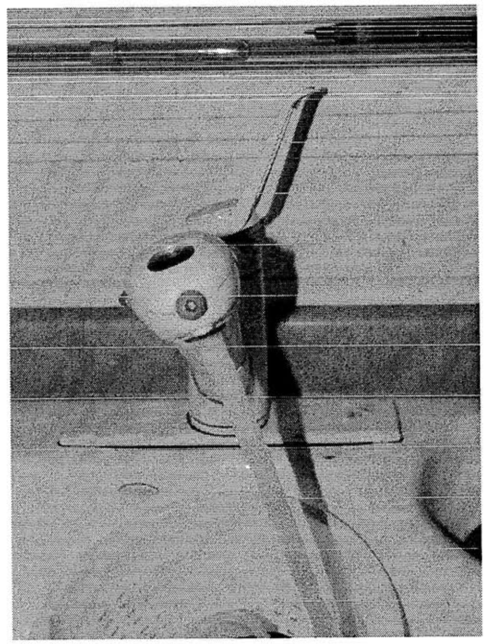


Figure 4.49: Photo of Kitchen Sink

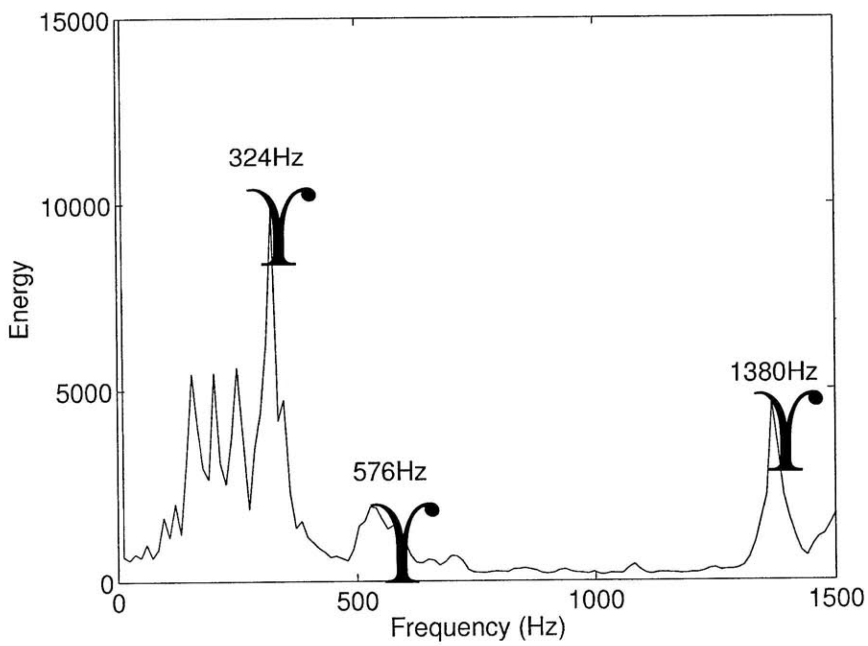


Figure 4.50: ESD for Kitchen Sink

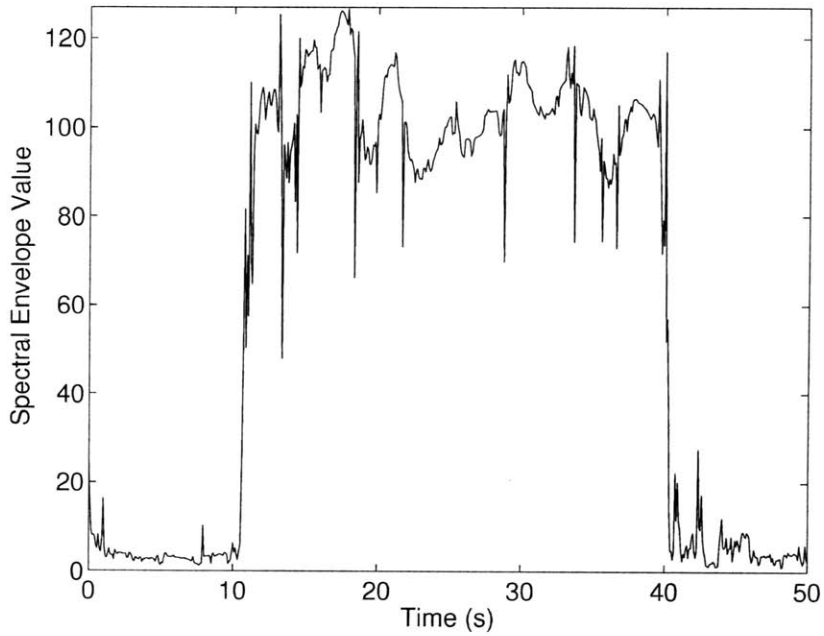


Figure 4.51: Spectral Envelope for Kitchen Sink at 324 Hz

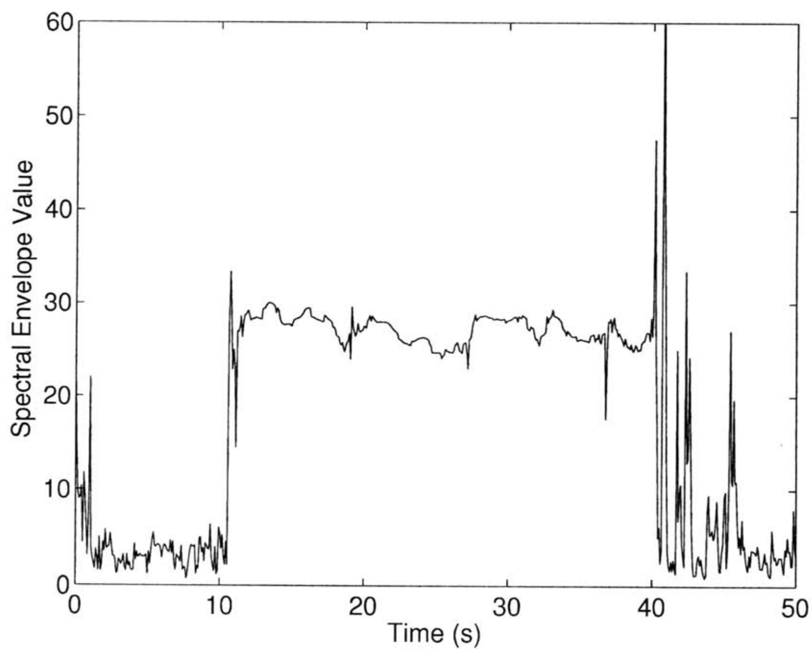


Figure 4.52: Spectral Envelope for Kitchen Sink at 576 Hz

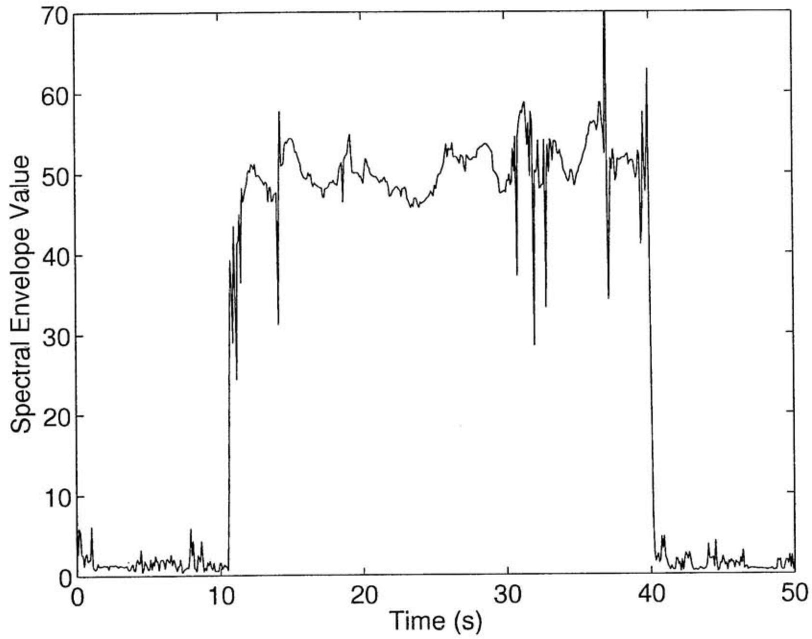


Figure 4.53: Spectral Envelope for Kitchen Sink at 1380 Hz

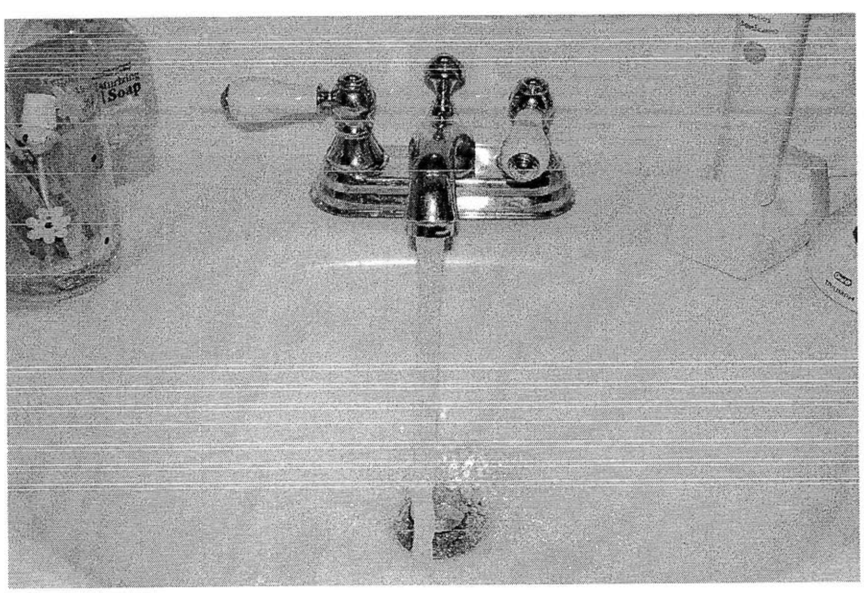


Figure 4.54: Photo of Bathroom Sink

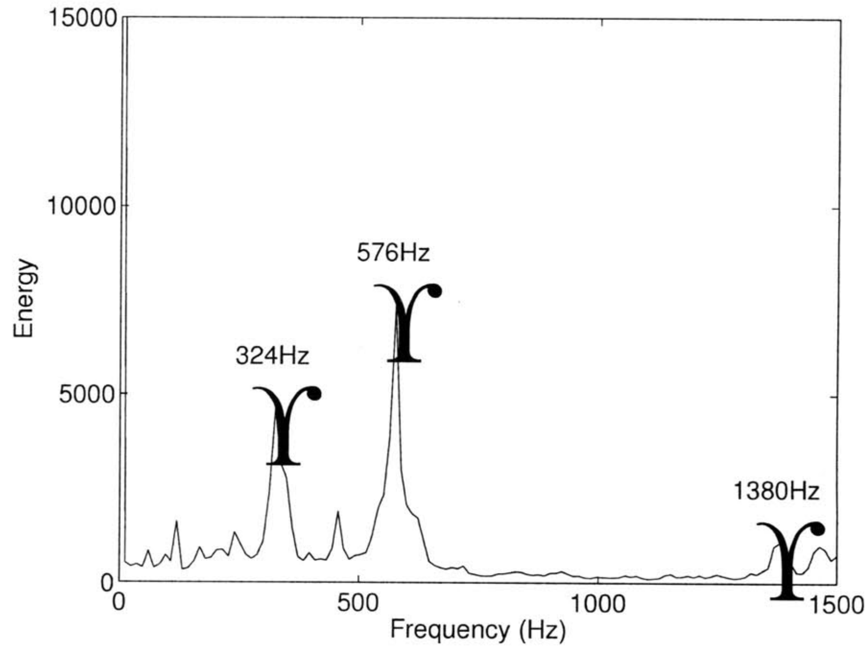


Figure 4.55: ESD for Bathroom Sink

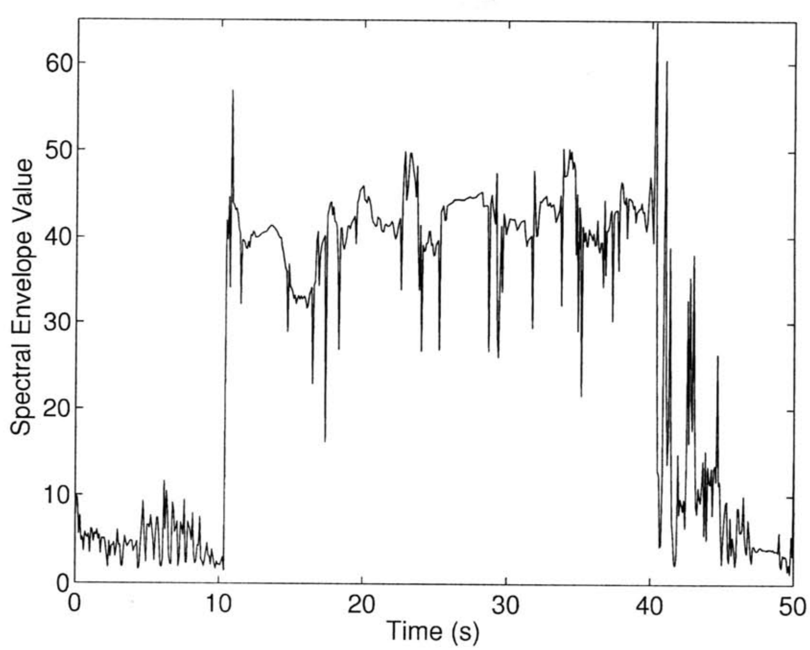


Figure 4.56: Spectral Envelope for Bathroom Sink at 324 Hz

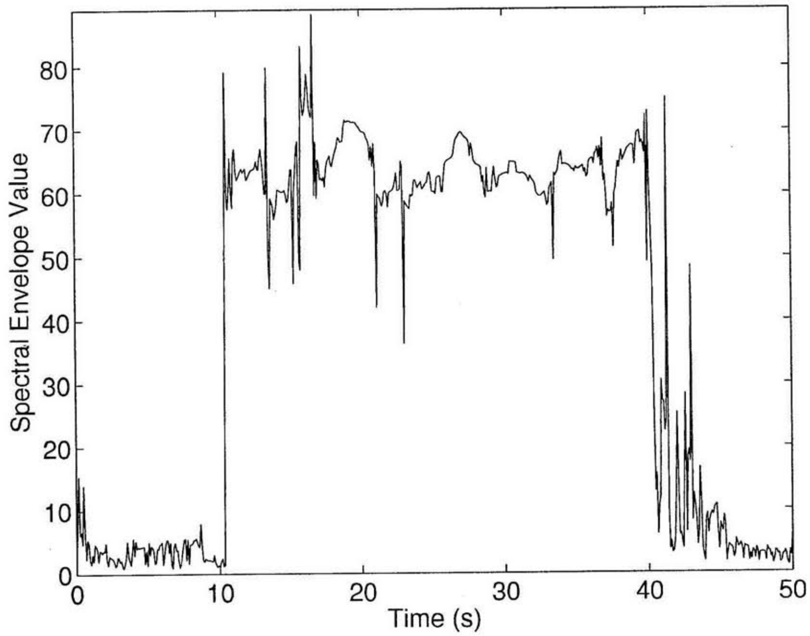


Figure 4.57: Spectral Envelope for Bathroom Sink at 576 Hz

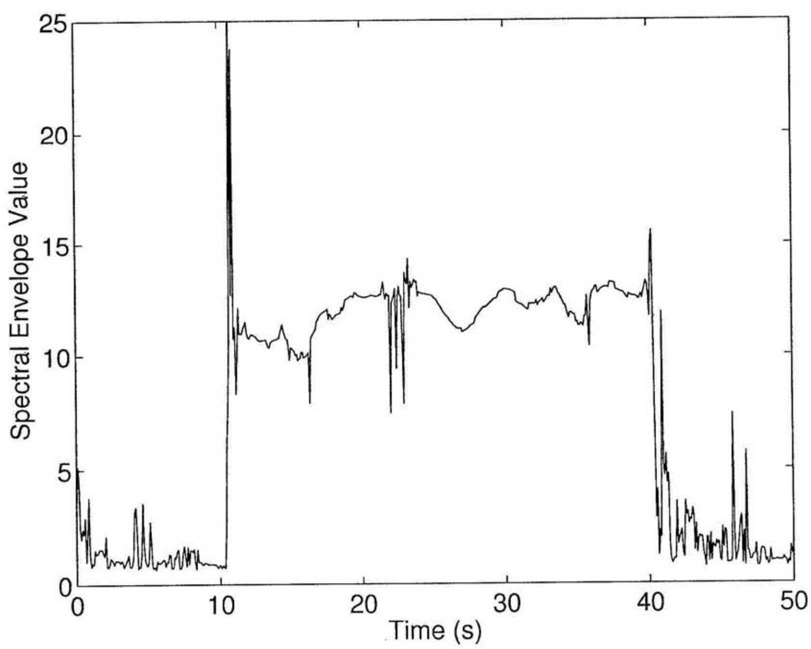


Figure 4.58: Spectral Envelope for Bathroom Sink at 1380 Hz

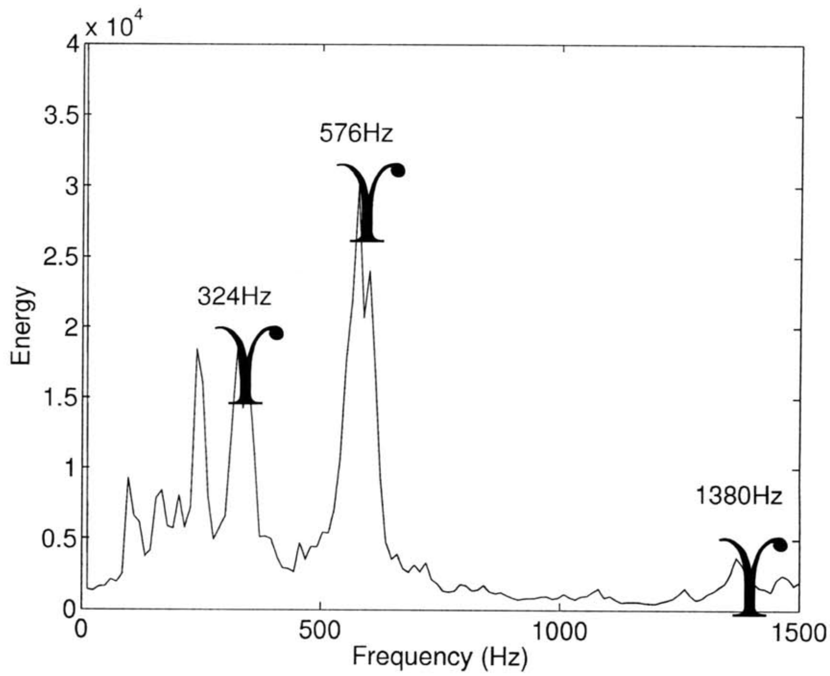


Figure 4.59: ESD for Shower and Kitchen Sink in Overlapping Operation

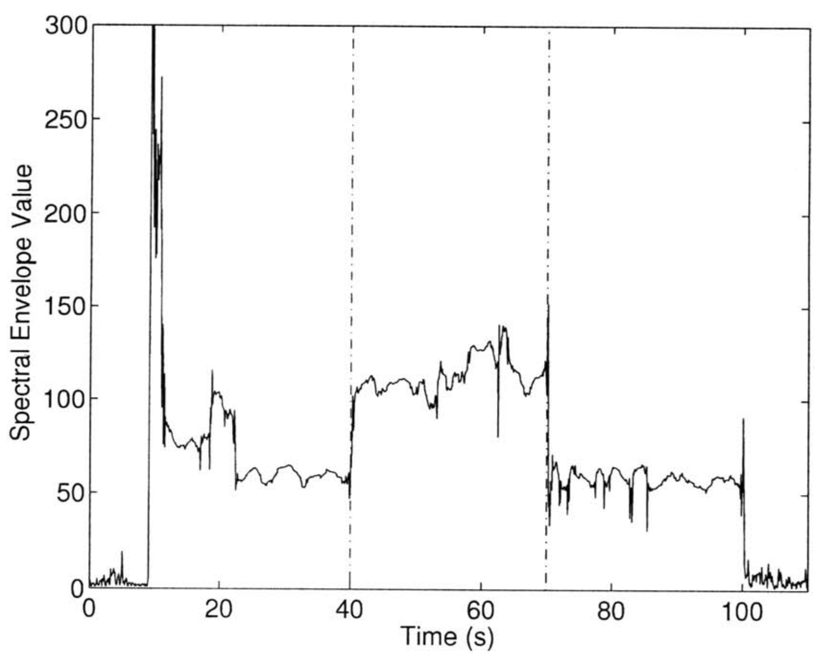


Figure 4.60: Spectral Envelope for Shower and Kitchen Sink in Overlapping Operation at 324 Hz

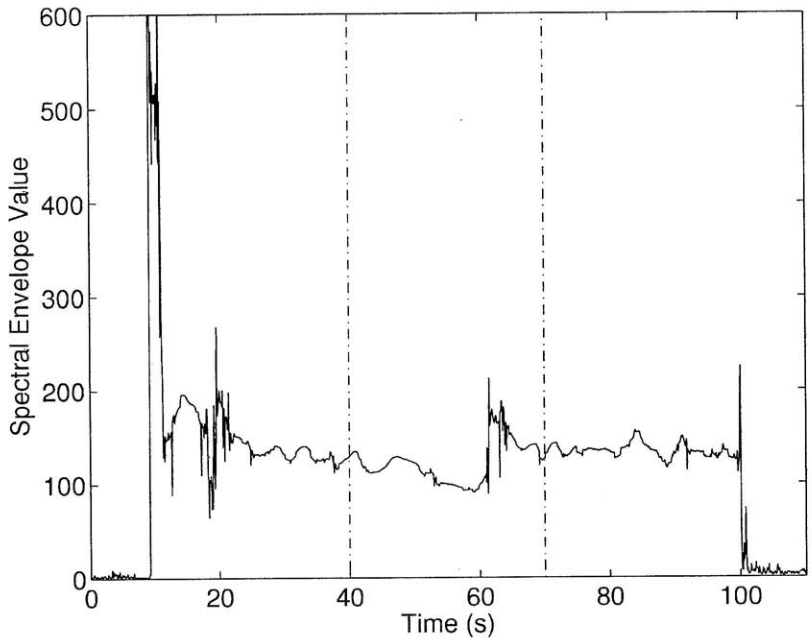


Figure 4.61: Spectral Envelope for Shower and Kitchen Sink in Overlapping Operation at 384 Hz

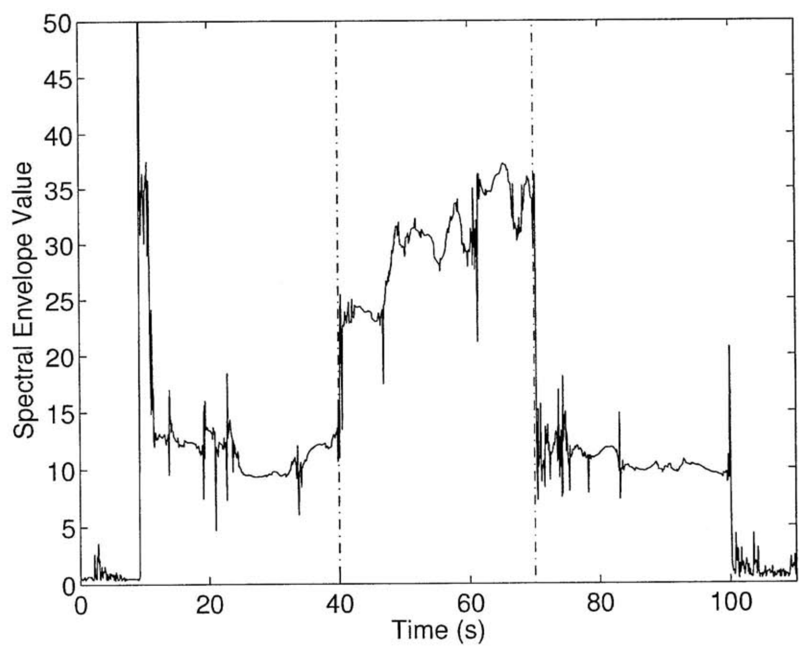


Figure 4.62: Spectral Envelope for Shower and Kitchen Sink in Overlapping Operation at 924 Hz

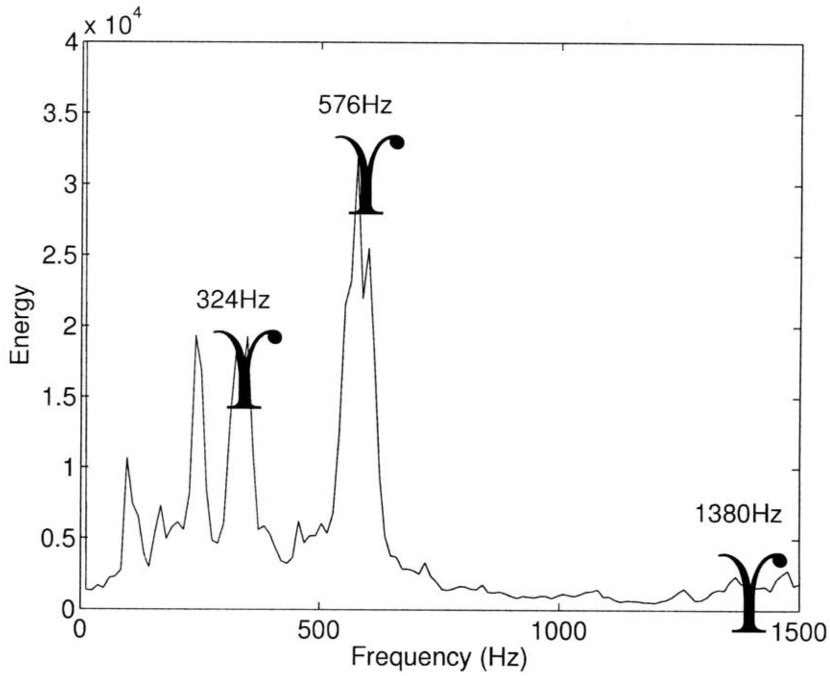


Figure 4.63: ESD for Shower and Bathroom Sink in Overlapping Operation

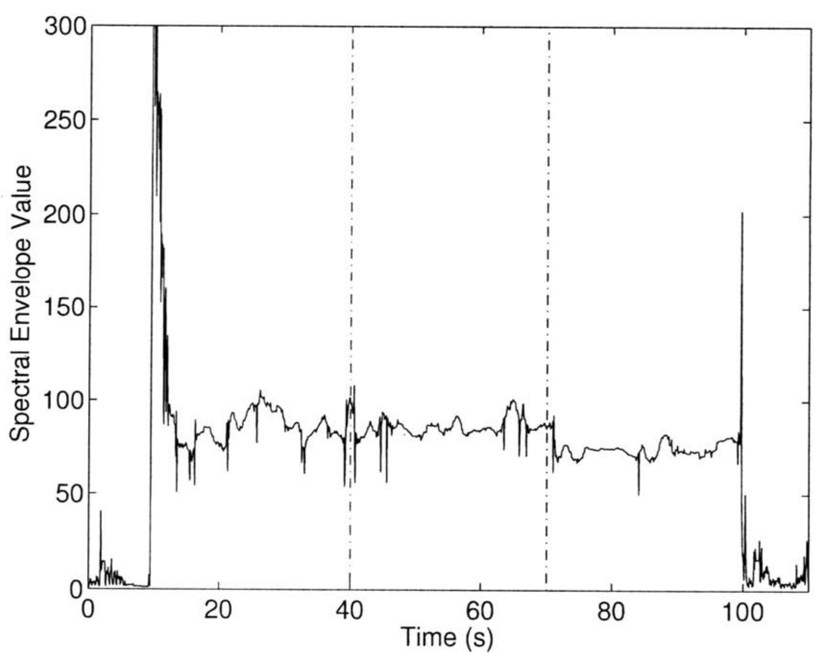


Figure 4.64: Spectral Envelope for Shower and Bathroom Sink in Overlapping Operation at 72 Hz

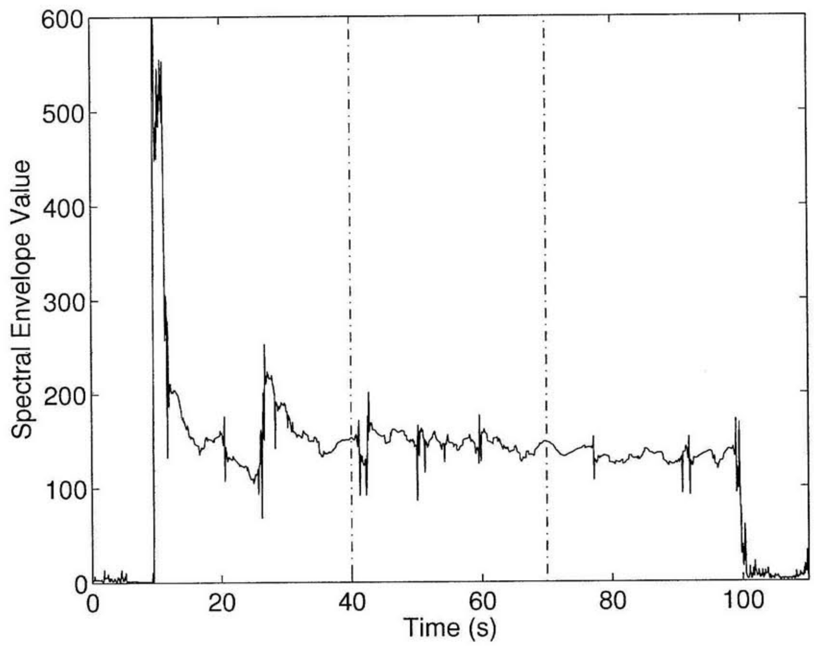


Figure 4.65: Spectral Envelope for Shower and Bathroom Sink in Overlapping Operation at 384 Hz

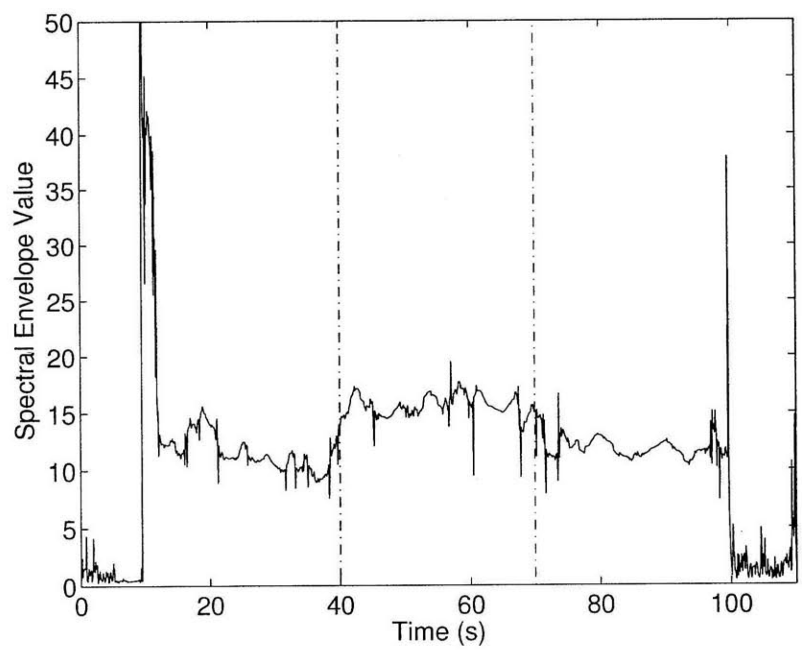


Figure 4.66: Spectral Envelope for Shower and Bathroom Sink in Overlapping Operation at 924 Hz

Chapter 5

Free Space Load Monitoring

Development work was done on a free space sensor to provide information about the operation and location of electrical loads. This sensor was an electroquasistatic (EQS) sensor to detect voltage-mode events. The free-space EQS sensor was able to detect events in a room, such as the activation of a line upon turning on a power strip or switching a light switch.

5.1 EQS Sensor Design

The EQS sensor was implemented as a voltage-mode sensor. The sensor used a pickup made of two large capacitive metal plate electrodes. The electrodes were connected with BNC cables to the circuit.

See Figure 5.1 and Figure 5.2 for schematics of the EQS sensor circuit. An instrumentation amplifier provides high impedance measurement nodes for measuring the electrode voltages. To get clean voltages on these measurement nodes, it was necessary to connect the BNC ground shielding of each electrode to a buffered version of the signal voltage, and to properly bias the measurement nodes using appropriately sized current sources. The instrumentation amplifier differential output is run through some low-pass and high-pass filters, and some level shift and gain stages. A PIC microcontroller is used to subtract off the low frequency drifting level of the signal, which leaves behind the high frequency content of interest. A programmable variable gain block is controlled by the microcontroller as well, to scale the signal. Finally, the signal is run through a low pass filter and some gain stages before terminating at a jumper.

The jumper was connected to an FPGA-based Spectral Envelope preprocessor [8]. This preprocessor was used to log the data and create the plots in the following section.

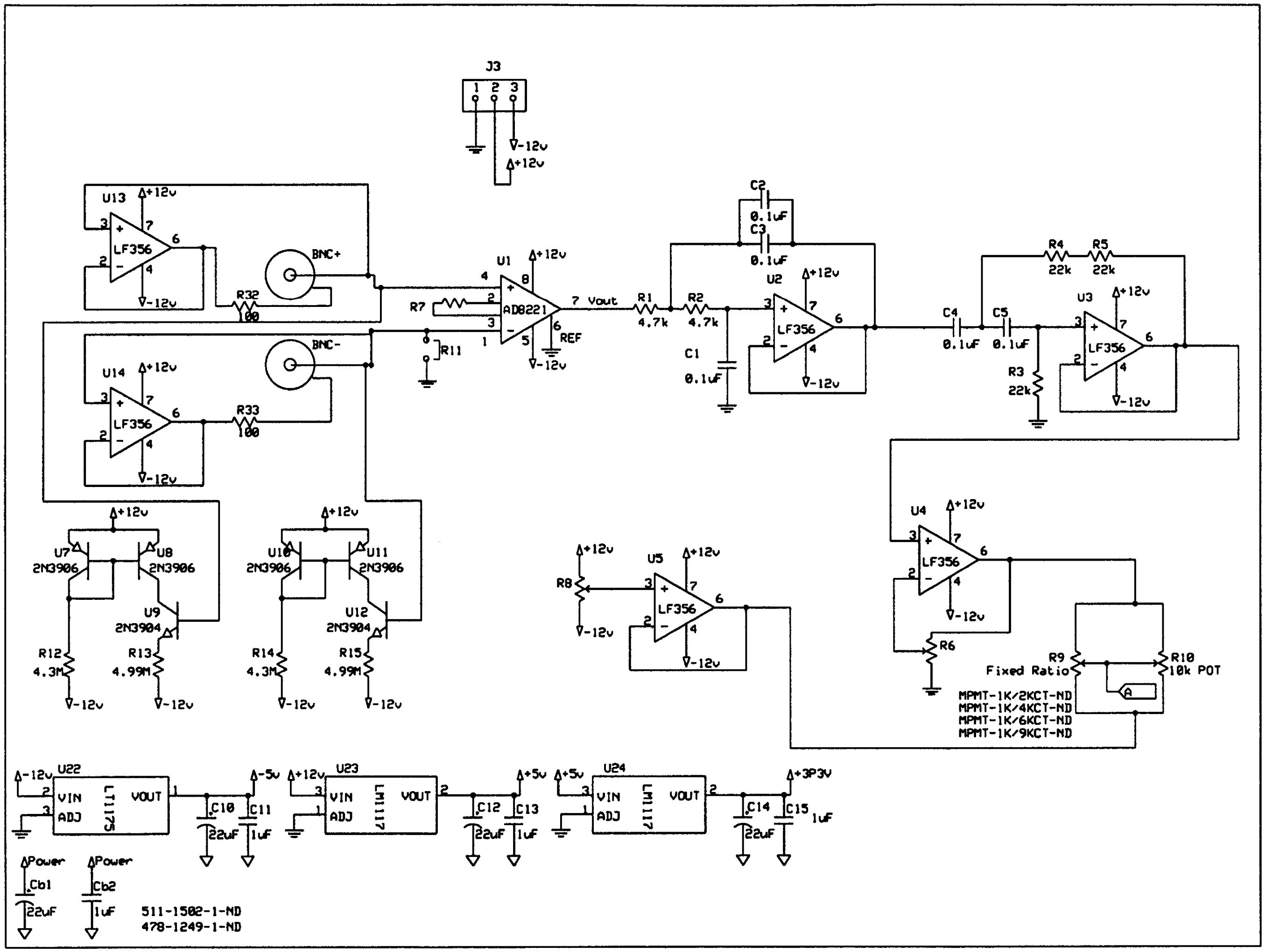


Figure5.1: EQS Sensor Schematic Page 1

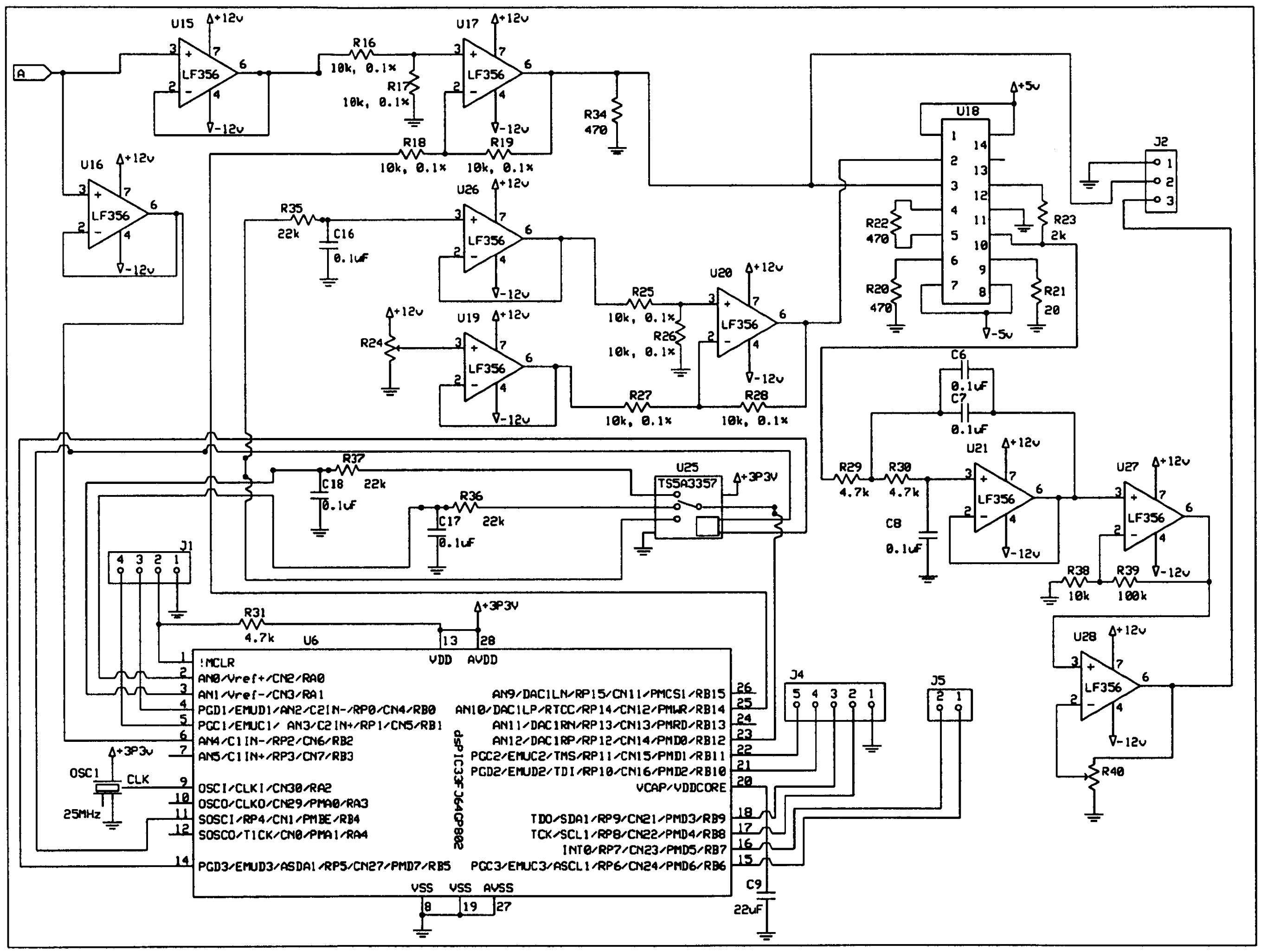


Figure5.2: EQS Sensor Schematic Page 2

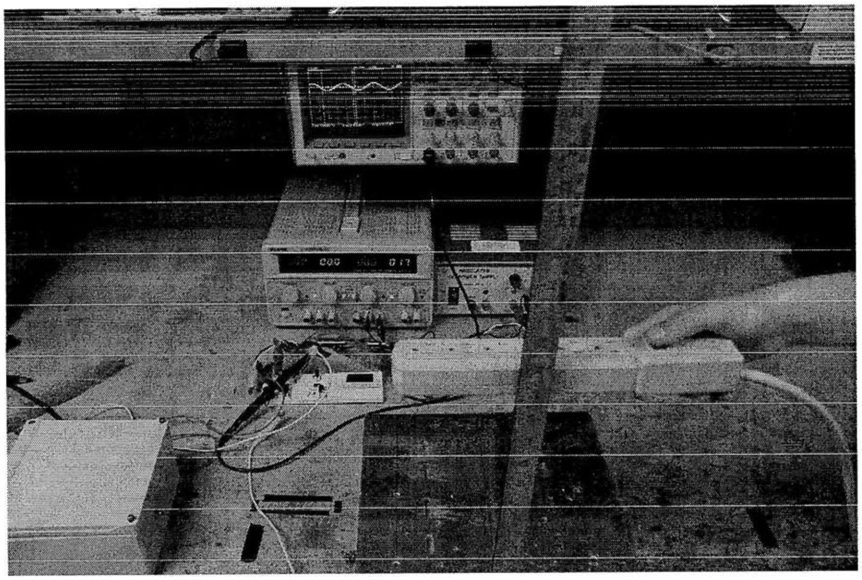


Figure 5.3: Photo of EQS Sensor Experimental Setup

5.2 EQS Sensor Experiments

Experiments were performed with a prototype of the EQS free-space sensor. Figure 5.3 and Figure 5.4 show the experimental setup used to gather data from the EQS sensor for a power strip load connected to a wall outlet. Figures 5.5 and 5.6 show Spectral Envelope data while a power strip is turning on at distances of 3 in and 12 in from the sensor electrode plate. In another experiment with the same setup, the lights in the room were switched from off to on. Figure 5.7 shows Spectral Envelope data while the room lights were turned on. The lights were approximately 59 in above the sensor electrode plate.

The data plotted in Figure 5.5, Figure 5.6, and Figure 5.7 was collected by the FPGA-based spectral envelope preprocessor. The preprocessor plots show the Spectral Envelope content of the signal at the 1st (a1) and 3rd (a3) harmonics of the 60 Hz wall electricity connected to the power strip. As seen in the power strip experiment data, the magnitude of the EQS sensor response to identical loads falls off with distance. It is this relationship between response and distance that allows the EQS sensor to help localize the activation of similar electrical loads.

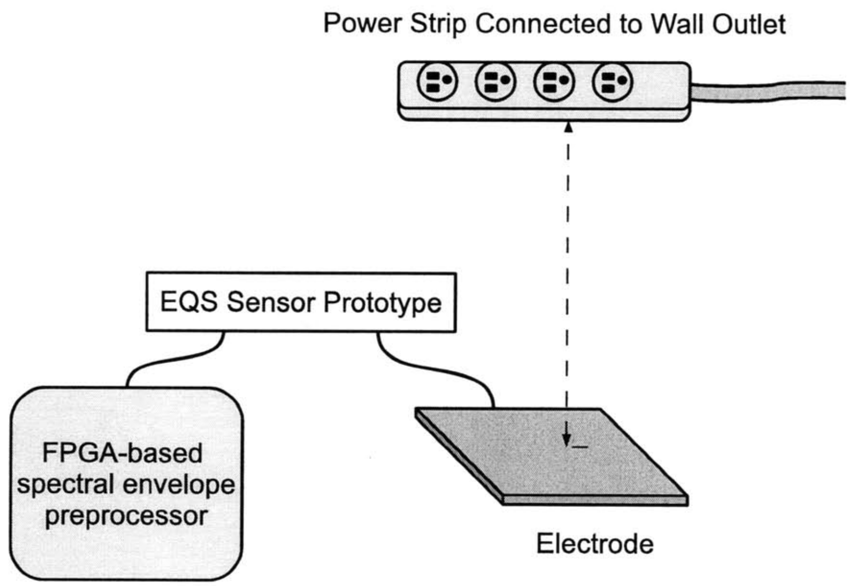


Figure 5.4: Diagram of EQS Sensor Experimental Setup

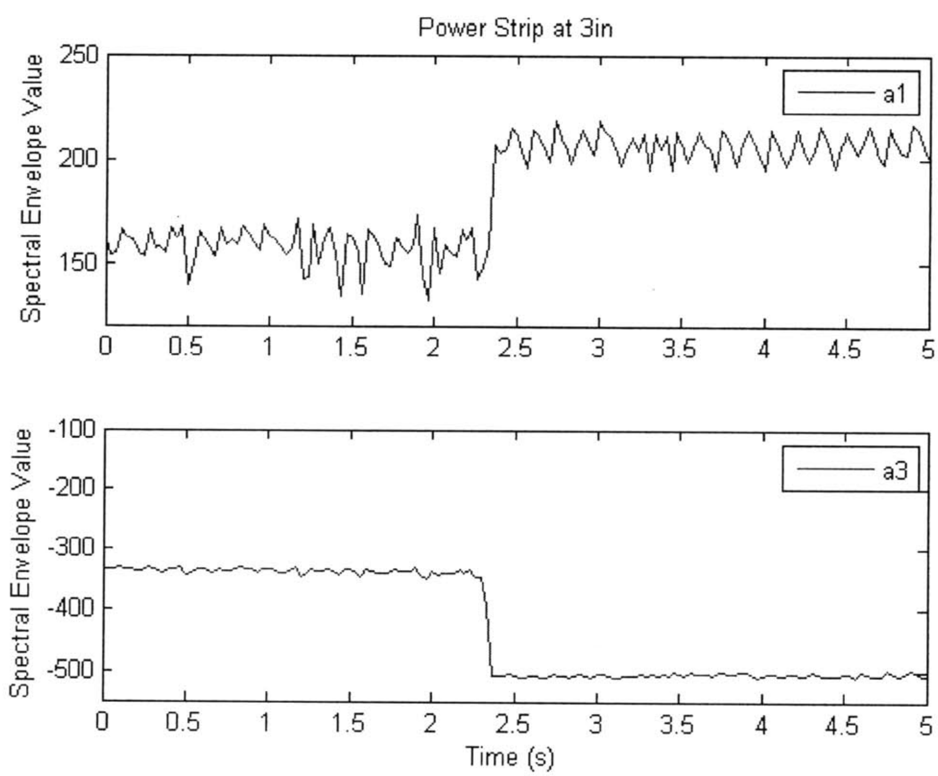


Figure 5.5: EQS Data for Power Strip Load at 3 in Range

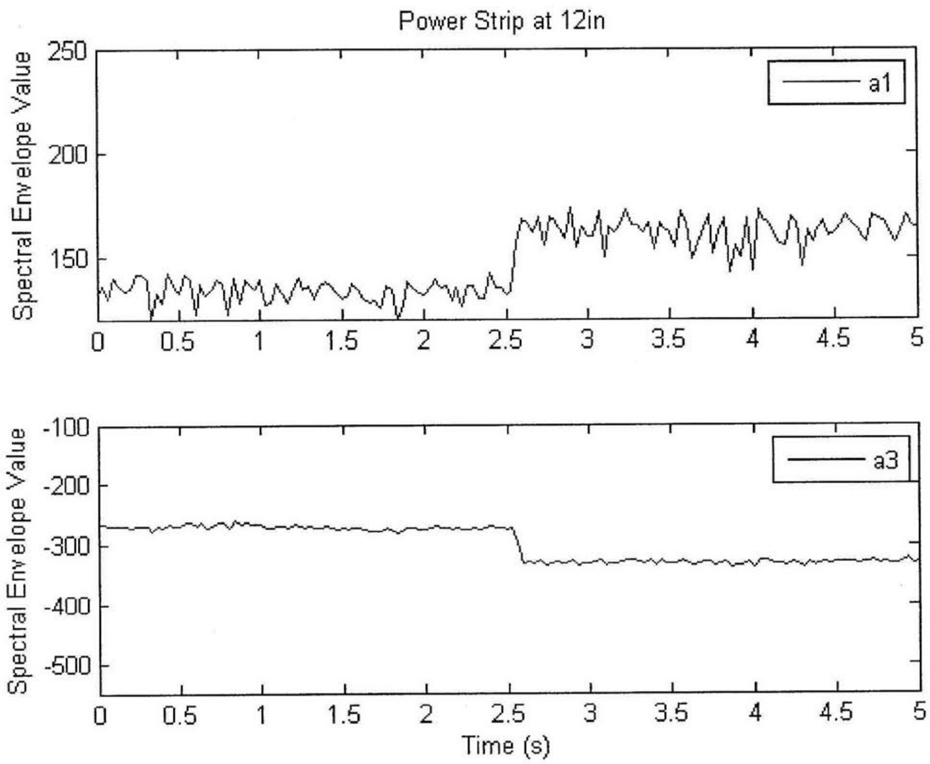


Figure 5.6: EQS Data for Power Strip Load at 12 in Range

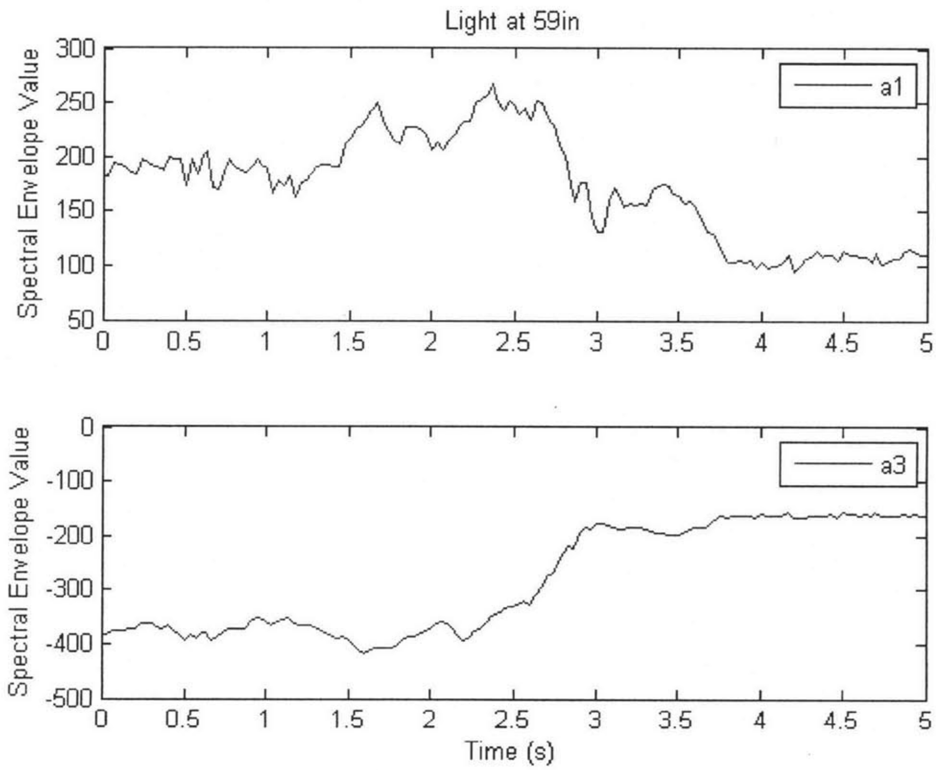


Figure 5.7: EQS Data for Room Lights Load at 59 in Range

Chapter 6

Conclusion

6.1 Contributions

This work presented a non-intrusive water utility monitoring system. The non-intrusive water utility monitoring system uses vibration measurements taken at a single point on a water utility network to identify loads on the network. Experimental data, both from a laboratory setup and from a field test site, demonstrated that loads on a water distribution network could be identified by the non-intrusive water utility monitoring system. The loads were recognizable both in individual and in overlapping operation.

The development of a free-space sensing system to supplement a power monitoring system by helping to localize the activation of loads was also presented. The free-space sensor presented was an electroquasistatic (EQS) sensor to detect voltage-mode events. Preliminary experiments with the sensor demonstrated that it was able to detect events in a room, such as the activation of a line upon turning on a power strip or switching a light switch.

6.2 Future Work

Future work that could be performed on the non-intrusive water utility monitoring system includes further analysis of the relationships between vibration frequency peak magnitudes and water pressure or flow rate. Future work could also include understanding the reasons why in experimental data of loads in overlapping operation, it appears that the characteristic peak ratios of two different loads in operation follow some rough, if nonlinear, superposition in shape. The feasibility of determining flow rate from the vibration data collected by the system could also be explored.

More experimental data could be taken with the EQS free-space sensor. The development of a magnetoquasistatic (MQS) sensor to detect magnetic field events could be explored.

Appendix A

MATLAB Code

A.1 Vibration Data Signal Processing Functions

A.1.1 ESD Calculation Function

```
% WaterNILM Signal Processing
% Sabrina Neuman (sneuman@mit.edu)
% Zack Remscrim (remscrim@mit.edu)

%------------------------------------------------------------
% Calculate ESD Function
%------------------------------------------------------------
% This function is for calculating the ESD
% for a given
%        fs              = sample freq
%        dPtr     = .WAV file
% It returns
%        time_sum           = the ESD (sum of single time window FFT's)
%        proper_freq = frequency
%        proper_time = time

function [time_sum, proper_time, proper_freq, T, P, wav_length] =
    Fxn_Calculate_ESD(fs,ptsPerWindow,dPtr);

%----------------------------------------------
% Calculate the Energy Spectral Density
     wav_length             = length(dPtr)/fs;
                                                   % length of the .WAV file , in
        seconds
    F = linspace(0,fs,ptsPerWindow);
                                         % frequencies up to the sampling
        frequency, spaced over the points per window
    T = linspace(0,wav_length*fs,wav_length*fs/ptsPerWindow);
        % time in samples over the whole .WAV, spaced by the points per
        window
    P = zeros(length(F),length(T));
                                                   % initialize matrix to store
        windowed FFTs
     for t = 1:length(T)
```

```
        P(:,t) = fft(dPtr(1+(t-1)*ptsPerWindow:t*ptsPerWindow));
            % fill matrix with windowed FFTs
    end
    time_sum     = zeros(length(F),1);
                                    % initialize vector, length of total
        frequencies
    for i = 1:length(F)
        for j = 1:length(T)
            time_sum(i) = time_sum(i) + abs(P(i,j));
                        % fill vector with sum of windowed FFT
                magnitudes, by frequency
        end
    end
    time_sum     = time_sum(1:length(F)/2);
                                    % take half for time_sum
    proper_time = linspace(0,wav_length,length(T));
                        % make properly spaced time vector
    proper_freq = F(1:length(F)/2);
                                                % make properly spaced frequency
          vector
%-----------------------------------------

end
% END Function
```

A.1.2 Spectral Envelope Calculation Function

```
% WaterNILM Signal Processing
% Sabrina Neuman (sneuman@mit.edu)
% Zack Remscrim (remscrim@mit.edu)

%-------------------------------------------------------
% Calculate and Filter Spectral Envelope Function
%-------------------------------------------------------
% This function is for calculating the ESD
% for a given
%       fs          = sample freq
%       dPtr    = .WAV file
% It returns
%       time_sum         = the ESD (sum of single time window FFT's)
%       proper_freq = frequency
%       proper_time = time

function [refiltered_1] = Fxn_Calculate_and_Filter_Spectral_Envelope(
    bandf,T,P);

%-----------------------------------------
% Calculate frequency band transient or "Spectral Envelope"
    band_1 = zeros(1,length(T));
```

```
                                  % initialize band transient vector
    f1 = bandf/12+1;
                                          % initialize band center
        for i = 1:length(T)
            for offset = -4:4
                band_1(1,i)=band_1(1,i)+abs(P(f1+offset,i));     % fill
                    band transient with sum over band
            end
            band_1(1,i)=abs(band_1(1,i));
        end
%------------------------------------
% Filtering the "Spectral Envelope" the first time
    windowSize=10;
    filtered_1=zeros(1,length(band_1));
    padded_1=horzcat(zeros(1,windowSize),band_1,zeros(1,windowSize));
    for i=1:length(filtered_1)
        lSum_1=band_1(i);
        rSum_1=band_1(i);
        for j=1:windowSize
            lSum_1=lSum_1+padded_1(windowSize+i-j);
            rSum_1=rSum_1+padded_1(windowSize+i+j);
        end
        if(abs(lSum_1-rSum_1)/(windowSize+1)>abs(band_1(i))/8)
            filtered_1(i)=band_1(i);
        else
            filtered_1(i)=(lSum_1+rSum_1-band_1(i))/(2*windowSize+1);
        end
    end
%------------------------------------
% Filtering the "Spectral Envelope" a second time
        windowSize=5;
    refiltered_1=zeros(1,length(filtered_1));
    padded_1=horzcat(zeros(1,windowSize),filtered_1 ,zeros(1,windowSize))
        ;
     for i=1:length(refiltered_1)
        lSum_1=filtered_1(i);
        rSum_1=filtered_1(i);
        for j=1:windowSize
            lSum_1=lSum_1+padded_1(windowSize+i-j);
            rSum_1=rSum_1+padded_1(windowSize+i+j);
        end
        if(abs(lSum_1-rSum_1)/(windowSize+1)>abs(filtered_1(i))/8)
            refiltered_1(i)=filtered_1(i);
        else
            refiltered_1(i)=(lSum_1+rSum_1-filtered_1(i))/(2*windowSize
                +1);
        end
     end
%------------------------------------
```

```
end
% END Function
```

A.1.3 Beam λ Calculation Function

```
% WaterNILM Pipe Freq Predictions
% Sabrina Neuman (sneuman@mit.edu)
% Zack Remscrim (remscrim@mit.edu)

%---------------------------------------------
% Pipe Frequency Prediction Lambda Calculation
%---------------------------------------------
% Lambda calculation

function result = Pipe_Freq_Pred_lambda(r,isclamped)
if (isclamped==0)
    result = r*pi;
else
    clamped = @(x) 1 - cos(x)*cosh(x);  %clamped-clamped
    counter = 1;
    numroots = 0;
    while (numroots<r)
        startpt = counter*pi;
        endpt = (startpt+pi);
            z = fzero(clamped,[startpt endpt]);
            if (isnan(z)==0)
                numroots = numroots+1;
            end
            counter = counter+1;
    end
    result = z;
end
```

A.2 Lateral Vibration in a Pipe Segment Experiments Plots

A.2.1 Lateral Vibrations Plots

```
% WaterNILM Signal Processing
% Sabrina Neuman (sneuman@mit.edu)
% Zack Remscrim (remscrim@mit.edu)

%------------------------------------------------
% Lateral Vibrations
%------------------------------------------------
% This script is for plotting the ESD of two experiments:
%       Water-Filled Pipe struck with a Hammer, Accelerometer Data
%       Water Flowing through the Pipe, Accelerometer Data
% in order to demonstrate that the preferred frequency peaks
% seen when water flows through a pipe are at the modes of the
% beam-like lateral vibration of the pipe.

%------------------------------------------------
close all
clear
%------------------------------------
for dat = 3:4;                  % which wav files to process
%------------------------------------
fs                              = 48000;        % sampling freq of
   48,000 Hz
ptsPerWindow     = 4000;        % points per FFT window
%------------------------------------------------
%------------------------------------
% Read in the data from wav files:
    switch dat
        case 1
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\
   Lateral_Vibrations_Data\Data_1_5\accel_full_mid_fixed_2' ...
);
titlestring = 'water_flow'; % general title
        case 2
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\
   Lateral_Vibrations_Data\Data_1_6\
   accel_full_mid_fixed_water_banging_5' ...
);
titlestring = 'hammer_strike'; % general title
        case 3
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\
   Lateral_Vibrations_Data\Data_1_5\accel_full_mid_fixed_2' ...
);
```

```
titlestring = 'moving_accel_mid'; % general title
        case 4
dPtr = wavread( ...
'C:\Documents and Settings\sneuman\Desktop\MEng\MEng Thesis\MATLAB\
   Lateral Vibrations Data\Data_1_5\accel_full_third_fixed_1' ...
);
titlestring = 'moving_accel_fraction'; % general title
    end
%------------------------------------------
% Calculate the Energy Spectral Density
[time_sum, proper_time, proper_freq, T, P, wav_length] =
   Fxn_Calculate_ESD(fs,ptsPerWindow,dPtr);
%------------------------------------------
% Choose ESD plot axis
            freq1 = 72.02;
            freq2 = 384.1;
            freq3 = 935;
                        ymin = 0;
            ymax = 1000;
            label72 = '72Hz \newline \newline \newline';
            circl72 = 0;
            label384 = '384Hz \newline \newline \newline';
            circl384 = 0;
            label924 = '~924Hz \newline \newline \newline \newline';
            circl924 = 0;
        switch dat
        case 1
            freq1 = 72.02;
            freq2 = 384.1;
            freq3 = 935;
            ymin = 90;
            ymax = 500;
            label72 = '     72Hz \newline \newline \newline \newline';
            circl72 = -10;
            label384 = '384Hz \newline \newline \newline \newline';
            circl384 = -10;
            label924 = '~924Hz       \newline \newline \newline \newline 
               \newline \newline \newline \newline \newline \newline \
               newline \newline \newline \newline \newline \newline';
            circl924 = 150;
        case 2
            freq1 = 72.02;
            freq2 = 384.1;
            freq3 = 935;
            ymin = 0;
            ymax = 2300;
            label72 = '               72Hz \newline \newline';
            circl72 = -80;
            label384 = '384Hz \newline \newline \newline \newline';
            circl384 = -80;
```

```
            label924 = '~924Hz␣␣␣␣␣␣\newline␣\newline␣\newline␣\newline␣\newline␣\newline␣\newline';
            circl924 = 100;
        case 3
            freq1 = 72.02;
            freq2 = 384.1;
            freq3 = 935;
            ymin = 0;
            ymax = 500;
            label72 = '␣␣␣␣␣1st␣\newline␣\newline␣\newline';
            circl72 = -10;
            label384 = '3rd␣\newline␣\newline␣\newline';
            circl384 = -10;
            label924 = '5th␣␣␣␣␣␣\newline␣\newline␣\newline␣\newline␣\newline␣\newline␣\newline␣\newline␣\newline␣\newline␣\newline␣\newline␣\newline';
            circl924 = 150;
        case 4
            freq1 = 204;
            freq2 = 636;
            freq3 = 960;
            ymin = 0;
            ymax = 800;
            label72 = '␣␣␣␣␣2nd␣\newline␣\newline␣\newline␣\newline␣\newline␣\newline';
            circl72 = -20;
            label384 = '4th␣\newline␣\newline␣\newline␣\newline␣\newline␣\newline␣\newline␣\newline␣\newline␣\newline␣\newline␣\newline␣\newline␣\newline␣\newline␣\newline␣\newline';
            circl384 = -20;
            label924 = '5th␣␣␣␣␣␣\newline␣\newline␣\newline␣\newline␣\newline␣\newline␣\newline␣\newline␣\newline␣\newline␣\newline';
            circl924 = 200;
        end
%--------------------------------------------
% Plot the Energy Spectral Density
    figure
        plot(proper_freq,time_sum)
    % mark 72 Hz
    text(freq1,time_sum(7),label72,'FontSize',16,'HorizontalAlignment','center')
    %text(freq1,circl72+time_sum(7),'\circ','FontSize',72,'HorizontalAlignment','center')
    % mark 384 Hz
    text(freq2,time_sum(33),label384,'FontSize',16,'HorizontalAlignment','center')
    %text(freq2,circl384+time_sum(33),'\circ','FontSize',72,'HorizontalAlignment','center')
    % mark 924 Hz
```

```
    text(freq3 ,time_sum(78) ,label924 , 'FontSize' ,16, 'HorizontalAlignment'
        , 'center')
    %text(freq3, circl924+time_sum(78), '\circ', 'FontSize',72, '
        HorizontalAlignment', 'center')
    set(gca, 'FontSize',16)
        xlim([0 1000])
    ylim([ymin ymax])
    titlestring_ESD = horzcat('ESD_for_', titlestring);
    %title(titlestring_ESD, 'FontSize',16);
    set(gcf, 'Name', titlestring_ESD);
    xlabel('Frequency_(Hz)', 'FontSize',16);
    ylabel('Energy', 'FontSize',16);
    %saveas(gcf,['ESD for ', titlestring, '.bmp'])
    saveas(gcf,['ch4_', titlestring, '_ESD', '.eps'])
%------------------------------------------------
%---------------------------------------------------------
end % End of for loop
%---------------------------------------------------------
%------------------------------------------------
% Plot the Expected Fixed-Fixed Pipe Mode Locations
% CLAMPED/FIXED
x=[0:0.001:1];
figure
lam1 = Pipe_Freq_Pred_lambda(1,1)
lam2 = Pipe_Freq_Pred_lambda(2,1)
lam3 = Pipe_Freq_Pred_lambda(3,1)
lam4 = Pipe_Freq_Pred_lambda(4,1)
lam5 = Pipe_Freq_Pred_lambda(5,1)
plot(x,-(((sin(lam1)+sinh(lam1))/(cos(lam1)-cosh(lam1))).*(sin(lam1*x)-
    sinh(lam1*x))+cos(lam1*x)-cosh(lam1*x)) ...
    ,'b-', ...
     x,-(((sin(lam2)+sinh(lam2))/(cos(lam2)-cosh(lam2))).*(sin(lam2*x)-
        sinh(lam2*x))+cos(lam2*x)-cosh(lam2*x)) ...
    ,'b--', ...
     x,-(((sin(lam3)+sinh(lam3))/(cos(lam3)-cosh(lam3))).*(sin(lam3*x)-
        sinh(lam3*x))+cos(lam3*x)-cosh(lam3*x)) ...
    ,'b-.', ...
     x,-(((sin(lam4)+sinh(lam4))/(cos(lam4)-cosh(lam4))).*(sin(lam4*x)-
        sinh(lam4*x))+cos(lam4*x)-cosh(lam4*x)) ...
    ,'b:', ...
     x,-(((sin(lam5)+sinh(lam5))/(cos(lam5)-cosh(lam5))).*(sin(lam5*x)-
        sinh(lam5*x))+cos(lam5*x)-cosh(lam5*x)) ...
    ,'b-o')
%line([(1/3) ; (1/3)],[-2 ; 2], 'LineStyle', '-.')
%line([(1/5) ; (1/5)],[-2 ; 2], 'LineStyle', '-.')
set(gca, 'FontSize',16)
legend('1st_Mode', '2nd_Mode', '3rd_Mode', '4th_Mode', '5th_Mode', 'FontSize'
    ,14, 'Location', 'NorthEast')
%title('Fixed-Fixed Modes', 'FontSize',16);
set(gcf, 'Name', 'Fixed-Fixed_Modes');
```

```
xlabel('Length','FontSize',16);
ylabel('Amplitude','FontSize',16);
saveas(gcf,horzcat('ch2_fixed_fixed_modes','.eps'))
%--------------------------------------------
```

A.3 Load Pressure Frequency Coloring Experiments Plots

A.3.1 Background Pressure Plots

```
% WaterNILM Signal Processing
% Sabrina Neuman (sneuman@mit.edu)
% Zack Remscrim (remscrim@mit.edu)

%
% Background Pressure from Source Sink
%
% This script is for plotting the ESD for
%         Directly from Sink, Pressure Data
%         Short Garden Hose, Pressure Data
%         Long Garden Hose, Pressure Data
% to determine the background pressure variations that the
% source sink produces.

%
close all
clear
%
% Initialize vectors for averaging
ESDforPressSinkDirect        = zeros(1,2000)';    % initialize
ESDforPressSinkShortHose     = zeros(1,2000)';    % initialize
ESDforPressSinkLongHose      = zeros(1,2000)';    % initialize
%
for dat = 1:9;                       % which wav files to process
%
fs                                   = 48000;          % sampling freq of
   48,000 Hz
ptsPerWindow      = 4000;            % points per FFT window
%
%
% Read in the data from wav files:
    switch dat
        case 1
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\
   Frequency_Coloring_Data\Data_3_24\pressure_directly_from_sink_30s_1'
   ...
    , [1 (30*fs+1)]);
titlestring = 'Pressure_From_Sink_Directly_1'; % general title
        case 2
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\
   Frequency_Coloring_Data\Data_3_24\pressure_directly_from_sink_30s_2'
   ...
    , [1 (30*fs+1)]);
```

```
titlestring = 'Pressure␣From␣Sink␣Directly␣2'; % general title
        case 3
dPtr = wavread( ...
'C:\Documents␣and␣Settings\sneuman\Desktop\MEng\MEng␣Thesis\MATLAB\
   Frequency␣Coloring␣Data\Data_3_24\pressure_directly_from_sink_30s_3'
   ...
    , [1 (30*fs+1)]);
titlestring = 'Pressure␣From␣Sink␣Directly␣3'; % general title
        case 4
dPtr = wavread( ...
'C:\Documents␣and␣Settings\sneuman\Desktop\MEng\MEng␣Thesis\MATLAB\
   Frequency␣Coloring␣Data\Data_3_24\pressure_directly_short_hose_30s_1
   '...
    , [1 (30*fs+1)]);
titlestring = 'Pressure␣From␣Sink␣Short␣Hose␣1'; % general title
        case 5
dPtr = wavread( ...
'C:\Documents␣and␣Settings\sneuman\Desktop\MEng\MEng␣Thesis\MATLAB\
   Frequency␣Coloring␣Data\Data_3_24\pressure_directly_short_hose_30s_2
   '...
    , [1 (30*fs+1)]);
titlestring = 'Pressure␣From␣Sink␣Short␣Hose␣2'; % general title
        case 6
dPtr = wavread( ...
'C:\Documents␣and␣Settings\sneuman\Desktop\MEng\MEng␣Thesis\MATLAB\
   Frequency␣Coloring␣Data\Data_3_24\pressure_directly_short_hose_30s_3
   '...
    , [1 (30*fs+1)]);
titlestring = 'Pressure␣From␣Sink␣Short␣Hose␣3'; % general title
        case 7
dPtr = wavread( ...
'C:\Documents␣and␣Settings\sneuman\Desktop\MEng\MEng␣Thesis\MATLAB\
   Frequency␣Coloring␣Data\Data_3_24\pressure_directly_long_hose_30s_1'
   ...
    , [1 (30*fs+1)]);
titlestring = 'Pressure␣From␣Sink␣Long␣Hose␣1'; % general title
        case 8
dPtr = wavread( ...
'C:\Documents␣and␣Settings\sneuman\Desktop\MEng\MEng␣Thesis\MATLAB\
   Frequency␣Coloring␣Data\Data_3_24\pressure_directly_long_hose_30s_2'
   ...
    , [1 (30*fs+1)]);
titlestring = 'Pressure␣From␣Sink␣Long␣Hose␣2'; % general title
        case 9
dPtr = wavread( ...
'C:\Documents␣and␣Settings\sneuman\Desktop\MEng\MEng␣Thesis\MATLAB\
   Frequency␣Coloring␣Data\Data_3_24\pressure_directly_long_hose_30s_3'
   ...
    , [1 (30*fs+1)]);
titlestring = 'Pressure␣From␣Sink␣Long␣Hose␣3'; % general title
```

```
    end
%------------------------------------------
% Calculate the Energy Spectral Density
[time_sum, proper_time, proper_freq, T, P, wav_length] =
   Fxn_Calculate_ESD(fs,ptsPerWindow,dPtr);
%------------------------------------------
% Take sums of ESD's, for averaging
    if      ((dat>0)&&(dat<4))
        ESDforPressSinkDirect       = ESDforPressSinkDirect+time_sum;
    elseif ((dat>3)&&(dat<7))
        ESDforPressSinkShortHose    = ESDforPressSinkShortHose+time_sum;
    elseif ((dat>6)&&(dat<10))
        ESDforPressSinkLongHose     = ESDforPressSinkLongHose+time_sum;
    end
%------------------------------------------
%-----------------------------------------------------
end % End of for loop
%-----------------------------------------------------
%------------------------------------------
% Plot all averaged pressure ESD's together
    figure
        plot(     proper_freq,(ESDforPressSinkDirect./3),'b-',...
              proper_freq,(ESDforPressSinkShortHose./3),'b--',...
              proper_freq,(ESDforPressSinkLongHose./3),'b-.')
        set(gca,'FontSize',16)
    legend('Pressure Sink Directly','Pressure Sink Short Hose','Pressure
       Sink Long Hose','Location','NorthEast','FontSize',16)
    axis([0 1000 0 425])
    %title('Avg ESDs for Different Background Source Pressure Readings
       ','FontSize',16);
    set(gcf, 'Name', 'Avg ESDs');
    xlabel('Frequency (Hz)','FontSize',16);
    ylabel('Log(Energy)','FontSize',16);
    %saveas(gcf,horzcat('Avg ESDs for Different Background Source
       Pressure Readings','.bmp'))
    saveas(gcf,horzcat('ch4_background_pressure_ESD','.eps'))
%------------------------------------------
```

A.3.2 Variable Load on Hose Plots

```
% WaterNILM Signal Processing
% Sabrina Neuman (sneuman@mit.edu)
% Zack Remscrim (remscrim@mit.edu)

%-----------------------------------------------------
% Different Loads on Short Hose Pressure
%-----------------------------------------------------
% This script is for plotting the ESD for
%        Big Hole on Garden Sprayer on Short Garden Hose, Pressure Data
```

```
%       Medium Hole on Garden Sprayer on Short Garden Hose, Pressure
   Data
%       Small Hole on Garden Sprayer on Short Garden Hose, Pressure Data
%       No Hole on Garden Sprayer on Short Garden Hose, Pressure Data
% to determine the typical pressure variations for a
% variable aperture load on a fixed length of hose.

%------------------------------------------------
close all
clear
%------------------------------------------------
% Initialize vectors for averaging
ESDforPressBigHole      = zeros(1,2000)';   % initialize
ESDforPressMedHole      = zeros(1,2000)';   % initialize
ESDforPressSmallHole    = zeros(1,2000)';   % initialize
ESDforPressNoHole       = zeros(1,2000)';   % initialize
%------------------------------------------------
for dat = 1:12;                    % which wav files to process
%------------------------------------------------
fs                                 = 48000;      % sampling freq of
   48,000 Hz
ptsPerWindow     = 4000;           % points per FFT window
%------------------------------------------------
%------------------------------------------------
% Read in the data from wav files:
    switch dat
        case 1
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\
   Frequency_Coloring_Data\Data_3_25\pressure_big_hole_30s_no_pipe_1'
   ...
    , [1 (30*fs+1)]);
titlestring = 'Pressure_Big_Hole_1'; % general title
        case 2
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\
   Frequency_Coloring_Data\Data_3_25\pressure_big_hole_30s_no_pipe_2'
   ...
    , [1 (30*fs+1)]);
titlestring = 'Pressure_Big_Hole_2'; % general title
        case 3
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\
   Frequency_Coloring_Data\Data_3_25\pressure_big_hole_30s_no_pipe_3'
   ...
    , [1 (30*fs+1)]);
titlestring = 'Pressure_Big_Hole_3'; % general title
%------------------------------------------------
        case 4
dPtr = wavread( ...
```

```
'C:\Documents and Settings\sneuman\Desktop\MEng\MEng Thesis\MATLAB\
    Frequency Coloring Data\Data_3_25\pressure_med_hole_30s_no_pipe_1'
    ...
    , [1 (30*fs+1)]);
titlestring = 'Pressure Med Hole 1'; % general title
        case 5
dPtr = wavread( ...
'C:\Documents and Settings\sneuman\Desktop\MEng\MEng Thesis\MATLAB\
    Frequency Coloring Data\Data_3_25\pressure_med_hole_30s_no_pipe_2'
    ...
    , [1 (30*fs+1)]);
titlestring = 'Pressure Med Hole 2'; % general title
        case 6
dPtr = wavread( ...
'C:\Documents and Settings\sneuman\Desktop\MEng\MEng Thesis\MATLAB\
    Frequency Coloring Data\Data_3_25\pressure_med_hole_30s_no_pipe_3'
    ...
    , [1 (30*fs+1)]);
titlestring = 'Pressure Med Hole 3'; % general title
%-----------------------------------------------------------------------
        case 7
dPtr = wavread( ...
'C:\Documents and Settings\sneuman\Desktop\MEng\MEng Thesis\MATLAB\
    Frequency Coloring Data\Data_3_25\pressure_small_hole_30s_no_pipe_1'
    ...
    , [1 (30*fs+1)]);
titlestring = 'Pressure Small Hole 1'; % general title
        case 8
dPtr = wavread( ...
'C:\Documents and Settings\sneuman\Desktop\MEng\MEng Thesis\MATLAB\
    Frequency Coloring Data\Data_3_25\pressure_small_hole_30s_no_pipe_2'
    ...
    , [1 (30*fs+1)]);
titlestring = 'Pressure Small Hole 2'; % general title
        case 9
dPtr = wavread( ...
'C:\Documents and Settings\sneuman\Desktop\MEng\MEng Thesis\MATLAB\
    Frequency Coloring Data\Data_3_25\pressure_small_hole_30s_no_pipe_3'
    ...
    , [1 (30*fs+1)]);
titlestring = 'Pressure Small Hole 3'; % general title
%-----------------------------------------------------------------------
        case 10
dPtr = wavread( ...
'C:\Documents and Settings\sneuman\Desktop\MEng\MEng Thesis\MATLAB\
    Frequency Coloring Data\Data_3_25\pressure_no_hole_30s_no_pipe_1'...
    , [1 (30*fs+1)]);
titlestring = 'Pressure No Hole 1'; % general title
        case 11
dPtr = wavread( ...
```

```
'C:\Documents and Settings\sneuman\Desktop\MEng\MEng Thesis\MATLAB\
   Frequency Coloring Data\Data_3_25\pressure_no_hole_30s_no_pipe_2'...
   , [1 (30*fs+1)]);
titlestring = 'Pressure No Hole 2'; % general title
        case 12
dPtr = wavread( ...
'C:\Documents and Settings\sneuman\Desktop\MEng\MEng Thesis\MATLAB\
   Frequency Coloring Data\Data_3_25\pressure_no_hole_30s_no_pipe_3'...
   , [1 (30*fs+1)]);
titlestring = 'Pressure No Hole 3'; % general title
    end
%------------------------------------------------------------
% Calculate the Energy Spectral Density
[time_sum, proper_time, proper_freq, T, P, wav_length] =
   Fxn_Calculate_ESD(fs,ptsPerWindow,dPtr);
%------------------------------------------------------------
% Take sums of ESD's, for averaging
    if      ((dat>0)&&(dat<4))
        ESDforPressBigHole      = ESDforPressBigHole+time_sum;
    elseif  ((dat>3)&&(dat<7))
        ESDforPressMedHole      = ESDforPressMedHole+time_sum;
    elseif  ((dat>6)&&(dat<10))
        ESDforPressSmallHole    = ESDforPressSmallHole+time_sum;
    elseif  ((dat>9)&&(dat<13))
        ESDforPressNoHole       = ESDforPressNoHole+time_sum;
    end
%------------------------------------------------------------
%--------------------------------------------------------------------------
end % End of for loop
%--------------------------------------------------------------------------
%------------------------------------------------------------
% Plot all averaged pressure ESD's together
        figure
        plot(      proper_freq,(ESDforPressBigHole./3),'b-',...
             proper_freq,(ESDforPressMedHole./3),'b--',...
             proper_freq,(ESDforPressSmallHole./3),'b-.',...
             proper_freq,(ESDforPressNoHole./3),'b:')
    set(gca,'FontSize',16)
    legend('Pressure Big Hole','Pressure Medium Hole','Pressure Small
       Hole','Pressure No Hole','Location','NorthEast','FontSize',16)
    axis([0 1000 0 100])
    %title('Avg ESDs for Different Loads on a Hose Pressure Readings','
       FontSize',16);
    set(gcf, 'Name', 'Avg ESDs');
    xlabel('Frequency (Hz)','FontSize',16);
    ylabel('Energy','FontSize',16);
    %saveas(gcf,horzcat('Avg ESDs for Different Loads on a Hose Pressure
         Readings','.bmp'))
    saveas(gcf,horzcat('ch4_different_loads_hose_ESD','.eps'))
%------------------------------------------------------------
```

A.3.3 Variable Load on Hose and Pipe Plots

```
% WaterNILM Signal Processing
% Sabrina Neuman (sneuman@mit.edu)
% Zack Remscrim (remscrim@mit.edu)

%---------------------------------------------------------
% Different Loads on Hose and Pipe Pressure
%---------------------------------------------------------
% This script is for plotting the ESD for
%       Big Hole on Garden Sprayer on Hose and Pipe, Accelerometer and
    Pressure Data
%       Medium Hole on Garden Sprayer on Hose and Pipe, Accelerometer
    and Pressure Data
%       Small Hole on Garden Sprayer on Hose and Pipe, Accelerometer and
     Pressure Data
%       No Hole on Garden Sprayer on Hose and Pipe, Accelerometer and
    Pressure Data
% to determine the typical pressure variations and
% vibration data for a variable aperture load connected to a fixed-fixed
% pipe segment.

%---------------------------------------------------------
close all
clear
%------------------------------------
% Initialize vectors for averaging
ESDforPressBigHole      = zeros(1,2000)';    % initialize
ESDforPressMedHole      = zeros(1,2000)';    % initialize
ESDforPressSmallHole    = zeros(1,2000)';    % initialize
ESDforPressNoHole       = zeros(1,2000)';    % initialize
%------------------------------------
ESDforAccelBigHole      = zeros(1,2000)';    % initialize
ESDforAccelMedHole      = zeros(1,2000)';    % initialize
ESDforAccelSmallHole    = zeros(1,2000)';    % initialize
ESDforAccelNoHole       = zeros(1,2000)';    % initialize
%------------------------------------
for dat = 1:16;                     % which wav files to process
%------------------------------------
fs                                  = 48000;        % sampling freq of
    48,000 Hz
ptsPerWindow      = 4000;           % points per FFT window
%---------------------------------------------------------
%------------------------------------
% Read in the data from wav files:
    switch dat
        case 1
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\
    Frequency_Coloring_Data\data_3_31_press\big_hole_1'...
```

```
    , [1 (30*fs+1)]);
titlestring = 'Pressure_Big_Hole_1'; % general title
        case 2
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\
   Frequency_Coloring_Data\data_3_31_press\big_hole_2'...
    , [1 (30*fs+1)]);
titlestring = 'Pressure_Big_Hole_2'; % general title
%---------------------------------------------
        case 3
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\
   Frequency_Coloring_Data\data_3_31_press\med_hole_1'...
    , [1 (30*fs+1)]);
titlestring = 'Pressure_Med_Hole_1'; % general title
        case 4
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\
   Frequency_Coloring_Data\data_3_31_press\med_hole_2'...
    , [1 (30*fs+1)]);
titlestring = 'Pressure_Med_Hole_2'; % general title
%---------------------------------------------
        case 5
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\
   Frequency_Coloring_Data\data_3_31_press\small_hole_1'...
    , [1 (30*fs+1)]);
titlestring = 'Pressure_Small_Hole_1'; % general title
        case 6
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\
   Frequency_Coloring_Data\data_3_31_press\small_hole_2'...
    , [1 (30*fs+1)]);
titlestring = 'Pressure_Small_Hole_2'; % general title
%---------------------------------------------
        case 7
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\
   Frequency_Coloring_Data\data_3_31_press\no_hole_1'...
    , [1 (30*fs+1)]);
titlestring = 'Pressure_No_Hole_1'; % general title
        case 8
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\
   Frequency_Coloring_Data\data_3_31_press\no_hole_2'...
    , [1 (30*fs+1)]);
titlestring = 'Pressure_No_Hole_2'; % general title
%---------------------------------------------
%---------------------------------------------
        case 9
```

```
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\
    Frequency_Coloring_Data\Data_3_31_accel\big_hole_1'...
    , [1 (30*fs+1)]);
titlestring = 'Accel_Big_Hole_1'; % general title
        case 10
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\
    Frequency_Coloring_Data\Data_3_31_accel\big_hole_2'...
    , [1 (30*fs+1)]);
titlestring = 'Accel_Big_Hole_2'; % general title
%-----------------------------------------------------------
        case 11
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\
    Frequency_Coloring_Data\Data_3_31_accel\med_hole_1'...
    , [1 (30*fs+1)]);
titlestring = 'Accel_Med_Hole_1'; % general title
        case 12
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\
    Frequency_Coloring_Data\Data_3_31_accel\med_hole_2'...
    , [1 (30*fs+1)]);
titlestring = 'Accel_Med_Hole_2'; % general title
%-----------------------------------------------------------
        case 13
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\
    Frequency_Coloring_Data\Data_3_31_accel\small_hole_1'...
    , [1 (30*fs+1)]);
titlestring = 'Accel_Small_Hole_1'; % general title
        case 14
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\
    Frequency_Coloring_Data\Data_3_31_accel\small_hole_2'...
    , [1 (30*fs+1)]);
titlestring = 'Accel_Small_Hole_2'; % general title
%-----------------------------------------------------------
        case 15
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\
    Frequency_Coloring_Data\Data_3_31_accel\no_hole_1'...
    , [1 (30*fs+1)]);
titlestring = 'Accel_No_Hole_1'; % general title
        case 16
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\
    Frequency_Coloring_Data\Data_3_31_accel\no_hole_2'...
    , [1 (30*fs+1)]);
titlestring = 'Accel_No_Hole_2'; % general title
```

```
    end
%--------------------------------------------
% Calculate the Energy Spectral Density
[time_sum, proper_time, proper_freq, T, P, wav_length] =
   Fxn_Calculate_ESD(fs,ptsPerWindow,dPtr);
%--------------------------------------------
% Take sums of ESD's, for averaging
    if      ((dat>0)&&(dat<3))
        ESDforPressBigHole      = ESDforPressBigHole+time_sum;
    elseif  ((dat>2)&&(dat<5))
        ESDforPressMedHole      = ESDforPressMedHole+time_sum;
    elseif  ((dat>4)&&(dat<7))
        ESDforPressSmallHole    = ESDforPressSmallHole+time_sum;
    elseif  ((dat>6)&&(dat<9))
        ESDforPressNoHole       = ESDforPressNoHole+time_sum;
    elseif  ((dat>8)&&(dat<11))
        ESDforAccelBigHole      = ESDforAccelBigHole+time_sum;
    elseif  ((dat>10)&&(dat<13))
        ESDforAccelMedHole      = ESDforAccelMedHole+time_sum;
    elseif  ((dat>12)&&(dat<15))
        ESDforAccelSmallHole    = ESDforAccelSmallHole+time_sum;
    elseif  ((dat>14)&&(dat<17))
        ESDforAccelNoHole       = ESDforAccelNoHole+time_sum;
    end
%--------------------------------------------
%------------------------------------------------------
end % End of for loop
%------------------------------------------------------
%--------------------------------------------
% Plot all averaged pressure ESD's together
    figure
        plot(...
            proper_freq,(ESDforPressBigHole./2),'b-',...
            proper_freq,(ESDforPressMedHole./2),'b--',...
            proper_freq,(ESDforPressSmallHole./2),'b-.',...
            proper_freq,(ESDforPressNoHole./2),'b:')
    set(gca,'FontSize',16)
    legend('Pressure Big Hole','Pressure Medium Hole','Pressure Small 
       Hole','Pressure No Hole','Location','NorthEast','FontSize',16)
    axis([0 1000 2 10])
    %title('Avg ESDs for Different Loads on a Hose and Pipe Pressure
       Readings','FontSize',16);
    set(gcf, 'Name', 'Avg_ESDs');
    xlabel('Frequency (Hz)','FontSize',16);
    ylabel('Energy','FontSize',16);
    %saveas(gcf,horzcat('Avg ESDs for Different Loads on a Hose and Pipe
        Pressure Readings','.bmp'))
    saveas(gcf,horzcat('ch4_different_loads_pipe_press_ESD','.eps'))
%--------------------------------------------
% Plot all averaged accelerometer ESD's together
```

```
figure
    plot(...
        proper_freq,(ESDforAccelBigHole./2),'b-',...
        proper_freq,(ESDforAccelMedHole./2),'b--',...
        proper_freq,(ESDforAccelSmallHole./2),'b-.',...
        proper_freq,(ESDforAccelNoHole./2),'b:')
set(gca,'FontSize',16)
legend('Accel Big Hole','Accel Medium Hole','Accel Small Hole','
    Accel No Hole','Location','NorthEast','FontSize',16)
axis([0 1000 150 700])
%title('Avg ESDs for Different Loads on a Hose and Pipe
    Accelerometer Readings', 'FontSize',16);
set(gcf, 'Name', 'Avg ESDs');
xlabel('Frequency (Hz)','FontSize',16);
ylabel('Energy','FontSize',16);
%saveas(gcf,horzcat('Avg ESDs for Different Loads on a Hose and Pipe
     Accelerometer Readings', '.bmp'))
saveas(gcf,horzcat('ch4_different_loads_pipe_accel_ESD','.eps'))
%----------------------------------------
```

A.4 Individual Load Transients Experiments Plots

A.4.1 Individual Load Transients Plots

```
% WaterNILM Signal Processing
% Sabrina Neuman (sneuman@mit.edu)
% Zack Remscrim (remscrim@mit.edu)

%———————————————————————————————
% Individual Load Transients
%———————————————————————————————
% This script is for plotting the ESD and Spectral Envelope for
%       Garden Hose on Experimental Setup, Accelerometer Data
%       Shower Head on Experimental Setup, Accelerometer Data
%       Vegetable Sprayer on Experimental Setup, Accelerometer Data
% to show the different ESD peak magnitudes for each load,
% and then what the loads' Spectral Envelopes look like
% at the different ESD peak frequencies.

%———————————————————————————————
close all
clear
%—————————————————————
for dat = 1:3;                  % which wav files to process
%—————————————————————
fs                              = 48000;        % sampling freq of
    48,000 Hz
ptsPerWindow    = 4000;         % points per FFT window
%———————————————————————————————
%—————————————————————
% Read in the data from wav files:
    switch dat
        case 1
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\
    Individual_Load_Transients_Data\Data_1_13\
    accel_full_mid_hose_10_30_10_1' ...
,[1 ((50*fs)+1)]);
titlestring = 'garden_hose'; % general title
        case 2
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\
    Individual_Load_Transients_Data\Data_1_13\
    accel_full_mid_shower_10_30_10_2' ...
,[1 ((50*fs)+1)]);
titlestring = 'shower_head'; % general title
        case 3
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\
```

```
    Individual␣Load␣Transients␣Data\Data_1_13\
    accel_full_mid_vegspray_10_30_10_1' ...
,[1 ((50*fs)+1)]);
titlestring = 'veg_spray'; % general title
     end
%---------------------------------------------
% Calculate the Energy Spectral Density
[time_sum, proper_time, proper_freq, T, P, wav_length] =
    Fxn_Calculate_ESD(fs,ptsPerWindow,dPtr);
%---------------------------------------------
% Choose ESD plot axis
            ymin = 0;
            ymax = 500;
            label72 = '72Hz␣\newline␣\newline␣\newline';
            circl72 = 0;
            label384 = '384Hz␣\newline␣\newline␣\newline';
            circl384 = 0;
            label924 = '924Hz␣\newline␣\newline␣\newline';
            circl924 = 0;
        switch dat
        case 1
            ymin = 0;
            ymax = 500;
            label72 = '␣␣␣␣␣72Hz␣\newline␣\newline␣\newline';
            circl72 = -25;
            label384 = '384Hz␣\newline␣\newline␣\newline';
            circl384 = -25;
            label924 = '924Hz␣\newline␣\newline␣\newline';
            circl924 = -25;
        case 2
            ymin = 0;
            ymax = 2500;
            label72 = '␣␣␣␣␣72Hz␣\newline␣\newline␣\newline';
            circl72 = -100;
            label384 = '384Hz␣\newline␣\newline␣\newline';
            circl384 = -100;
            label924 = '924Hz␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣\newline␣\newline␣\
                newline';
            circl924 = -100;
        case 3
            ymin = 0;
            ymax = 250;
            label72 = '␣␣␣␣␣72Hz␣\newline␣\newline␣\newline';
            circl72 = -10;
            label384 = '384Hz␣\newline␣\newline␣\newline';
            circl384 = -10;
            label924 = '924Hz␣\newline␣\newline␣\newline';
            circl924 = -10;
        end
%---------------------------------------------
```

```
% Plot the Energy Spectral Density
    figure
        plot(proper_freq,time_sum)
    % mark 72 Hz
    text(72.02,time_sum(7),label72,'FontSize',16,'HorizontalAlignment','
        center')
    text(72.02,circl72+time_sum(7),'\circ','FontSize',72,'
        HorizontalAlignment','center')
    % mark 384 Hz
    text(384.1,time_sum(33),label384,'FontSize',16,'HorizontalAlignment'
        ,'center')
    text(384.1,circl384+time_sum(33),'\circ','FontSize',72,'
        HorizontalAlignment','center')
    % mark 924 Hz
    text(924.2,time_sum(78),label924,'FontSize',16,'HorizontalAlignment'
        ,'center')
    text(924.2,circl924+time_sum(78),'\circ','FontSize',72,'
        HorizontalAlignment','center')
    set(gca,'FontSize',16)
        xlim([0 1000])
    ylim([ymin ymax])
    titlestring_ESD = horzcat('ESD for ',titlestring);
    %title(titlestring_ESD,'FontSize',16);
    set(gcf, 'Name', titlestring_ESD);
    xlabel('Frequency (Hz)','FontSize',16);
    ylabel('Energy','FontSize',16);
    %saveas(gcf,['ESD for ',titlestring,'.bmp'])
    saveas(gcf,['ch4_',titlestring,'_ESD','.eps'])
%-----------------------------------------
% Pick out frequency bands
    for bandindex = 1:3
        switch bandindex
            case 1
                bandf = 72;
            case 2
                bandf = 384;
            case 3
                bandf = 924;
        end
%-----------------------------------------
% Calculate and twice-filter the frequency band transient or "Spectral
    Envelope"
[refiltered_1] = Fxn_Calculate_and_Filter_Spectral_Envelope(bandf,T,P);
%-----------------------------------------
% Choose consistant "Spectral Envelope" plot axis
            ymin = 0;
            ymax = 50;
        switch dat
        case 1
            ymin = 0;
```

```
            ymax = 6;
        case 2
            ymin = 0;
            ymax = 32;
        case 3
            ymin = 0;
            ymax = 7;
        end
%------------------------------------------
% Plot the twice-filtered frequency band transient or "Spectral Envelope
    "
%        figure
%        plot(proper_time, refiltered_1)
%    set(gca, 'FontSize', 16)
%        xlim([0 wav_length])
%      ylim([ymin ymax])
%        %title(horzcat(titlestring, ' at ', int2str(bandf) , 'Hz.'), '
    FontSize', 16);
%        set(gcf, 'Name', horzcat(titlestring, ' at ', int2str(bandf), 'Hz
    .'));
%        xlabel('Time (s)', 'FontSize', 16);
%        ylabel('Spectral Envelope Value', 'FontSize', 16);
%        %saveas(gcf, [titlestring, ' at ', int2str(bandf), 'Hz', '.bmp'])
%    saveas(gcf, ['ch4_', titlestring, '_SpecEnv_', int2str(bandf), 'Hz', '.eps
    '])
%------------------------------------------
% Choose closer view "Spectral Envelope" plot axis
            ymin = 0;
            ymax = 50;
        switch dat
        case 1
                                switch bandindex
                                case 1
                                        ymin = 0;
                                        ymax = 3;
                                case 2
                                        ymin = 0;
                                        ymax = 6;
                                case 3
                                        ymin = 0;
                                        ymax = 4.5;
                                end
        case 2
                                switch bandindex
                                case 1
                                        ymin = 0;
                                        ymax = 3;
                                case 2
                                        ymin = 0;
                                        ymax = 11.5;
```

```
                        case 3
                                ymin = 0;
                                ymax = 32;
                        end
        case 3
                        switch bandindex
                        case 1
                                ymin = 0;
                                ymax = 8;
                        case 2
                                ymin = 0;
                                ymax = 1.5;
                        case 3
                                ymin = 0;
                                ymax = 2.5;
                        end
        end
%------------------------------------------------------------
% Plot the twice-filtered frequency band transient or "Spectral Envelope
    "
        figure
        plot(proper_time, refiltered_1)
    set(gca, 'FontSize',16)
        xlim([0 wav_length])
    ylim([ymin ymax])
        %title(horzcat(titlestring, ' at ', int2str(bandf) , 'Hz.'), '
            FontSize',16);
        set(gcf, 'Name', horzcat(titlestring, ' at ', int2str(bandf), 'Hz.'
            ));
        xlabel('Time (s)', 'FontSize',16);
        ylabel('Spectral Envelope Value', 'FontSize',16);
        % saveas(gcf,[titlestring, ' at ', int2str(bandf), 'Hz', '.bmp'])
    saveas(gcf,['ch4_',titlestring,'_SpecEnv_',int2str(bandf),'Hz','.eps
        '])
%------------------------------------------------------------
        end % End of freq band for loop
%------------------------------------------------------------------
end % End of for loop
```

A.5 Multiple Load Transients Experiments Plots

A.5.1 Multiple Load Transients Plots

```
% WaterNILM Signal Processing
% Sabrina Neuman (sneuman@mit.edu)
% Zack Remscrim (remscrim@mit.edu)

%---------------------------------------------------------------
% Multiple Load Transients
%---------------------------------------------------------------
% This script is for plotting the ESD and Spectral Envelope for
%       Garden Hose on Experimental Setup, Accelerometer Data
%       Shower Head on Experimental Setup, Accelerometer Data
%       Vegetable Sprayer on Experimental Setup, Accelerometer Data
% to show the different ESD peak magnitudes for each load,
% and then what the loads' Spectral Envelopes look like
% at the different ESD peak frequencies.

%---------------------------------------------------------------
close all
clear
%-----------------------------------------------
for dat = 4:5;                  % which wav files to process
%-----------------------------------------------
fs                              = 48000;        % sampling freq of
   48,000 Hz
ptsPerWindow    = 4000;         % points per FFT window
%---------------------------------------------------------------
%-----------------------------------------------
% Read in the data from wav files:
    switch dat
        case 1
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\
   Individual_Load_Transients_Data\Data_1_13\
   accel_full_mid_hose_10_30_10_1' ...
,[1 ((50*fs)+1)]);
titlestring = 'garden_hose'; % general title
        case 2
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\
   Individual_Load_Transients_Data\Data_1_13\
   accel_full_mid_shower_10_30_10_2' ...
,[1 ((50*fs)+1)]);
titlestring = 'shower_head'; % general title
        case 3
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\
```

```
    Individual_Load_Transients_Data\Data_1_13\
    accel_full_mid_vegspray_10_30_10_1' ...
,[1 ((50*fs)+1)]);
titlestring = 'veg_spray'; % general title
        case 4
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\
    Multiple_Load_Transients_Data\Data_1_13\
    accel_full_mid_hose_and_shower_10_30_10_1' ...
,[1 ((50*fs)+1)]);
titlestring = 'garden_hose_shower_head'; % general title
        case 5
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\
    Multiple_Load_Transients_Data\Data_1_13\
    accel_full_mid_hose_and_vegspray_10_30_10_1' ...
,[1 ((50*fs)+1)]);
titlestring = 'garden_hose_veg_spray'; % general title
    end
%-----------------------------------------
% Calculate the Energy Spectral Density
[time_sum, proper_time, proper_freq, T, P, wav_length] =
    Fxn_Calculate_ESD(fs,ptsPerWindow,dPtr);
proper_time(end)
%-----------------------------------------
% Choose ESD plot axis
            ymin = 0;
            ymax = 500;
            label72 = '72Hz_\newline_\newline_\newline';
            circl72 = 0;
            label384 = '384Hz_\newline_\newline_\newline';
            circl384 = 0;
            label924 = '924Hz_\newline_\newline_\newline';
            circl924 = 0;
        switch dat
        case 1
            ymin = 0;
            ymax = 500;
            label72 = '_____72Hz_\newline_\newline_\newline';
            circl72 = -25;
            label384 = '384Hz_\newline_\newline_\newline';
            circl384 = -25;
            label924 = '924Hz_\newline_\newline_\newline';
            circl924 = -25;
        case 2
            ymin = 0;
            ymax = 2500;
            label72 = '_____72Hz_\newline_\newline_\newline';
            circl72 = -100;
            label384 = '384Hz_\newline_\newline_\newline';
```

```
            circl384 = −100;
            label924 = '924Hz␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣\newline␣\newline␣\
                newline';
            circl924 = −100;
        case 3
            ymin = 0;
            ymax = 500;
            label72 = '␣␣␣␣␣72Hz␣\newline␣\newline␣\newline';
            circl72 = −10;
            label384 = '384Hz␣\newline␣\newline␣\newline';
            circl384 = −10;
            label924 = '924Hz␣\newline␣\newline␣\newline';
            circl924 = −10;
        case 4
            ymin = 0;
            ymax = 600;
            label72 = '␣␣␣␣␣72Hz␣\newline␣\newline␣\newline␣\newline';
            circl72 = −15;
            label384 = '384Hz␣\newline␣\newline␣\newline␣\newline';
            circl384 = −15;
            label924 = '924Hz␣\newline␣\newline␣\newline␣\newline';
            circl924 = −15;
        case 5
            ymin = 0;
            ymax = 450;
            label72 = '␣␣␣␣␣72Hz␣\newline␣\newline␣\newline␣\newline';
            circl72 = −15;
            label384 = '384Hz␣\newline␣\newline␣\newline␣\newline';
            circl384 = −15;
            label924 = '924Hz␣\newline␣\newline␣\newline␣\newline';
            circl924 = −15;
        end
%——————————————————————————————
% Plot the Energy Spectral Density
    figure
        plot(proper_freq,time_sum)
    % mark 72 Hz
    text(72.02,time_sum(7),label72,'FontSize',16,'HorizontalAlignment','
        center')
    text(72.02,circl72+time_sum(7),'\circ','FontSize',72,'
        HorizontalAlignment','center')
    % mark 384 Hz
    text(384.1,time_sum(33),label384,'FontSize',16,'HorizontalAlignment'
        ,'center')
    text(384.1,circl384+time_sum(33),'\circ','FontSize',72,'
        HorizontalAlignment','center')
    % mark 924 Hz
    text(924.2,time_sum(78),label924,'FontSize',16,'HorizontalAlignment'
        ,'center')
    text(924.2,circl924+time_sum(78),'\circ','FontSize',72,'
```

```
        HorizontalAlignment','center')
    set(gca,'FontSize',16)
        xlim([0 1000])
    ylim([ymin ymax])
    titlestring_ESD = horzcat('ESD_for_',titlestring);
    %title(titlestring_ESD,'FontSize',16);
    set(gcf, 'Name', titlestring_ESD);
    xlabel('Frequency_(Hz)','FontSize',16);
    ylabel('Energy','FontSize',16);
    %saveas(gcf,['ESD for ',titlestring,'.bmp'])
    saveas(gcf,['ch4_',titlestring,'_ESD','.eps'])
%------------------------------------------------
% Pick out frequency bands
    for bandindex = 1:3
        switch bandindex
            case 1
                bandf = 72;
            case 2
                bandf = 384;
            case 3
                bandf = 924;
        end
%------------------------------------------------
% Calculate and twice-filter the frequency band transient or "Spectral
    Envelope"
[refiltered_1] = Fxn_Calculate_and_Filter_Spectral_Envelope(bandf,T,P);
%------------------------------------------------
% Choose consistant "Spectral Envelope" plot axis
            ymin = 0;
            ymax = 50;
        switch dat
        case 1
            ymin = 0;
            ymax = 6;
        case 2
            ymin = 0;
            ymax = 32;
        case 3
            ymin = 0;
            ymax = 7;
        case 4
            ymin = 0;
            ymax = 32;
        case 5
            ymin = 0;
            ymax = 7;
        end
%------------------------------------------------
% Plot the twice-filtered frequency band transient or "Spectral Envelope
    "
```

```
%       figure
%       plot(proper_time, refiltered_1)
%   set(gca, 'FontSize', 16)
%       xlim([0 wav_length])
%     ylim([ymin ymax])
%       %title(horzcat(titlestring, ' at ', int2str(bandf) , 'Hz.'), '
    FontSize', 16);
%       set(gcf, 'Name', horzcat(titlestring, ' at ', int2str(bandf), 'Hz
    .'));
%       xlabel('Time (s)', 'FontSize', 16);
%       ylabel('Spectral Envelope Value', 'FontSize', 16);
%       %saveas(gcf, [titlestring, ' at ', int2str(bandf), 'Hz', '.bmp'])
%   saveas(gcf, ['ch4_', titlestring, '_SpecEnv_', int2str(bandf), 'Hz', '.eps
    '])
%------------------------------------
% Choose closer view "Spectral Envelope" plot axis
        ymin = 0;
        ymax = 50;
    switch dat
    case 1
                    switch bandindex
                    case 1
                        ymin = 0;
                        ymax = 4;
                    case 2
                        ymin = 0;
                        ymax = 6;
                    case 3
                        ymin = 0;
                        ymax = 5;
                    end
    case 2
                    switch bandindex
                    case 1
                        ymin = 0;
                        ymax = 4;
                    case 2
                        ymin = 0;
                        ymax = 12;
                    case 3
                        ymin = 0;
                        ymax = 32;
                    end
    case 3
                    switch bandindex
                    case 1
                        ymin = 0;
                        ymax = 6;
                    case 2
                        ymin = 0;
```

```
                    ymax = 4;
                case 3
                    ymin = 0;
                    ymax = 4;
                end
        case 4
                switch bandindex
                case 1
                    ymin = 0;
                    ymax = 4;
                case 2
                    ymin = 0;
                    ymax = 13;
                case 3
                    ymin = 0;
                    ymax = 15;
                end
        case 5
                switch bandindex
                case 1
                    ymin = 0;
                    ymax = 8;
                case 2
                    ymin = 0;
                    ymax = 6;
                case 3
                    ymin = 0;
                    ymax = 3;
                end
        end
%------------------------------------------------
% Plot the twice-filtered frequency band transient or "Spectral Envelope
   "
        figure
        plot(proper_time, refiltered_1)
    line([20 ; 20],[ymin ; ymax],'LineStyle','-.')
    line([30 ; 30],[ymin ; ymax],'LineStyle','-.')
    set(gca,'FontSize',16)
        xlim([0 wav_length])
    ylim([ymin ymax])
        %title(horzcat(titlestring,' at ',int2str(bandf) ,'Hz.'),'
            FontSize',16);
        set(gcf, 'Name', horzcat(titlestring,' at ',int2str(bandf),'Hz.'
            ));
        xlabel('Time (s)','FontSize',16);
        ylabel('Spectral Envelope Value','FontSize',16);
        % saveas(gcf,[titlestring,' at ',int2str(bandf),'Hz','.bmp'])
    saveas(gcf,['ch4_',titlestring,'_SpecEnv_',int2str(bandf),'Hz','.eps
        '])
%------------------------------------------------
```

```
        end % End of freq band for loop
%-----------------------------------------------------------
end % End of for loop
```

A.6 Flow Rate Experiments Plots

A.6.1 Flow Rate Plots

```
% WaterNILM Signal Processing
% Sabrina Neuman (sneuman@mit.edu)
% Zack Remscrim (remscrim@mit.edu)

%---------------------------------------------------------
% Flow Rate
%---------------------------------------------------------
% This script is for plotting the ESD for
%       Garden Hose on Experimental Setup, Accelerometer Data, Flow Rate
    0.0 GPM x2
%       Garden Hose on Experimental Setup, Accelerometer Data, Flow Rate
    1.1 GPM x2
%       Garden Hose on Experimental Setup, Accelerometer Data, Flow Rate
    2.0 GPM x2
%       Garden Hose on Experimental Setup, Accelerometer Data, Flow Rate
    3.0 GPM x2
%       Garden Hose on Experimental Setup, Accelerometer Data, Flow Rate
    4.0 GPM x2
%       Garden Hose on Experimental Setup, Accelerometer Data, Flow Rate
    4.8 GPM x2
% to show frequency content magnitude at different flow rates

%---------------------------------------------------------
close all
clear
%-----------------------------------
% Initialize vectors for averaging
ESDfor0P0 = zeros(1,2000)';
ESDfor1P1 = zeros(1,2000)';
ESDfor2P0 = zeros(1,2000)';
ESDfor3P0 = zeros(1,2000)';
ESDfor4P0 = zeros(1,2000)';
ESDfor4P8 = zeros(1,2000)';
%-----------------------------------
for dat = 1:12;                 % which wav files to process
%-----------------------------------
fs                              = 48000;        % sampling freq of
   48,000 Hz
ptsPerWindow      = 4000;       % points per FFT window
%---------------------------------------------------------
%-----------------------------------
% Read in the data from wav files:
    switch dat
            case 1
dPtr = wavread( ...
```

```
'C:\Documents␣and␣Settings\sneuman\Desktop\MEng\MEng␣Thesis\MATLAB\Flow␣
   Rate␣Data\Data_1_29\accel_full_mid_hose_flow0P0_10s_1' ...
,[(1) ((10*fs)+1)]);
titlestring = 'Flow␣0.0␣GPM'; % general title
        case 2
dPtr = wavread( ...
'C:\Documents␣and␣Settings\sneuman\Desktop\MEng\MEng␣Thesis\MATLAB\Flow␣
   Rate␣Data\Data_1_29\accel_full_mid_hose_flow0P0_10s_2' ...
,[(1) ((10*fs)+1)]);
titlestring = 'Flow␣0.0␣GPM'; % general title
        case 3
dPtr = wavread( ...
'C:\Documents␣and␣Settings\sneuman\Desktop\MEng\MEng␣Thesis\MATLAB\Flow␣
   Rate␣Data\Data_1_29\accel_full_mid_hose_flow1P1_10s_1' ...
,[(1) ((10*fs)+1)]);
titlestring = 'Flow␣1.1␣GPM'; % general title
        case 4
dPtr = wavread( ...
'C:\Documents␣and␣Settings\sneuman\Desktop\MEng\MEng␣Thesis\MATLAB\Flow␣
   Rate␣Data\Data_1_29\accel_full_mid_hose_flow1P1_10s_2' ...
,[(1) ((10*fs)+1)]);
titlestring = 'Flow␣1.1␣GPM'; % general title
        case 5
dPtr = wavread( ...
'C:\Documents␣and␣Settings\sneuman\Desktop\MEng\MEng␣Thesis\MATLAB\Flow␣
   Rate␣Data\Data_1_29\accel_full_mid_hose_flow2P0_10s_1' ...
,[(1) ((10*fs)+1)]);
titlestring = 'Flow␣2.0␣GPM'; % general title
        case 6
dPtr = wavread( ...
'C:\Documents␣and␣Settings\sneuman\Desktop\MEng\MEng␣Thesis\MATLAB\Flow␣
   Rate␣Data\Data_1_29\accel_full_mid_hose_flow2P0_10s_2' ...
,[(1) ((10*fs)+1)]);
titlestring = 'Flow␣2.0␣GPM'; % general title
        case 7
dPtr = wavread( ...
'C:\Documents␣and␣Settings\sneuman\Desktop\MEng\MEng␣Thesis\MATLAB\Flow␣
   Rate␣Data\Data_1_29\accel_full_mid_hose_flow3P0_10s_1' ...
,[(1) ((10*fs)+1)]);
titlestring = 'Flow␣3.0␣GPM'; % general title
        case 8
dPtr = wavread( ...
'C:\Documents␣and␣Settings\sneuman\Desktop\MEng\MEng␣Thesis\MATLAB\Flow␣
   Rate␣Data\Data_1_29\accel_full_mid_hose_flow3P0_10s_2' ...
,[(1) ((10*fs)+1)]);
titlestring = 'Flow␣3.0␣GPM'; % general title
        case 9
dPtr = wavread( ...
'C:\Documents␣and␣Settings\sneuman\Desktop\MEng\MEng␣Thesis\MATLAB\Flow␣
   Rate␣Data\Data_1_29\accel_full_mid_hose_flow4P0_10s_1' ...
```

```
,[(1) ((10*fs)+1)]);
titlestring = 'Flow_4.0_GPM'; % general title
        case 10
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\Flow_
   Rate_Data\Data_1_29\accel_full_mid_hose_flow4P0_10s_2' ...
,[(1) ((10*fs)+1)]);
titlestring = 'Flow_4.0_GPM'; % general title
        case 11
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\Flow_
   Rate_Data\Data_1_29\accel_full_mid_hose_flow4P8_10s_1' ...
,[(1) ((10*fs)+1)]);
titlestring = 'Flow_4.8_GPM'; % general title
        case 12
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\Flow_
   Rate_Data\Data_1_29\accel_full_mid_hose_flow4P8_10s_2' ...
,[(1) ((10*fs)+1)]);
titlestring = 'Flow_4.8_GPM'; % general title
        end
%------------------------------------------
% Calculate the Energy Spectral Density
[time_sum, proper_time, proper_freq, T, P, wav_length] =
   Fxn_Calculate_ESD(fs,ptsPerWindow,dPtr);
%------------------------------------------
% Take sums of ESD's, for averaging
    if          ((dat>0)&&(dat<3))
        ESDfor0P0 = ESDfor0P0+time_sum;
    elseif      ((dat>2)&&(dat<5))
        ESDfor1P1 = ESDfor1P1+time_sum;
    elseif      ((dat>4)&&(dat<7))
        ESDfor2P0 = ESDfor2P0+time_sum;
    elseif      ((dat>6)&&(dat<9))
        ESDfor3P0 = ESDfor3P0+time_sum;
    elseif      ((dat>8)&&(dat<11))
        ESDfor4P0 = ESDfor4P0+time_sum;
    elseif      ((dat>10)&&(dat<13))
        ESDfor4P8 = ESDfor4P8+time_sum;
    end
%------------------------------------------
%-----------------------------------------------------
end % End of for loop
%-----------------------------------------------------
%------------------------------------------
% Plot all averaged flow rate ESD's together
    figure
        %plot( ...
    semilogy( ...
            proper_freq,(ESDfor0P0/2),'b-',...
```

```
            proper_freq ,(ESDfor1P1/2) ,'b—' ,...
            proper_freq ,(ESDfor2P0/2) ,'b-.' ,...
            proper_freq ,(ESDfor3P0/2) ,'b:' ,...
            proper_freq ,(ESDfor4P0/2) ,'b-o' ,...
            proper_freq ,(ESDfor4P8/2) ,'b-s')
    set(gca,'FontSize',16)
    legend('0.0_GPM','1.1_GPM','2.0_GPM','3.0_GPM','4.0_GPM','4.8_GPM','
       Location','NorthEast','FontSize',14)
    axis([0 1000 50 1000])
    %title('Avg ESDs for Different Flow Rates 0 to 1000 Hz');
    set(gcf, 'Name', 'Avg_ESDs');
    xlabel('Frequency_(Hz)','FontSize',16);
    %ylabel('Energy','FontSize',16);
    ylabel('Log(Energy)','FontSize',16);
    %saveas(gcf,horzcat('Avg ESDs for Different Flow Rates 0 to 1000 Hz
       ','.bmp'))
    saveas(gcf,horzcat('ch4_flow_rate_ESD','.eps'))
%———————————————————————————
```

A.7 Field Experiments Plots

A.7.1 Field Site Individual Load Transients Plots

```
% WaterNILM Signal Processing
% Sabrina Neuman (sneuman@mit.edu)
% Zack Remscrim (remscrim@mit.edu)

%
% Field Site Individual Load Transients
%
% This script is for plotting the ESD and Spectral Envelope for
%       Shower at Field Site, Microphone Data
%       Bathroom Sink at Field Site, Microphone Data
%       Kitchen Sink at Field Site, Microphone Data
% to show the different ESD peak magnitudes for each load,
% and then what the loads' Spectral Envelopes look like
% at the different ESD peak frequencies.

%
close all
clear
%
for dat = 1:3;                    % which wav files to process
%
fs                                = 48000;        % sampling freq of
    48,000 Hz
ptsPerWindow      = 4000;         % points per FFT window
%
%
% Read in the data from wav files:
    switch dat
        case 1
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\Field
    _Test_Data\COLD_sean_house_data\C_bath_shower_W_1' ...
,[((20*fs)+1) ((70*fs)+1)]);
titlestring = 'shower'; % general title
        case 2
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\Field
    _Test_Data\COLD_sean_house_data\C_bath_sink_C_1' ...
,[((20*fs)+1) ((70*fs)+1)]);
titlestring = 'bathroom_sink'; % general title
        case 3
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\Field
    _Test_Data\COLD_sean_house_data\C_kitchen_sink_C_2' ...
,[((20*fs)+1) ((70*fs)+1)]);
```

```
titlestring = 'kitchen_sink'; % general title
    end
%------------------------------------------
% Calculate the Energy Spectral Density
[time_sum, proper_time, proper_freq, T, P, wav_length] =
    Fxn_Calculate_ESD(fs,ptsPerWindow,dPtr);
%------------------------------------------
% Choose ESD plot axis
            ymin = 0;
            ymax = 15000;
            label324 = '324Hz \newline \newline \newline';
            circl324 = -400;
            label576 = '576Hz \newline \newline \newline';
            circl576 = -400;
            label1380 = '1380Hz \newline \newline \newline';
            circl1380 = -400;
        switch dat
        case 1
            ymin = 0;
            ymax = 15000;
            label324 = '324Hz \newline \newline \newline \newline';
            circl324 = -400;
            label576 = '576Hz \newline \newline';
            circl576 = -1000;
            label1380 = '1380Hz \newline \newline \newline \newline';
            circl1380 = -400;
        case 2
            ymin = 0;
            ymax = 15000;
            label324 = '324Hz \newline \newline \newline \newline';
            circl324 = -400;
            label576 = '576Hz \newline \newline \newline \newline';
            circl576 = -400;
            label1380 = '1380Hz \newline \newline \newline \newline';
            circl1380 = -400;
        case 3
            ymin = 0;
            ymax = 15000;
            label324 = '324Hz \newline \newline \newline \newline';
            circl324 = -400;
            label576 = '576Hz \newline \newline \newline \newline';
            circl576 = -400;
            label1380 = '1380Hz \newline \newline \newline \newline \
                newline';
            circl1380 = 400;
        end
%------------------------------------------
% Plot the Energy Spectral Density
    figure
        plot(proper_freq,time_sum)
```

```
    % mark 324 Hz
    text(324.1,time_sum(28),label324,'FontSize',16,'HorizontalAlignment'
        ,'center')
    text(324.1,circl324+time_sum(28),'\circ','FontSize',72,'
        HorizontalAlignment','center')
    % mark 576 Hz
    text(576.1,time_sum(49),label576,'FontSize',16,'HorizontalAlignment'
        ,'center')
    text(576.1,circl576+time_sum(49),'\circ','FontSize',72,'
        HorizontalAlignment','center')
    % mark 1380 Hz
    text(1380,time_sum(116),label1380,'FontSize',16,'HorizontalAlignment
        ','center')
    text(1380,circl1380+time_sum(116),'\circ','FontSize',72,'
        HorizontalAlignment','center')
    set(gca,'FontSize',16)
        xlim([0 1500])
    ylim([ymin ymax])
    titlestring_ESD = horzcat('ESD for ',titlestring);
    %title(titlestring_ESD);
    set(gcf, 'Name', titlestring_ESD);
    xlabel('Frequency (Hz)','FontSize',16);
    ylabel('Energy','FontSize',16);
    %saveas(gcf,['ESD for ',titlestring,'.bmp'])
    saveas(gcf,['ch4_',titlestring,'_ESD','.eps'])
%------------------------------------------------------------
% Pick out frequency bands
    for bandindex = 1:3
        switch bandindex
            case 1
                bandf = 324;
            case 2
                bandf = 576;
            case 3
                bandf = 1380;
        end
%------------------------------------------------------------
% Calculate and twice-filter the frequency band transient or "Spectral
  Envelope"
[refiltered_1] = Fxn_Calculate_and_Filter_Spectral_Envelope(bandf,T,P);
%------------------------------------------------------------
% Choose consistant "Spectral Envelope" plot axis
            ymin = 0;
            ymax = 1000;
        switch dat
        case 1
            ymin = 0;
            ymax = 1000;
        case 2
            ymin = 0;
```

```
            ymax = 1000;
        case 3
            ymin = 0;
            ymax = 1000;
        end
%------------------------------------------
% Plot the twice-filtered frequency band transient or "Spectral Envelope
   "
%       figure
%       plot(proper_time, refiltered_1)
%     set(gca, 'FontSize',16)
%       xlim([0 wav_length])
%     ylim([ymin ymax])
%       %title(horzcat(titlestring,' at ',int2str(bandf) ,'Hz.'));
%       set(gcf, 'Name', horzcat(titlestring,' at ',int2str(bandf),'Hz
   .'));
%       xlabel('Time (s)','FontSize',16);
%       ylabel('Spectral Envelope Value','FontSize',16);
%       %saveas(gcf,[titlestring,' at ',int2str(bandf),'Hz','.bmp'])
%     saveas(gcf,['ch4_',titlestring,'_SpecEnv_',int2str(bandf),'Hz','.
   eps'])
%------------------------------------------
% Choose closer view "Spectral Envelope" plot axis
            ymin = 0;
            ymax = 1000;
        switch dat
        case 1
                        switch bandindex
                        case 1
                                ymin = 0;
                                ymax = 400;
                        case 2
                                ymin = 0;
                                ymax = 700;
                        case 3
                                ymin = 0;
                                ymax = 80;
                        end
        case 2
                        switch bandindex
                        case 1
                                ymin = 0;
                                ymax = 65;
                        case 2
                                ymin = 0;
                                ymax = 89;
                        case 3
                                ymin = 0;
                                ymax = 25;
                        end
```

```
        case 3
                                switch bandindex
                                case 1
                                        ymin = 0;
                                        ymax = 127;
                                case 2
                                        ymin = 0;
                                        ymax = 60;
                                case 3
                                        ymin = 0;
                                        ymax = 70;
                                end
        end
%-----------------------------------------------------------
% Plot the twice-filtered frequency band transient or "Spectral Envelope
   "
        figure
        plot(proper_time,refiltered_1)
    set(gca,'FontSize',16)
        xlim([0 wav_length])
      ylim([ymin ymax])
        %title(horzcat(titlestring,' at ',int2str(bandf) ,'Hz.'),'
            FontSize',16);
        set(gcf, 'Name', horzcat(titlestring,' at ',int2str(bandf),'Hz.'
            ));
        xlabel('Time (s)','FontSize',16);
        ylabel('Spectral Envelope Value','FontSize',16);
        % saveas(gcf,[titlestring,' at ',int2str(bandf),'Hz','.bmp'])
    saveas(gcf,['ch4_',titlestring,'_SpecEnv_',int2str(bandf),'Hz','.eps'
       ])
%-----------------------------------------------------------
        end % End of freq band for loop
%-----------------------------------------------------------
end % End of for loop
```

A.7.2 Field Site Multiple Load Transients Plots

```
% WaterNILM Signal Processing
% Sabrina Neuman (sneuman@mit.edu)
% Zack Remscrim (remscrim@mit.edu)

%-----------------------------------------------------------
% Field Site Multiple Load Transients
%-----------------------------------------------------------
% This script is for plotting the ESD and Spectral Envelope for
%       Shower at Field Site, Microphone Data
%       Bathroom Sink at Field Site, Microphone Data
%       Kitchen Sink at Field Site, Microphone Data
% to show the different ESD peak magnitudes for each load,
```

```
% and then what the loads ' Spectral Envelopes look like
% at the different ESD peak frequencies.

%------------------------------------------------------------
close all
clear
%---------------------------------------------
for dat = 4:5;                    % which wav files to process
%---------------------------------------------
fs                                = 48000;          % sampling freq of
    48,000 Hz
ptsPerWindow      = 4000;         % points per FFT window
%------------------------------------------------------------
%---------------------------------------------
% Read in the data from wav files:
    switch dat
        case 1
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\Field
    _Test_Data\COLD_sean_house_data\C_bath_shower_W_1 ' ...
,[((20*fs)+1) ((70*fs)+1)]);
titlestring = 'shower'; % general title
        case 2
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\Field
    _Test_Data\COLD_sean_house_data\C_bath_sink_C_1 ' ...
,[((20*fs)+1) ((70*fs)+1)]);
titlestring = 'bathroom_sink'; % general title
        case 3
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\Field
    _Test_Data\COLD_sean_house_data\C_kitchen_sink_C_2 ' ...
,[((20*fs)+1) ((70*fs)+1)]);
titlestring = 'kitchen_sink'; % general title
        case 4
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\Field
    _Test_Data\COLD_sean_house_data\C_shower_and_bath_sink_1 ' ...
,[((20*fs)+1) ((130*fs)+1)]);
titlestring = 'shower_bathroom_sink'; % general title
        case 5
dPtr = wavread( ...
'C:\Documents_and_Settings\sneuman\Desktop\MEng\MEng_Thesis\MATLAB\Field
    _Test_Data\COLD_sean_house_data\C_shower_and_kitchen_sink_1 ' ...
,[((20*fs)+1) ((130*fs)+1)]);
titlestring = 'shower_kitchen_sink'; % general title
    end
%---------------------------------------------
% Calculate the Energy Spectral Density
[time_sum, proper_time, proper_freq, T, P, wav_length] =
```

```
    Fxn_Calculate_ESD(fs,ptsPerWindow,dPtr);
%------------------------------------------------
% Choose ESD plot axis
            ymin = 0;
            ymax = 15000;
        switch dat
        case 1
            ymin = 0;
            ymax = 15000;
            label324 = '324Hz␣\newline␣\newline␣\newline␣\newline';
            circl324 = -400;
            label576 = '576Hz␣\newline␣\newline';
            circl576 = -1000;
            label1380 = '1380Hz␣\newline␣\newline␣\newline␣\newline';
            circl1380 = -400;
        case 2
            ymin = 0;
            ymax = 15000;
            label324 = '324Hz␣\newline␣\newline␣\newline␣\newline';
            circl324 = -400;
            label576 = '576Hz␣\newline␣\newline␣\newline␣\newline';
            circl576 = -400;
            label1380 = '1380Hz␣\newline␣\newline␣\newline␣\newline';
            circl1380 = -400;
        case 3
            ymin = 0;
            ymax = 15000;
            label324 = '324Hz␣\newline␣\newline␣\newline␣\newline';
            circl324 = -400;
            label576 = '576Hz␣\newline␣\newline␣\newline␣\newline';
            circl576 = -400;
            label1380 = '1380Hz␣\newline␣\newline␣\newline␣\newline␣\
                newline';
            circl1380 = 400;
        case 4
            ymin = 0;
            ymax = 40000;
            label324 = '324Hz␣\newline␣\newline␣\newline␣\newline';
            circl324 = -1000;
            label576 = '576Hz␣\newline␣\newline␣\newline␣\newline';
            circl576 = -1000;
            label1380 = '1380Hz␣\newline␣\newline␣\newline␣\newline␣\
                newline';
            circl1380 = -400;
        case 5
            ymin = 0;
            ymax = 40000;
            label324 = '324Hz␣\newline␣\newline␣\newline␣\newline';
            circl324 = -1000;
            label576 = '576Hz␣\newline␣\newline␣\newline␣\newline';
```

```
            circl576 = -1000;
            label1380 = '1380Hz_\newline_\newline_\newline_\newline_';
            circl1380 = -600;
        end
%------------------------------------------------
% Plot the Energy Spectral Density
    figure
        plot(proper_freq ,time_sum)
    % mark 324 Hz
    text(324.1,time_sum(28),label324 , 'FontSize',16,'HorizontalAlignment'
        ,'center')
    text(324.1,circl324+time_sum(28),'\circ','FontSize',72,'
        HorizontalAlignment','center')
    % mark 576 Hz
    text(576.1,time_sum(49),label576 , 'FontSize',16,'HorizontalAlignment'
        ,'center')
    text(576.1,circl576+time_sum(49),'\circ','FontSize',72,'
        HorizontalAlignment','center')
    % mark 1380 Hz
    text(1380,time_sum(116),label1380 , 'FontSize',16,'HorizontalAlignment
        ','center')
    text(1380,circl1380+time_sum(116),'\circ','FontSize',72,'
        HorizontalAlignment','center')
    set(gca,'FontSize',16)
        xlim([0 1500])
    ylim([ymin ymax])
    titlestring_ESD = horzcat('ESD_for_',titlestring);
    %title(titlestring_ESD , 'FontSize',16);
    set(gcf, 'Name', titlestring_ESD);
    xlabel('Frequency_(Hz)','FontSize',16);
    ylabel('Energy','FontSize',16);
    %saveas(gcf,['ESD for ',titlestring , '.bmp'])
    saveas(gcf,['ch4_',titlestring , '_ESD','.eps'])
%------------------------------------------------
% Pick out frequency bands
    for bandindex = 1:3
        switch bandindex
            case 1
                bandf = 324;
            case 2
                bandf = 576;
            case 3
                bandf = 1380;
        end
%------------------------------------------------
% Calculate and twice-filter the frequency band transient or "Spectral
    Envelope"
[refiltered_1] = Fxn_Calculate_and_Filter_Spectral_Envelope(bandf,T,P);
%------------------------------------------------
% Choose consistant "Spectral Envelope" plot axis
```

```
            ymin = 0;
            ymax = 1000;
        switch dat
        case 1
            ymin = 0;
            ymax = 1000;
        case 2
            ymin = 0;
            ymax = 1000;
        case 3
            ymin = 0;
            ymax = 1000;
        case 4
            ymin = 0;
            ymax = 1000;
        case 5
            ymin = 0;
            ymax = 1000;
        end
%————————————————————————————————
% Plot the twice-filtered frequency band transient or "Spectral Envelope
    "
%           figure
%           plot(proper_time, refiltered_1)
%       set(gca, 'FontSize',16)
%           xlim([0 wav_length])
%       ylim([ymin ymax])
%           %title(horzcat(titlestring,' at ',int2str(bandf) ,'Hz.'),'
    FontSize',16);
%           set(gcf, 'Name', horzcat(titlestring,' at ',int2str(bandf),'Hz
    .'));
%           xlabel('Time (s)','FontSize',16);
%           ylabel('Spectral Envelope Value','FontSize',16);
%           %saveas(gcf,[titlestring,' at ',int2str(bandf),'Hz','.bmp'])
%       saveas(gcf,['ch4_',titlestring,'_SpecEnv_',int2str(bandf),'Hz','.
    eps'])
%————————————————————————————————
% Choose closer view "Spectral Envelope" plot axis
            ymin = 0;
            ymax = 1000;
        switch dat
        case 1
                                switch bandindex
                                case 1
                                        ymin = 0;
                                        ymax = 1000;
                                case 2
                                        ymin = 0;
                                        ymax = 1000;
                                case 3
```

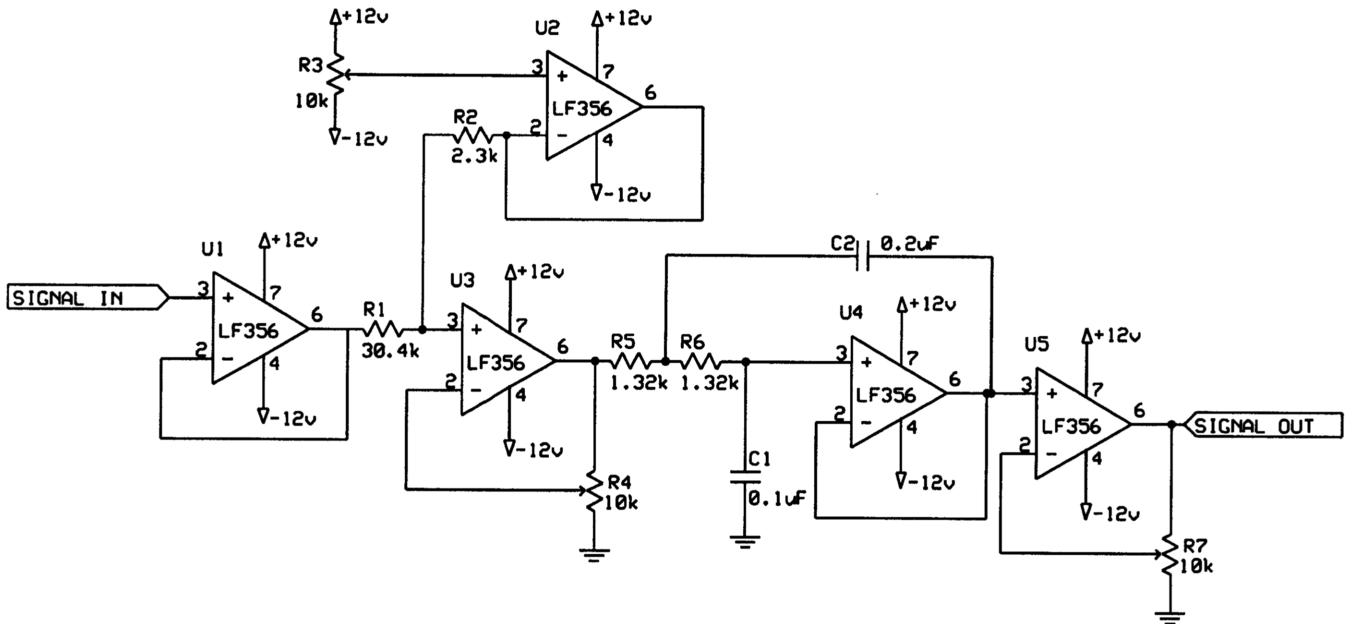
```
                ymin = 0;
                ymax = 1000;
        end
    case 2
        switch bandindex
        case 1
                ymin = 0;
                ymax = 200;
        case 2
                ymin = 0;
                ymax = 200;
        case 3
                ymin = 0;
                ymax = 200;
        end
    case 3
        switch bandindex
        case 1
                ymin = 0;
                ymax = 200;
        case 2
                ymin = 0;
                ymax = 200;
        case 3
                ymin = 0;
                ymax = 200;
        end
    case 4
        switch bandindex
        case 1
                ymin = 0;
                ymax = 300;
        case 2
                ymin = 0;
                ymax = 600;
        case 3
                ymin = 0;
                ymax = 50;
        end
    case 5
        switch bandindex
        case 1
                ymin = 0;
                ymax = 300;
        case 2
                ymin = 0;
                ymax = 600;
        case 3
                ymin = 0;
                ymax = 50;
```

```
                    end
        end
%------------------------------------------------
% Plot the twice-filtered frequency band transient or "Spectral Envelope
    "
        figure
        plot(proper_time,refiltered_1)
    line([40 ; 40],[ymin ; ymax],'LineStyle','-.')
    line([70 ; 70],[ymin ; ymax],'LineStyle','-.')
    set(gca,'FontSize',16)
        xlim([0 wav_length])
     ylim([ymin ymax])
        %title(horzcat(titlestring,' at ',int2str(bandf) ,'Hz.'),'
            FontSize',16);
        set(gcf, 'Name', horzcat(titlestring,' at ',int2str(bandf),'Hz.'
            ));
        xlabel('Time (s)','FontSize',16);
        ylabel('Spectral Envelope Value','FontSize',16);
        % saveas(gcf,[titlestring,' at ',int2str(bandf),'Hz','.bmp'])
    saveas(gcf,['ch4_',titlestring,'_SpecEnv_',int2str(bandf),'Hz','.eps
        '])
%------------------------------------------------
        end % End of freq band for loop
%-------------------------------------------------------------
end % End of for loop
```

Appendix B

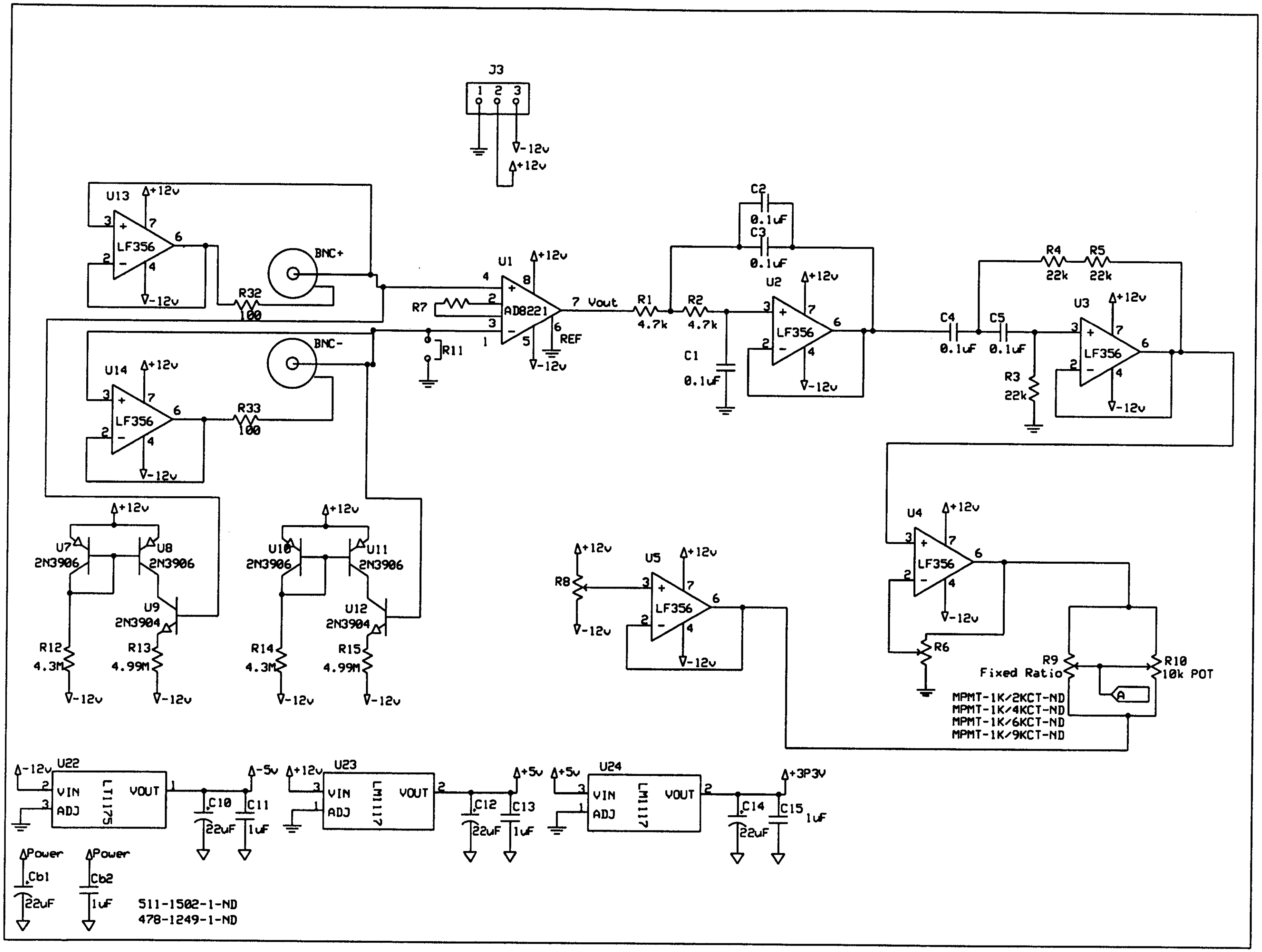
Circuit Schematics

B.1 Non-Intrusive Water Utility Monitor Signal Processing Schematic

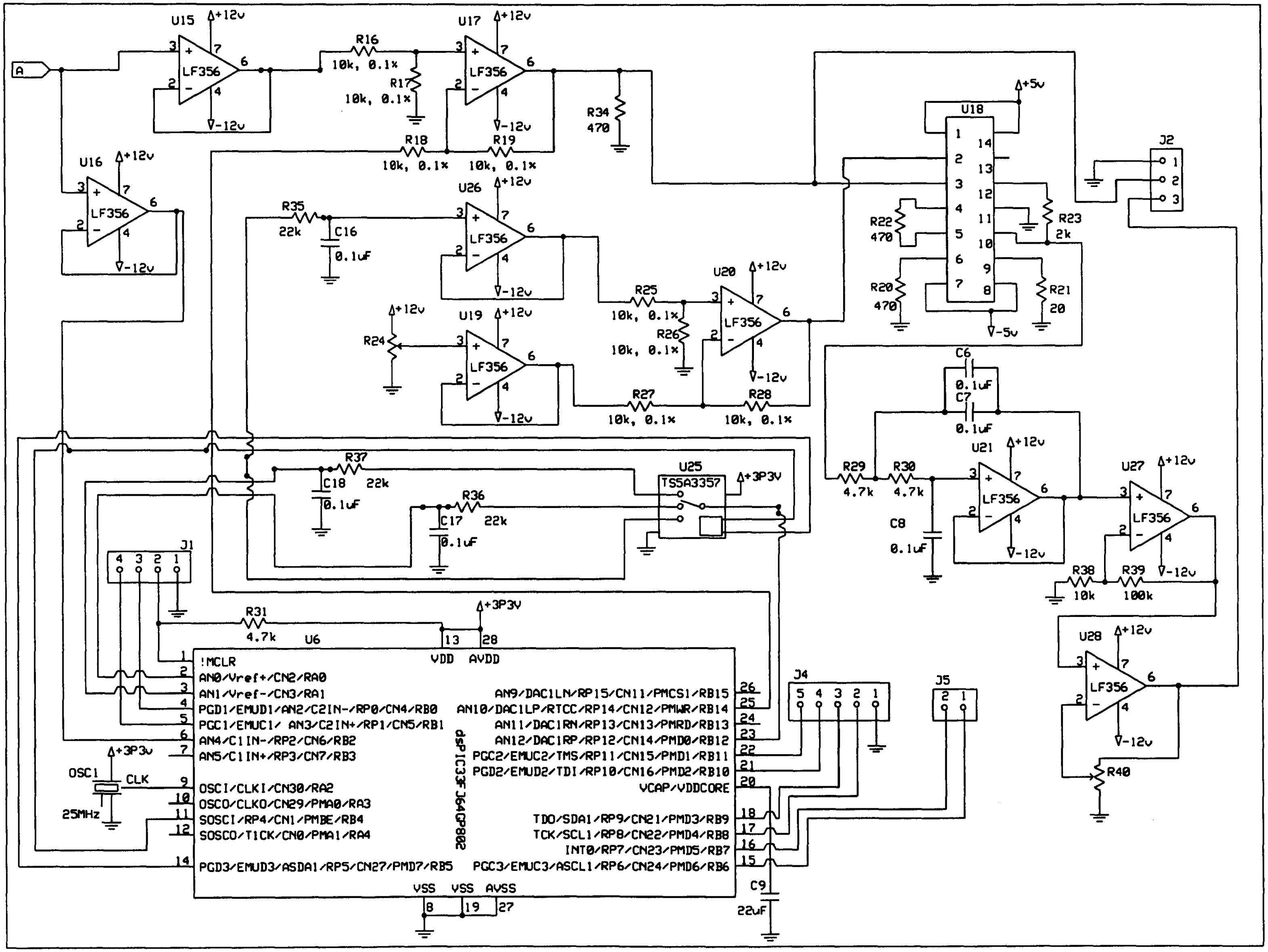


FigureB.1: Non-Intrusive Water Utility Monitor Signal Processing Schematic

B.2 Electroquasistatic (EQS) Sensor Schematic



FigureB.2: EQS Sensor Schematic Page 1



FigureB.3: EQS Sensor Schematic Page 2

Bibliography

[1] S. B. Leeb. *A conjoint pattern recognition approach to nonintrusive load monitoring.* PhD thesis, Massachusetts Institute of Technology, 1993.

[2] S. B. Leeb, S. R. Shaw, and J. L. Kirtley Jr. Transient event detection in spectral envelope estimates for nonintrusive load monitoring. *IEEE Transactions on Power Delivery,* 10(3):1200–1210, 1995.

[3] S. R. Shaw, S. B. Leeb, L. K. Norford, and R. W. Cox. Nonintrusive load monitoring and diagnostics in power systems. *IEEE Transactions on Instrumentation and Measurement,* 57(7):1445–1454, 2008.

[4] D. B. Callaway, F. G. Tyzzer, and H. C. Hardy. Resonant vibrations in a water-filled piping system. *The Journal of the Acoustical Society of America,* 23(5):550–553, 1951.

[5] M. El-Raheb. Acoustic loading in straight pipes. *The Journal of the Acoustical Society of America,* 68(2):665–672, 1980.

[6] R. P. Evans, J. D. Blotter, and A. G. Stephens. Flow rate measurements using flow-induced pipe vibration. *Transactions of the ASME Journal of Fluids Engineering,* 126:280–285, 2004.

[7] M. P. Paidoussis. The canonical problem of the fluid-conveying pipe and radiation of the knowledge gained to other dynamics problems across applied mechanics. *Journal of Sound and Vibration,* 310:462–492, 2008.

[8] Z. Remscrim, J. Paris, S. B. Leeb, S. R. Shaw, S. Neuman, C. Schantz, S. Muller, and S. Page. Fpga-based spectral envelope preprocessor for power monitoring and control. In *Applied Power Electronics Conference and Exposition (APEC), 2010 Twenty-Fifth Annual IEEE,* pages 2194 – 2201, 2010.