ACOUSTIC AND SEISMIC SIGNAL PROCESSING FOR FOOTSTEP DETECTION

by

ROSS E. BLAND

B.S., Massachusetts Institute of Technology (2005)

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
June 2006

Author_____
Department of Electrical Engineering and Computer Science
May 20, 2006

Certified by_____
Charles E. Rohrs
Research Scientist, Digital Signal Processing Group
Thesis Supervisor

Accepted by_____
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Acoustic and Seismic Signal Processing for Footstep Detection

by

Ross E. Bland

Submitted to the Department of Electrical Engineering and Computer Science
on May 25, 2006, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

The problem of detecting footsteps using acoustic and seismic sensors is approached from three different angles in this thesis. First, accelerometer data processing systems are designed to make footsteps more apparent to a human operator listening to accelerometer recordings. These systems work by modulating footstep signal energy into the ear's most sensitive frequency bands. Second, linear predictive modeling is shown to be an effective means to detect footsteps in accelerometer and microphone data. The time evolution of the third order linear prediction coefficients leads to the classical binary hypothesis testing framework. Lastly, a new method for blindly estimating the filters of a SIMO channel is presented. This method is attractive because it allows for a more tractable performance analysis.

Thesis Supervisor: Charles E. Rohrs
Title: Research Scientist, Digital Signal Processing Group

Acknowledgments

First and foremost, I should thank my thesis supervisor, Dr. Charles Rohrs. I find it remarkable that Charlie has always had my best interest in mind. He has done his very best to teach me how to conduct quality research. I probably don't realize how fortunate I am to have a supervisor so willing to dive into any problem I bring to him. His enthusiasm for research has been a source of motivation and energy for me. Additionally, our college football conversations certainly helped to reduce the stress associated with graduate school at MIT.

I also should thank everyone in DSPG: Al, Eric, Petros, Sourav, Maya, Joe, Tom, Melanie, Joonsung, Al K., Dennis, Matt, and of course Zahi. I'd like to thank Al for encouraging me to share my ideas in group meetings, for teaching me how to think creatively, and for showing me how to communicate precisely. Eric is absolutely instrumental to the group. He always has the answer and if he doesn't he'll find out for you. Petros has stepped in and saved the day on numerous occasions (even from another continent). Sourav has been an awesome colleague. I am always amazed at how willing he is to drop what he is doing to help me with latex or to figure out some problem from research. I should thank Joe for his company in the office on the weekends. Tom was really great to field all of my random questions about acoustics and audio signal processing. Much thanks to Melanie for being the awesome problem solver she is. I thank Joonsung for providing me with a hero. I thank Al K. for the company in the group office at odd hours of the night. Dennis has been a great officemate and I have really enjoyed getting to know him better. I appreciate Zahi keeping the office fun. Lastly, I thank Matt for all of the laughs and good times. I can't imagine going through MIT without him.

I should also thank Dr. James Sabatier from the Department of Physics and Astronomy at The University of Mississippi for his help in data collection.

Finally, this research was supported in part by participation in the Georgia Institute of Technology MURI 2004 sponsored by Army Research Office (ARO) under award W911NF-04-1-0190 and was supported in part by participation in the Advanced Sensors Collaborative Technology Alliance (CTA) sponsored by the U.S. Army Research Laboratory under Cooperative agreement DAAD19-01-2-008.

Contents

1 Introduction **15**

1.1 Problems Considered 16

1.2 Previous Work 16

1.2.1 Kurtosis and Cadence 17

1.2.2 Short-time Fourier Transform Analysis 18

1.2.3 Footstep Feature Extraction for Personal Identification 20

1.3 Outline of the Thesis 22

2 Digital Processing to Aid the Listener in Footstep Detection **23**

2.1 The Data 23

2.2 Motivation for Processing 23

2.3 Digital Processing 24

2.3.1 Noise Removal 24

2.3.2 First Method: Shifting 11.025-22.05kHz Band Down in Frequency 27

2.3.3 Second Method: Placing High Footstep Energy in the Most Sensitive Frequency Range 27

2.4 Summary 30

3 Linear Predictive Modeling to Identify Footsteps **31**

3.1 Linear Prediction Background 31

3.2 Linear Predictive Modeling of Footstep Data 34

3.2.1 Microphone Data Noise Removal 34

3.2.2 Sectioning Data Using a Sliding Window 38

3.2.3 Model Order 39

3.3 Visualizing Accelerometer Footsteps in Three-Dimensional Coefficient Space 40

3.3.1 Identifying First and Second Footstep Components in Coefficient Space . . . 40

3.3.2 Window Length Effects 42

3.3.3 Consistency of Coefficient Behavior Across Different Footsteps 44

3.3.4 Linear Prediction Applied to Stealthy Footsteps 46

3.4 Microphone Footsteps in Three-Dimensional Linear Prediction Coefficient Space . . 50

3.4.1 Microphone Recordings of Regular Footsteps 51

3.4.2 Microphone Recordings of Stealthy Footsteps 55

3.5 Summary 57

4 Blind Channel Identification using Least Squares **59**

4.1 SIMO Channel 59

4.2 Least Squares Algorithm 61

4.3 Alternative Estimate: Pinning a Channel Coefficient 62

4.4 Comparison of the SVD and Pinning Channel Estimation Methods 63

4.5 Deriving the Distribution of the Channel Estimate 66

4.6 Summary 72

5 Conclusion **75**

5.1 Digital Processing to Aid the Listener in Footstep Detection 75

5.2 Linear Predictive Modeling to Identify Footsteps 76

5.3 Blind Channel Identification Using Least Squares 76

A Digital Processing Implementations in MATLAB **77**

A.1 Method 1 Processing 77

A.2 Method 2 Processing 78

B Available Data Files **81**

Bibliography **83**

List of Figures

1-1 Three acoustic and three seismic sensors monitoring an area for footsteps. 17

1-2 Houston's Detection Scheme 19

1-3 An example of the normalized STFT of footstep data given by Houston in [3]. 19

1-4 Mel-Cepstrum Analysis (from [7]). (a) Amplitude vs. Frequency Plot. (b) Fourth and fifth peak frequencies generated from multiple sections of the data for each of the five walkers. 20

1-5 Distribution of walking intervals for five subjects (from [7]). 21

1-6 Envelope of the footstep spectrum for five subjects (from [7]). 21

2-1 The ISO 226:2003 equal loudness contours shown as solid curves. The dashed contours represent the previous standard, ISO 226:1987. 24

2-2 Short-time Fourier Transform of a Section of Accelerometer Data Containing a Footstep 25

2-3 Noise Power Spectral Density Estimate Computed using Welch's Method 26

2-4 First Method Processing: Shifting of the 11.025-22.05kHz band down to 0-11.025kHz. 27

2-5 STFT of a "Stealthy" Footstep 28

2-6 Second Method Processing: Frequencies with high footstep energy are shifted to the frequencies most sensitive to the ear. 29

3-1 Linear Prediction Model 32

3-2 (a) All-Pole Signal Model. (b) Inverse of the All-Pole System. 34

3-3 Top: Section of microphone data absent of footsteps. Bottom: Prediction error energy corresponding to data section. 36

3-4 Microphone Noise Reduction System 37

3-5 Microphone data noise reduction. Top: Before filtering. Bottom: After filtering. . . . 37

3-6 Linear predictive model coefficients are computed for windowed sections of the data. 38

3-7 Prediction Error Energy vs. Model Order for Accelerometer Footstep 39

3-8 Prediction Error Energy vs. Model Order for Microphone Footstep 40

3-9 Accelerometer Recording of a Single Footstep 41

3-10 Identification of first and second footstep components in coefficient space. 42

3-11 Multiple window lengths on the same footstep. 43

3-12 For various window lengths, $\mathcal{V}$ is computed from a section of data absent of footsteps. 45

3-13 A comparison of coefficient paths for three different footsteps. Window size of $L = 5000$. 46

3-14 Accelerometer Footstep Data. Top: Regular Walking Style. Bottom: Stealthy Walking Style. 47

3-15 Comparison of regular and stealthy footstep linear prediction coefficient paths. Regions of the regular footstep are labeled with arrows. 48

3-16 Stealthy footstep recorded with accelerometer. First component is shown in red, second component in blue. 49

3-17 Coefficient path for stealthy accelerometer footstep. 50

3-18 Three different regular microphone footsteps. The corresponding coefficient paths are shown in Figure (3-19) 51

3-19 Linear prediction coefficient paths for three regular footsteps recorded with a microphone. Plots (a) and (b) are different views of the same data. 53

3-20 Section of microphone data used to investigate coefficient behavior between footsteps. 54

3-21 Coefficient path produced from the data section in Figure (3-20). 55

3-22 Coefficient paths generated from data sections in Figure (3-19) 56

3-23 Comparison of regular and stealthy footstep coefficient paths for microphone data. . 57

4-1 SIMO Channel 60

4-2 Six sensors measuring footsteps. 61

4-3 Performance Comparison of SVD and Pinning Methods 64

4-4 Performance Comparison of SVD and Pinning Methods with $h_1[K] = 0.1$ 65

4-5 The R distribution. (a) PDF estimates of $R(5)$, $R(10)$, and $R(20)$. (b) Variance of the distribution of $R(n)$ for n from 3 to 30. 69

4-6 Histogram plots comparing the estimated probability density for $\tilde{h}_1[0]$ and an R random variable scaled by $\frac{\sigma_{num}}{\sigma^2_{y_2}}$. (a) $N+1=3$ (b) $N+1=10$ (c) $N+1=30$ (d) $N+1=100$. 70

4-7 Plots comparing the histograms of $\frac{\mathbf{y}_2^T \mathbf{y}_n}{\sigma_{num}}$ to the PDF of a zero mean, unit variance Gaussian random variable. Each plot corresponds to a different value for N. (a) $N+1=3$ (b) $N+1=10$ (c) $N+1=30$ (d) $N+1=100$. 71

CHAPTER 1 *Introduction*

Many applications can be envisioned for a system that is able to detect human activity in some area. Probably the most obvious application is the detection of intruders in a secure region. It is often the case that a system capable of detecting pedestrians is desired. Any system to accomplish this task obviously needs a way to sense the pedestrian. Acoustic and seismic sensors are capable of measuring pedestrian activity and are often employed for this task for a number of reasons. Such sensor systems are inexpensive, passive (do not emit energy), and potentially easily installed.

In the case that the system should alarm only when a pedestrian is present, processing of the acoustic and seismic signals should make it possible for the system to discriminate footsteps from other acoustic and seismic sources such as highways, railroads, operating machinery, and trees and bushes swaying in the wind. As for indoor detection, heating and air conditioning units are a common noise source. The decision to alarm can be made by a human operator or by a signal processing algorithm. However, even when the system involves a human operator, signal processing should be performed to make the task easier and more error-free for the operator.

Discriminating footstep signals from noise sources is a challenging problem to solve with human operators or software. The footstep signal to noise ratio decreases rapidly with the distance between the sensors and the pedestrian. In addition, impulsive noise often looks and sounds like footsteps. Another complication is that footstep signals can vary greatly from one person to the next as well as one environment to the next. While this is a challenging problem, with high-performance sensors and sophisticated signal processing, it is hardly impossible.

A system that is successful in discriminating footsteps from noise sources may also be extended to solve similarly related problems. For example, it may be possible to identify a pedestrian based on the recorded footstep signals. In addition, evaluation of the acoustic and seismic signals produced by a person walking may aid in the diagnosis of gait-affecting diseases.

1.1 Problems Considered

There are three closely related problems explored in this thesis. One form of a human operator is a person listening to acoustic and seismic data in order to monitor an area for footsteps. The first problem is how can acoustic and seismic signals be processed to aid the listener in footstep detection. While it may seem odd that the operator is listening to seismic data, the seismic data is actually sampled at 44.1kHz (standard audio CD sampling rate) and contains significant energy in the audio frequency range.

There are no constraints on the processing that is performed on the acoustic and seismic data. The goal is simply to process the data in anyway that will enhance the listener's ability to detect footsteps and discriminate them from other acoustic and seismic sources. One issue with this problem is the difficulty in measuring the performance of processing methods. Performance can only be measured qualitatively by listening to the original data file and processed data file and comparing the two.

The second problem considered in this thesis is the characterization of footsteps recorded with acoustic and seismic sensors. In order to discriminate footsteps from similar acoustic and seismic noise sources, footstep signal feature extraction must be developed. The extracted features can be employed in a decision rule that states when the presence of a footstep is declared. A possibly more difficult but similarly related problem considers using the extracted features to infer qualities of the walker such as height, weight, or even identity.

Lastly, this thesis looks at the problem of finding alternative methods for blind channel identification. A method that is computationally inexpensive and allows for a tractable analysis of performance is desired. Blind channel identification is important for footstep detection in that blind channel identification is useful for sensor fusion. Sensor fusion is the process of combining signals from multiple sensors. Figure (1-1) depicts a situation in which three accelerometers (seismic sensors) and three microphones (acoustic sensors) are monitoring an area for footsteps. Blind channel identification can be used in the process of combining all six signals to produce a seventh signal that is then processed to determine if an intruder is present.

1.2 Previous Work

The problem of detecting footsteps using acoustic or seismic sensors has been studied before. In this section, the most relevant published material is discussed.

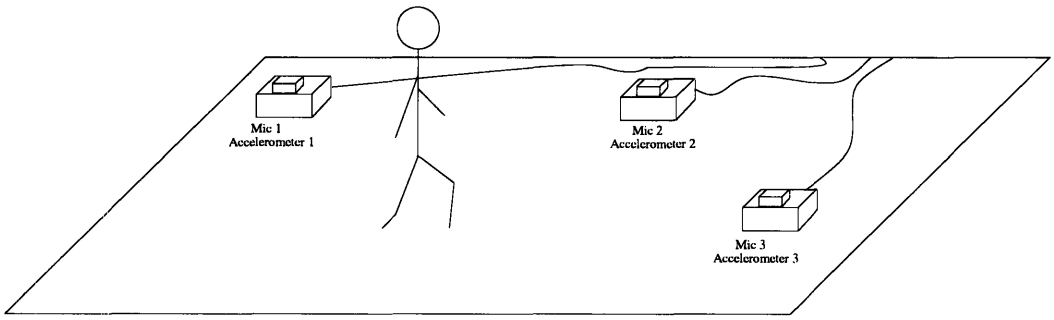


FIGURE 1-1: Three acoustic and three seismic sensors monitoring an area for footsteps.

1.2.1 Kurtosis and Cadence

In [8], Succi considers using the kurtosis and the cadence of seismic signals to detect footsteps. Kurtosis is the degree of peakedness of a distribution and is defined as a normalized form of the fourth central moment. While there are several flavors of kurtosis, Succi uses the kurtosis proper denoted β_2 and defined by

$$\beta_2 = \frac{\mu_4}{(\mu_2)^2} \tag{1.1}$$

where μ_i denotes the i^{th} central moment [9].

For N samples, Succi estimates the kurtosis of the amplitude distribution of a seismic signal $s[n]$ as

$$\hat{\beta}_2 = \frac{\frac{1}{N}\sum_{n=1}^{N}(s[n]-\mu)^4}{\left(\frac{1}{N}\sum_{n=1}^{N}(s[n]-\mu)^2\right)^2} \tag{1.2}$$

where μ is the sample average of the signal over the range $1 \leq n \leq N$. This estimate measures the shape of the signal. Scaling the data has no effect on the estimate.

Since kurtosis is a measure of the peakedness of a distribution, the kurtosis of a section of data is increased by impulsive events. Succi reasons that kurtosis is a good indicator of footsteps since footsteps appear as sharp spikes in the data unlike several seismic noise sources such as wind blowing over the ground or vehicle noise.

Succi computes $\hat{\beta}_2$ for overlapping four second sections of the data. He finds that for each of three recordings, background noise, light passenger vehicle, and heavy armored personnel carrier, the mean of $\hat{\beta}_2$ is approximately 3. Succi reports a mean

$\hat{\beta}_2$ of 6 for recordings of footsteps. The variance of $\hat{\beta}_2$ is 0.4 for ambient noise, 0.6 for the light passenger vehicle, 1.9 for the armored personnel carrier, and 2.5 for the footsteps.

Succi's kurtosis method of detecting footsteps essentially reduces the data to a single parameter that is then used to decide if a walker is present. This method is attractive because of its simplicity. However, this detection scheme is fairly limited since it uses only one parameter to characterize four second sections of the signal. A major problem is that any noise source that generates spikes in the data will be interpreted as a footstep.

Succi also offers an alternative single parameter footstep detection scheme. Succi reasons that the cadence of the signal can be used to detect footsteps. Each footstep produces a spike in the seismic signal. The time between spikes is measured and used to determine if a walker is present. This detection scheme also has its limitations. One of the most apparent problems is that the success of the cadence detection scheme is limited by noise sources that generate large amplitude values in the data.

1.2.2 Short-time Fourier Transform Analysis

Houston in [3] takes a somewhat different approach to footstep detection in which he utilizes the short-time Fourier transform (STFT). Houston's detection scheme focuses on the periodic arrival of footsteps instead of on the information in individual footstep signals.

A block diagram summarizing Houston's processing is shown in Figure (1-2). Houston finds that the 10-40Hz band contains most of the footstep energy and therefore passes the data through a bandpass filter with these cutoff frequencies. The absolute value of the signal is then taken before downsampling the signal to a sampling rate of 40Hz. The STFT of the data is then computed (26 second window, 95% overlap) and normalized using a split-window 2-pass normalizer (a conventional STFT analysis algorithm). Figure (1-3) shows the example of a resulting normalized STFT plot given by Houston.

A set of *ad hoc* decision rules are applied to the normalized Fourier transform of each 26 second window of data (one horizontal line in Figure (1-3). The presence of a walker is declared when the following criteria are met:

1. A primary frequency component must occur in the 0.5 to 3.0 Hz range.
2. The primary frequency component must have a second or third harmonic present.
3. The primary frequency component must be greater than 11dB over the noise level and the harmonic greater than 7dB, or vice versa.

Houston shows the results of applying his detection scheme to real footstep data recorded with seismic sensors in an outdoor environment. He concludes that his method shows promise but more data sets are needed to fully characterize its performance. Houston also mentions a couple of the potential pitfalls associated with his

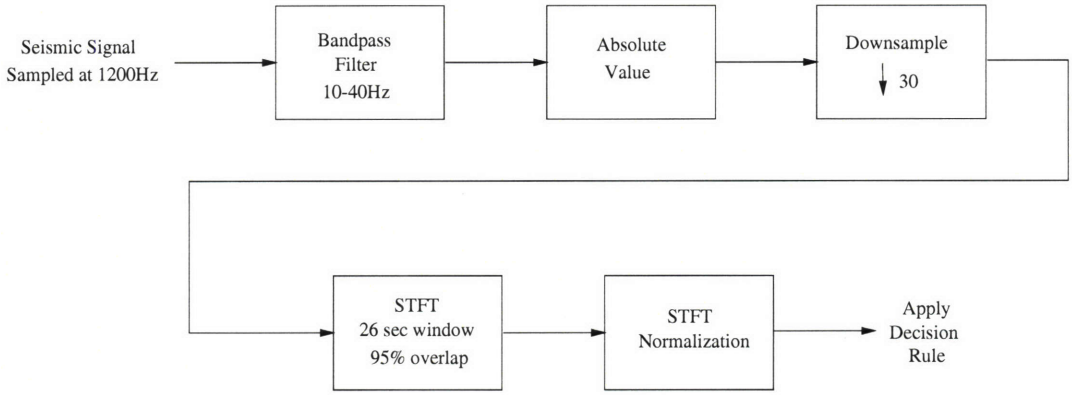


FIGURE 1-2: Houston's Detection Scheme

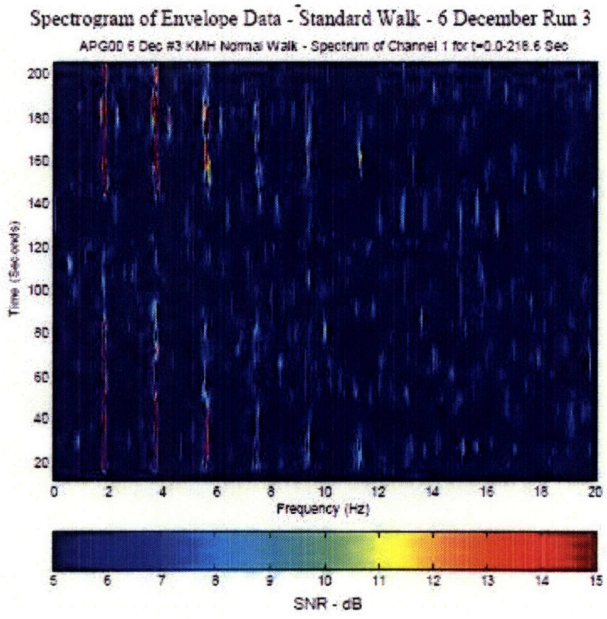


FIGURE 1-3: An example of the normalized STFT of footstep data given by Houston in [3].

detection scheme. He explains that it is possible for noise sources such as traffic on nearby roadways to mask the footstep signals. In addition, Houston mentions that stealthy walking, in which the walker takes soft footsteps, will not be detected with

his method.

1.2.3 Footstep Feature Extraction for Personal Identification

A problem closely related to footstep detection is that of using footstep signatures for identification purposes. In [7], Shoji considers using peak frequencies generated by a mel-cepstrum analysis, the time between successive footsteps, and samples of the spectrum envelope to identify walkers.

Mel-Cepstrum Analysis

Shoji applies a type of mel-cepstrum analysis to the footstep data to generate an amplitude versus frequency plot. The example plot given in [7] is shown in Figure (1-4(a)). Each curve in the plot corresponds to a different walker. The peak frequencies of the curves are used as feature parameters. For example, a plot given in [7] showing the fourth and fifth peak frequencies generated from multiple sections of the data for each of the five walkers is presented in Figure (1-4(b)). In this plot, subject C is the only subject that can be clearly discriminated from the others.

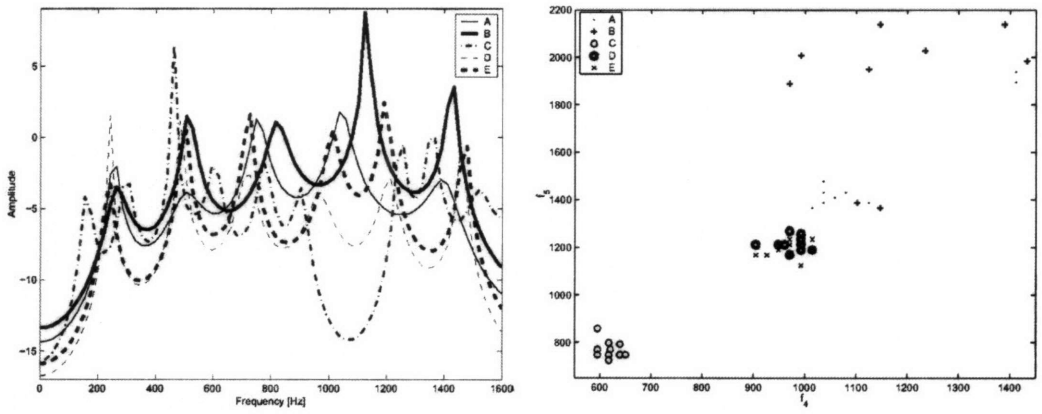


FIGURE 1-4: Mel-Cepstrum Analysis (from [7]). (a) Amplitude vs. Frequency Plot. (b) Fourth and fifth peak frequencies generated from multiple sections of the data for each of the five walkers.

Walking Intervals

The second feature parameter Shoji considers is the time between footsteps. The time between footsteps is measured by estimating the pitch using an autocorrelation method. A figure given by Shoji in [7] is shown in Figure (1-5). This figure shows

the distribution of the measured walking intervals for five subjects. In this plot, only subject B is easily discriminated from the others.

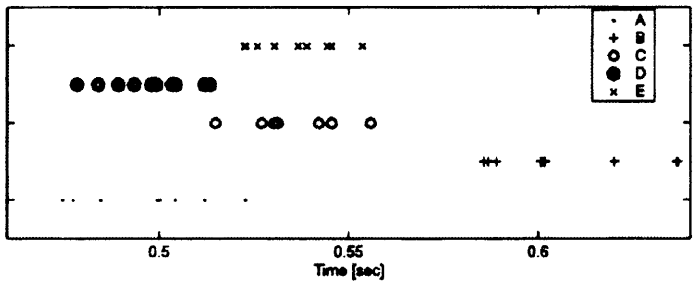


FIGURE 1-5: Distribution of walking intervals for five subjects (from [7]).

Envelope of the Footstep Spectrum

Lastly, Shoji considers using information in the envelope of the footstep spectrum to identify walkers. In [7] Shoji gives Figure (1-6) to illustrate the differences in the envelope of the footstep spectrum across different subjects. A vector of samples of the footstep spectrum envelope is computed and used to characterize footstep signals.

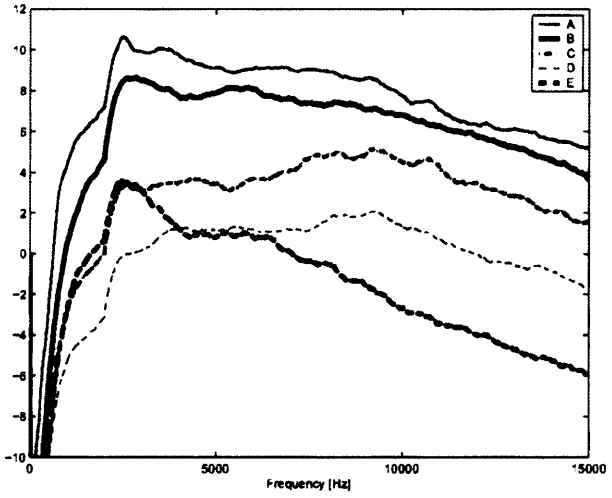


FIGURE 1-6: Envelope of the footstep spectrum for five subjects (from [7]).

The frequencies from the mel-cepstrum analysis, the walking interval times, and the samples of the footstep spectrum envelope are combined in a vector. Vector quantization is then used to cluster the parameters. This allows all three sources of information - the mel-cepstrum frequencies, the walking interval times, and the footstep spectrum envelope - to be simultaneously used to make a decision on the walker's identity.

1.3 Outline of the Thesis

This thesis explores the general problem of using acoustic and seismic sensors to detect footsteps. However, each chapter in the thesis considers a different aspect of the problem. Chapter 2 looks at digital processing that can help a human operator detect footsteps when listening to sensor data. More specifically, the short-time Fourier transform (STFT) is used to analyze accelerometer data (an accelerometer is a type of seismic sensor). The idea is that since the accelerometer data is sampled at 44.1kHz (CD quality sampling rate), the footsteps can be heard when playing the data through a speaker. With the use of STFT analysis, accelerometer data processing can be developed to make the footsteps more easily detected by the listener.

In Chapter 3, linear predictive analysis is considered for the automatic detection of footsteps in accelerometer and microphone data. Windowed sections of the sensor data are used to compute linear prediction coefficients. The feasibility of using the coefficients of a linear prediction model to discriminate footsteps from noise artifacts in the data is established.

Lastly, Chapter 4 explains the least squares solution to blind channel identification. Chapter 4 also investigates an alternative method to perform blind channel identification. This alternative method is attractive because it allows for a recursive solution. Additionally, this alternative method was motivated by the potential for a more tractable performance analysis.

CHAPTER 2 *Digital Processing to Aid the Listener in Footstep Detection*

A person walking through a room generates both acoustic and seismic vibrations mostly due to footsteps. These vibrations can be recorded with microphones and accelerometers, respectively. These recorded signals can be monitored to detect the presence of an intruder. This chapter presents digital processing that can be performed on the recorded signals in order to make the footsteps more apparent to a human listener.

2.1 The Data

The data discussed in this chapter was collected by placing a microphone and accelerometer in the center of a room. A person then walked in a circle (approximate diameter of 6 feet) around the sensors. Two data sets were recorded: one in which the person walked "regularly" and another in which the person walking tried to make a minimal amount of noise. The two data sets are identified as "regular" and "stealthy." The recorded data was digitized at 16 bits with a 44.1kHz sampling rate.

2.2 Motivation for Processing

Figure (2-1) shows two standards for equal loudness contours for listening in free sound fields. The solid contours are the most recent standards. As can be seen from these contours, the human ear is more sensitive to sounds near 3kHz than sounds at

lower or higher frequencies. The contours in Figure (2-1) were produced by testing subjects 18-25 years of age with normal hearing.

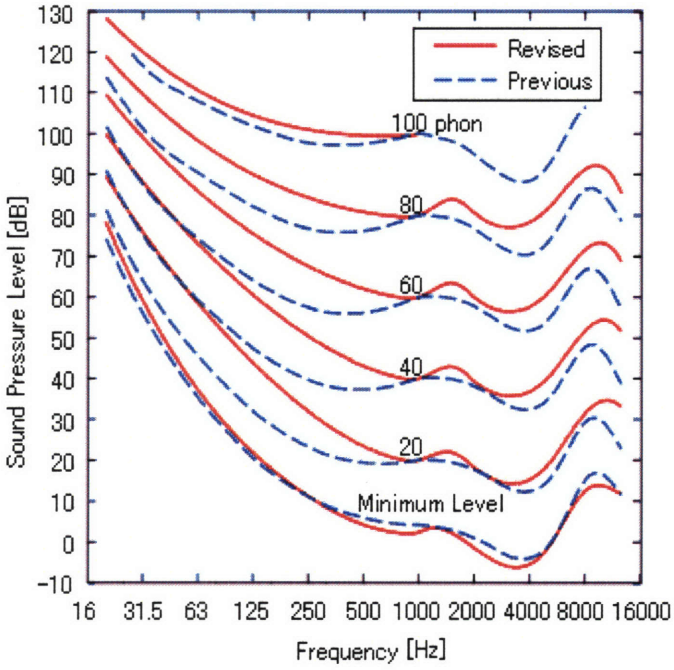


FIGURE 2-1: The ISO 226:2003 equal loudness contours shown as solid curves. The dashed contours represent the previous standard, ISO 226:1987.

Considering this filtering performed by the ear, it makes sense that moving the recorded data from low-sensitivity frequency ranges to high-sensitivity frequency ranges could aid the listener in detecting footsteps. The processing discussed in this chapter is based on this idea.

2.3 Digital Processing

2.3.1 Noise Removal

Before shifting low-sensitivity frequency ranges to high-sensitivity ranges, an attempt at noise removal should be made. The most common sources of noise in an indoor environment are the heating and cooling systems and the electrical noise in the sensor. Both of these noise sources are stationary over sufficiently short time intervals. The first step in removing the noise is to examine the frequency content of the data. At frequencies where the noise power is significantly greater than the signal (footstep) power, the data should be attenuated. The frequency content of the noise

can be estimated by performing spectral analysis on data recorded in the absence of footsteps. The frequency content of the footsteps can be estimated by using the short-time Fourier transform (STFT) on sections of data that are known to contain footsteps.

The STFT of a section of accelerometer data containing a footstep can be seen in Figure (2-2). The STFT was taken by windowing the data with a Hamming window of length 200 samples. The windowed segments overlapped 150 samples and the 1,024 point Fast Fourier Transform was used. From the figure, it is apparent that a footstep consists of two components. The first contains a wide range of frequency content in a short amount of time. The second component contains frequency content mostly in the range of 6-16kHz and spread over about 60 milliseconds. Considering the time-frequency nature of these two components as well as a time separation of about 100 milliseconds, it seems likely that the first component is the heel touching down and the second component is the foot coming to a halt against the floor.

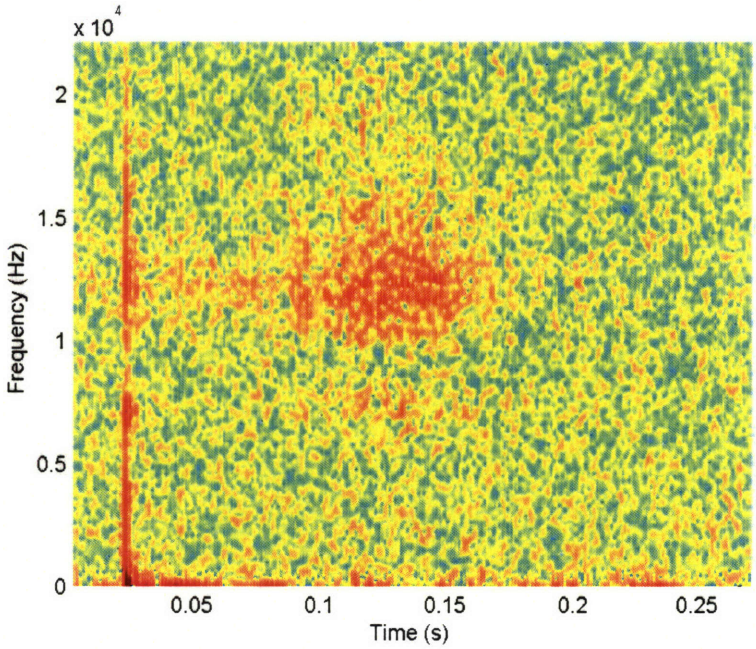


FIGURE 2-2: Short-time Fourier Transform of a Section of Accelerometer Data Containing a Footstep

Since the noise is mostly stationary, spectral analysis can be used to determine its frequency content. The curve labeled "pre-filtered" in Figure (2-3) is the power spectral density estimate of the noise computed from a section of the data absent of footsteps.

This estimate is computed using Welch's method. More specifically, a Hamming window of length 500 samples is used to compute modified periodograms. An overlap of 250 is used and the resulting periodograms are averaged to obtain the power spectral density estimate.

As can be seen from this estimate, the noise is fairly evenly distributed over the entire frequency range (0-22.05kHz) except for a spike at low frequencies. In filtering out the noise, there is a fundamental tradeoff between removing the noise and altering the signal of interest. Considering the frequency content of the footstep and as well as that of the noise, it makes sense to attenuate the data in the range from 0 to 500 Hz, since this is where the spike in the noise is located. The noise spectrum after applying a 100^{th} order FIR high-pass filter with cut-off frequency at about 500Hz to the data is labeled "post-filtered" in Figure (2-3).

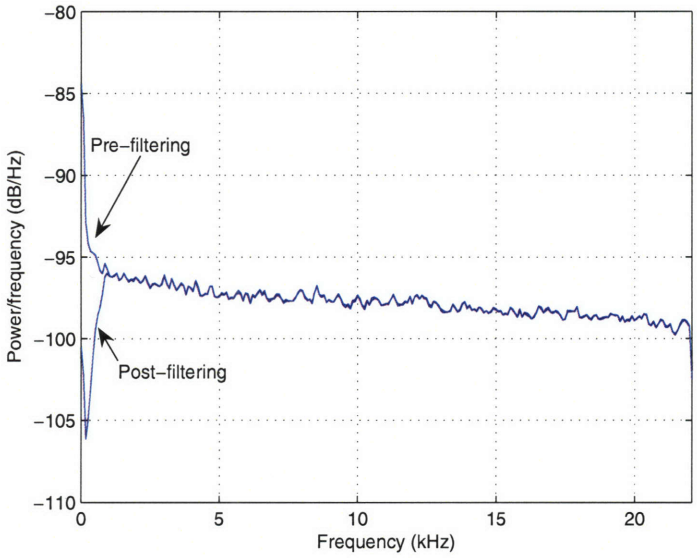


FIGURE 2-3: Noise Power Spectral Density Estimate Computed using Welch's Method

It is also important to consider the effect that this high-pass filter has on the footstep component of the data. The second component of the footstep, which is located above 6kHz, is not affected by the filtering. The first component, which contains energy below 500Hz, is affected by the noise removal filtering. However, as can be seen in Figure (2-3), 0-500Hz is a small fraction of the frequency range spanned by the first component. Thus, attenuating the 0-500Hz portion does not have a drastic effect on the first component of the footstep.

2.3.2 First Method: Shifting 11.025-22.05kHz Band Down in Frequency

The first approach to make the footsteps more distinct to the listener is to simply take the 11.025-22.05kHz section of the data and shift it down into the 0-11.025kHz range. This operation is shown in Figure(2-4). HPF and LPF denote high-pass filter and low-pass filter, respectively. The cosine used in the frequency shifting has an amplitude of 2 since multiplication by a cosine shifts a signal's spectrum as well as scales it by $\frac{1}{2}$. The 0-11.025kHz band of the output signal contains the superposition of the input signal's 0-11.025kHz and 11.025-22.05kHz bands. The output signal is band-limited to 11.025kHz.

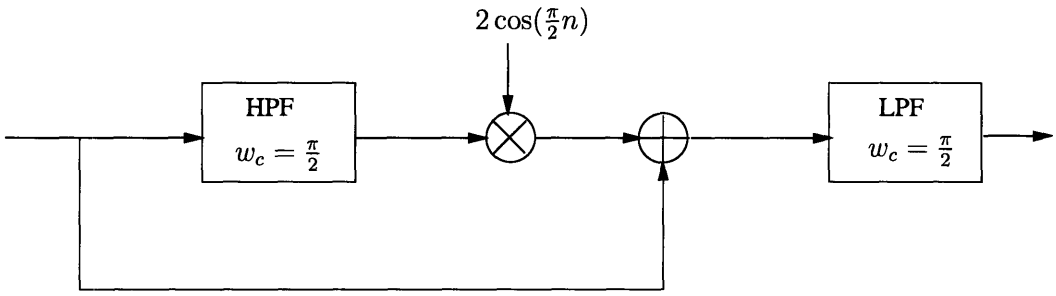


FIGURE 2-4: First Method Processing: Shifting of the 11.025-22.05kHz band down to 0-11.025kHz.

The original accelerometer data file, as well as the processed data file are available for comparison [1]. The actual implementation of the processing in Figure (2-4) can be found in Appendix A. The processing in Figure (2-4) was also applied to the accelerometer data recorded with a "stealthy" walker. The original recording of the stealthy walker as well as the processed recording are also available [1]. A description of the files available from [1] is given in Appendix B.

Listening to the data files before and after processing reveals that the processing of the "regular" walking had a noticeable affect on the recording. It is quite possible that for most listeners this processing can make the detection of footsteps easier. A noticeable affect is also heard in the "stealthy" case. However, in this case it seems that the processing does not make the footsteps significantly easier to detect.

2.3.3 Second Method: Placing High Footstep Energy in the Most Sensitive Frequency Range

The second approach involves taking frequency bands with high footstep energy and shifting them to the frequencies that are most sensitive to the ear. As can be seen in Figure (2-1), 1-6kHz is a region of high-sensitivity for the human ear. Figure (2-2) shows the two components of the footstep signal. The first component is spread fairly evenly over the entire frequency range while the second component is mostly in the

range of 6-16kHz. Figure (2-5) shows the STFT of a "stealthy" footstep. As can be seen in this figure, the first component of the footstep is nearly eliminated but the second component still has a strong presence. Therefore, it seems the 6-16kHz band contains significant footstep energy regardless of the type of walking. Thus, it seems plausible that shifting this band into the high-sensitivity 1-6kHz region could aid the listener in detecting footsteps.

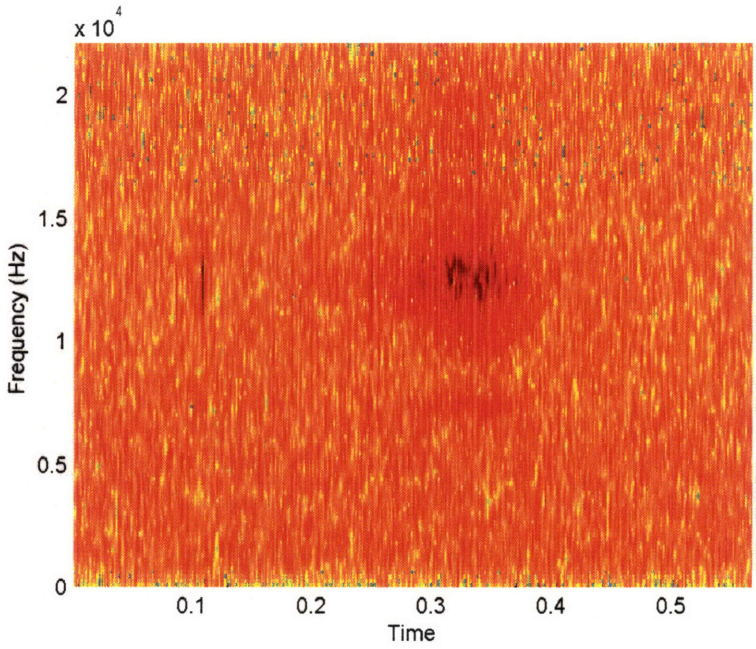


FIGURE 2-5: STFT of a "Stealthy" Footstep

A linear scaling of the frequency axis cannot be used to pack the 6-16kHz band into the 1-6kHz band since this would result in expansion of the data in the time domain. In order to pack the 10kHz band into the 5kHz space, the 10kHz band is decomposed into two 5kHz pieces. These two components are then both placed in the 1-6kHz region.

The system that performs this processing is shown in Figure (2-6). The top signal path generates $g[n]$, which contains the input signal's 6-11kHz content located at 1-6kHz. The bottom signal path generates $v[n]$, which contains the input signal's 11-16kHz content located at 1-6kHz. The output is the sum of the input signal, $g[n]$ and $v[n]$.

An implementation of the processing shown in Figure (2-6) can be found in Appendix A. Accelerometer data files demonstrating the affect of this processing are available

[1]. Listening to these files reveals that the footsteps are noticeably more distinct in the processed files for both regular and stealthy walking.

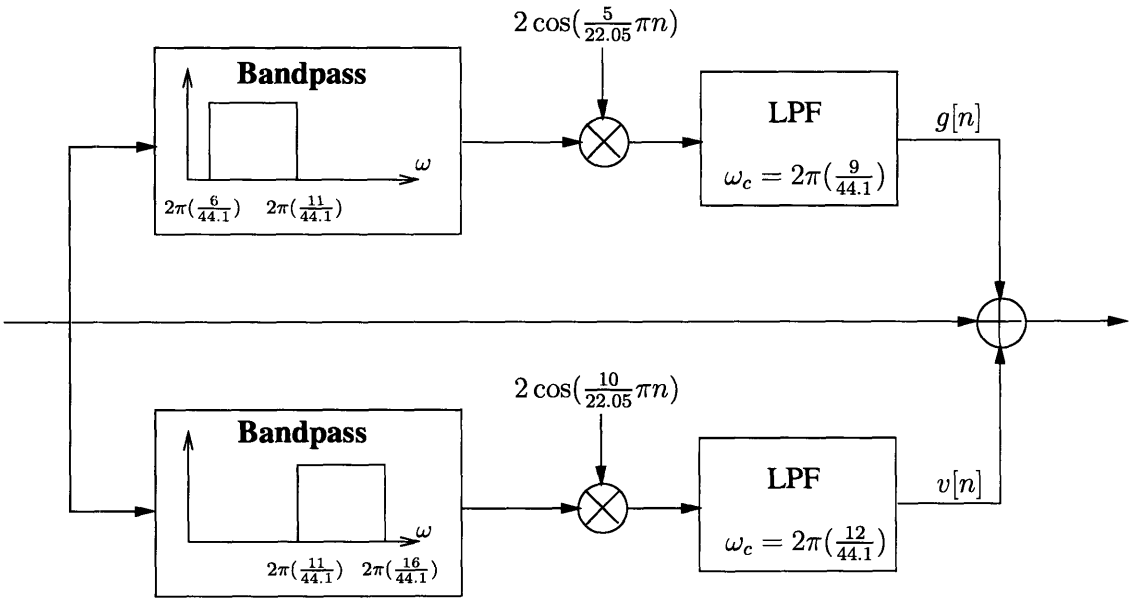


FIGURE 2-6: Second Method Processing: Frequencies with high footstep energy are shifted to the frequencies most sensitive to the ear.

2.4 Summary

This chapter explains how equal loudness curves for the human ear can be used to design data processing systems to enhance an operator's ability to detect footsteps when listening to accelerometer data. Two different methods are proposed for making footsteps in accelerometer data more distinct to the listener. The first method of processing simply shifts the 11-22kHz frequency band of the data down in frequency. This makes footstep energy originally located in the higher frequencies more easily heard. The second method of processing takes the high energy components of the footstep signal and shifts them into the ear's most sensitive frequency band. Before applying either method to accelerometer data, the noise is first reduced using a high-pass filter. The design of this high-pass filter is also discussed in this chapter.

CHAPTER 3

Linear Predictive Modeling to Identify Footsteps

Linear prediction can be used to identify footsteps in accelerometer and microphone recordings. Sections of the data are used to compute the coefficients of a linear prediction filter. Footstep signals in the data show up as changes in the filter coefficients. The values the coefficients take on over time can be used to discriminate a person walking from some other type of acoustic or seismic source. This chapter begins with a description of the linear predictive model as well as how the filter coefficients are computed.

3.1 Linear Prediction Background

Forward prediction, a well studied problem in time series analysis, considers the problem of predicting the future value of a stationary discrete-time stochastic process, given a collection of past sample values of the process. For example, a p^{th} order predictor uses the sample values $s[n-1], s[n-2], ..., s[n-p]$ to estimate $s[n]$. In general, the predictor can be written as some function $f(\cdot)$ of the given sample values $s[n-1], s[n-2], ..., s[n-p]$ as follows:

$$\hat{s}[n] = f(s[n-1], s[n-2], ..., s[n-p]). \tag{3.1}$$

A common simplification is to restrict $f(\cdot)$ to be a linear function of the past sample values, i.e.

$$\hat{s}[n] = \sum_{k=1}^{p} a_k s[n-k]. \tag{3.2}$$

The problem is then referred to as forward linear prediction [2].

There exists a deterministic version of the forward linear prediction problem and this is the linear prediction modeling that is used in this chapter. This problem can be framed as follows:

Deterministic Linear Prediction Problem Statement

Given a finite-duration signal $s[n]$, choose the filter coefficients $a_1, a_2, ..., a_p$ in Figure (3-1) such that the total energy in the output signal, $\sum(e[n])^2$, is minimized.

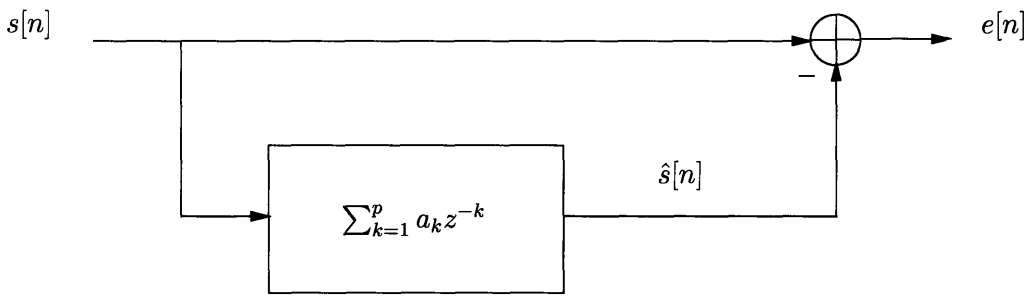


FIGURE 3-1: Linear Prediction Model

In Figure (3-1), $\hat{s}[n]$, the output of the predictor filter, is the linear estimate of $s[n]$ based on the past p values of $s[n]$. The filter coefficients are chosen to minimize the total energy in the prediction error signal. Assuming $s[n]$ is a finite-duration signal defined for $0 \leq n \leq N$, the prediction error signal energy can be written as

$$\mathcal{E} = \sum_{n=0}^{N} \left(s[n] - \sum_{k=1}^{p} a_k s[n-k] \right)^2. \tag{3.3}$$

The error signal energy can be minimized by differentiating with respect to each coefficient and setting the resulting equations to zero.

$$\begin{aligned} \frac{\partial \mathcal{E}}{\partial a_i} &= \sum_{n=0}^{N} -2 \left(s[n] - \sum_{k=1}^{p} a_k s[n-k] \right) s[n-i] \qquad i = 1, 2, ..., p. \\ &= 0 \end{aligned} \tag{3.4}$$

This set of equations can be rearranged to obtain

$$\sum_{n=0}^{N}\sum_{k=1}^{p} a_k s[n-k]s[n-i] = \sum_{n=0}^{N} s[n]s[n-i] \qquad i = 1,2,...,p. \tag{3.5}$$

Defining the deterministic autocorrelation as

$$\phi_{ss}[m] = \sum_{n=-\infty}^{\infty} s[n+m]s[n] \tag{3.6}$$

allows the set of p equations to be rewritten as

$$\sum_{k=1}^{p} a_k \phi_{ss}[i-k] = \phi_{ss}[i] \qquad i = 1,2,...,p. \tag{3.7}$$

These equations yield the predictor filter coefficients and are known as the Yule-Walker or Autocorrelation Normal Equations.

The predictor filter coefficients are often used to represent the signal $s[n]$ by a finite number of parameters. This may be better understood by considering the relationship between linear prediction and all-pole modeling, a form of parametric signal modeling.

In all-pole modeling, a signal is modeled as the impulse response of a linear time-invariant (LTI) system with p finite poles. This is depicted in Figure (3-2(a)). The parameters b_k determine the locations of the system's poles.

In order to simplify the math, the b_k's are often determined by considering the inverse problem shown in Figure (3-2(b)). With this approach, the b_k's can found by minimizing the total energy in the signal $g[n] - \delta[n]$. If s[n] is causal, i.e. $s[n] = 0$ for $n < 0$, it turns out minimizing $g[n] - \delta[n]$ results in Equations (3.7) and $b_k = a_k$ for $k = 1,2,...p$. Thus, the linear predictor coefficients can also be interpreted as parameters used to model $s[n]$ as the impulse response of a p^{th} order all-pole system.

When modeling a signal using linear prediction or all-pole modeling, the model order p must be chosen. Regardless whether the problem is framed as linear prediction or all-pole modeling, the most direct approach to choosing the order is to use the linear prediction coefficients or all-pole model parameters as the a_k's in Figure (3-1) and examine the prediction error energy for various model orders. This can be done by plotting the prediction error energy for the p^{th} order model

$$\mathcal{E}_p = \sum_{n=-\infty}^{\infty} \left(s[n] - \sum_{k=1}^{p} a_k^{(p)} s[n-k] \right)^2. \tag{3.8}$$

versus the model order p. Note that the prediction error energy for the 0^{th} order

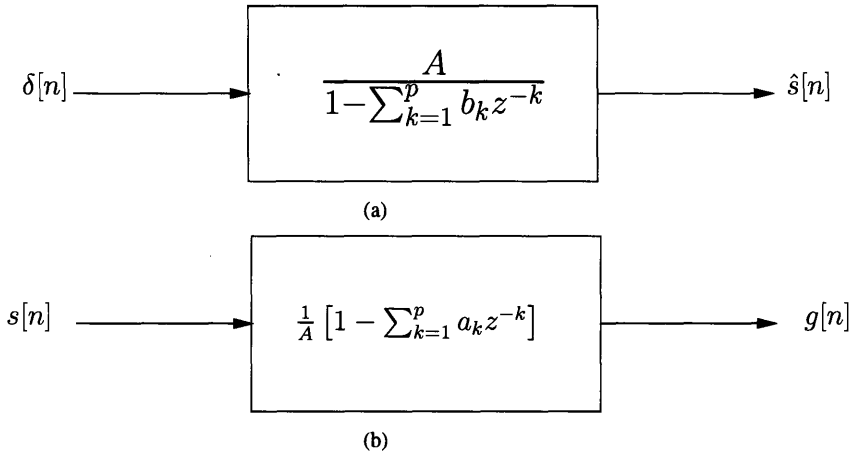


FIGURE 3-2: (a) All-Pole Signal Model. (b) Inverse of the All-Pole System.

model, $\mathcal{E}_0$, is just the total energy in the signal being modeled, $\sum(s[n])^2$. If the linear predictive or all-pole is a perfect model for $s[n]$ then the energy in the prediction error will go to zero for some p. It is often the case that there is some value of p above which increasing p has little or no effect on the prediction error energy. This value of p is considered an efficient choice for the model order [6].

3.2 Linear Predictive Modeling of Footstep Data

In this chapter, the same accelerometer and microphone data discussed in Chapter 2 is used to explore the idea of using linear predictive modeling to detect footsteps. Both the "regular" data set and the "stealthy" data set are discussed in this chapter. As mentioned in Chapter 2, the regular data set was produced by walking in a circle around the microphone and accelerometer while the stealthy data set corresponds to the walker trying to walk softly producing as small a footstep signal as possible. This section discusses a method for removing noise in the data as well as how the data is broken into sections by windowing. In addition, this section looks at model order, an important variable that must be chosen before calculating the linear prediction coefficients for a signal.

3.2.1 Microphone Data Noise Removal

Noise removal for the accelerometer data can be performed with high-pass filtering and is discussed in Chapter 2. The high-pass filter is chosen by comparing the noise power spectral density and the power spectral density of a section of data containing a footstep. The high-pass filter cut-off frequency is chosen such that the data are attenuated at frequencies where the noise power spectral density is significantly greater

than the footstep power spectral density.

Noise reduction should also be performed on the microphone data before applying linear predictive modeling for footstep detection. Accelerometer data noise removal uses a high-pass filter since most of the accelerometer noise energy is at low frequencies. The same is true for the microphone data noise and therefore, a high-pass filter is also used for microphone noise removal. However, a different approach is taken to design the noise removal filter for the microphone data.

The noise sections of the accelerometer data before high-pass filtering, i.e the sections of the pre-filtered data absent of footsteps, are not well modeled by linear prediction. That is, for all model orders p the linear prediction error signal, the output signal in Figure (3-1), always has energy that is a significant portion of the modeled signal's total energy, $\sum(s[n])^2$. This is not the case for microphone data. A section of noisy microphone data is shown in the top pane of Figure (3-3) Using this section of noise, the prediction error energy for various model orders p is plotted in dB in Figure (3-3).

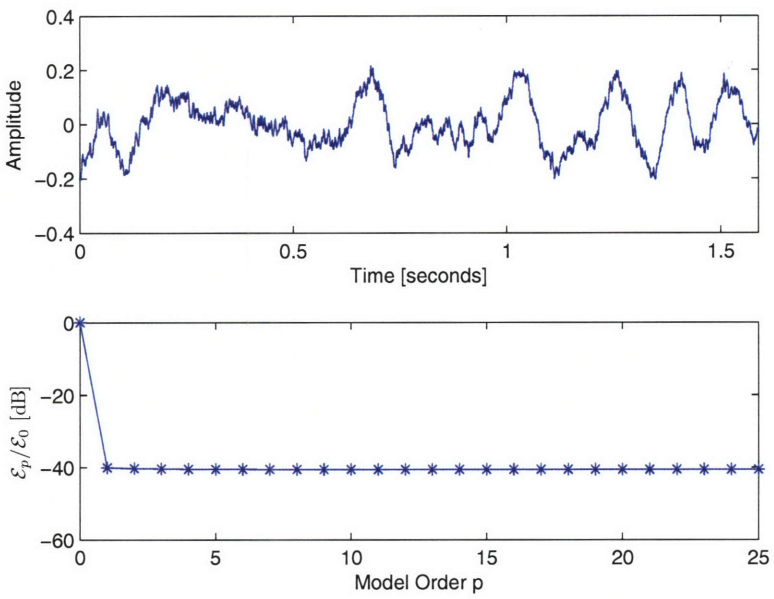


FIGURE 3-3: Top: Section of microphone data absent of footsteps. Bottom: Prediction error energy corresponding to data section.

As can be seen in Figure (3-3), the first order model produces error energy 40dB below the total noise section energy (the zeroth order error energy). The reason for such great performance with just a first order model can be understood by looking at the time domain plot of the noisy section of data as well as the first order model coefficient. For the first order model, $a_1 = 1$. This means the overall system from microphone noise to linear prediction error can be described by the system function $H(z) = 1 - z^{-1}$ or difference equation $y[n] = x[n] - x[n-1]$. This system is shown in Figure (3-4) and is a simple differentiator or high-pass filter. This filter makes sense after realizing that the energy in the microphone noise shown in Figure (3-3) is mainly due to low frequency components at approximately 5Hz. A simple high-pass filter such as that shown in Figure (3-4) will strongly attenuate these low frequency components. The ability of the system in Figure (3-4) to decrease noise energy suggests that microphone noise could be reduced by passing the microphone data through this system.

The system shown in Figure (3-4) decreases the noise energy but at the same time alters the footstep signals. This system is an effective method of noise reduction only if it removes enough noise to outweigh its effects on the footstep signals. The two signals in Figure (3-5) demonstrate the effect the noise removal filtering has on microphone data. The top signal is the pre-filtered microphone data and the bottom

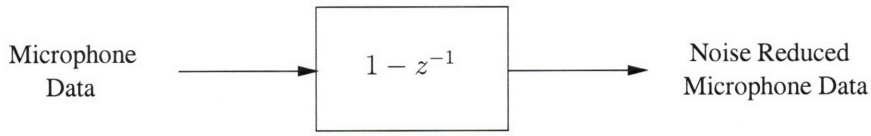


FIGURE 3-4: Microphone Noise Reduction System

signal is the same data post-filtering. At least from a visual perspective, the suggested noise removal filtering is a success in that the footsteps are much more apparent after filtering.

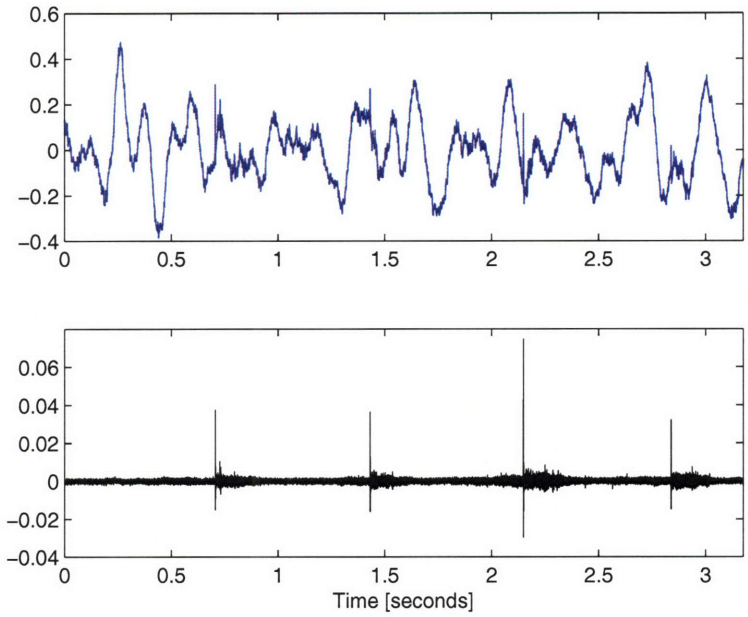


FIGURE 3-5: Microphone data noise reduction. Top: Before filtering. Bottom: After filtering.

3.2.2 Sectioning Data Using a Sliding Window

A sliding window is used to apply linear predictive modeling to the data. The linear prediction coefficients are computed for each windowed section of data. The idea is that footsteps can be detected by monitoring the coefficients as they are computed for each section of data. A footstep is declared if the coefficients take on values characteristic of a footstep, or if the trajectory of the coefficients is characteristic of a footstep. There are two parameters that affect the windowing of the data. These parameters are shown in Figure (3-6). Values need to be chosen for the window length, L, as well as the amount of overlap between subsequent windows, M. In addition, the order p of the linear predictive model computed for each section of data needs to be chosen.

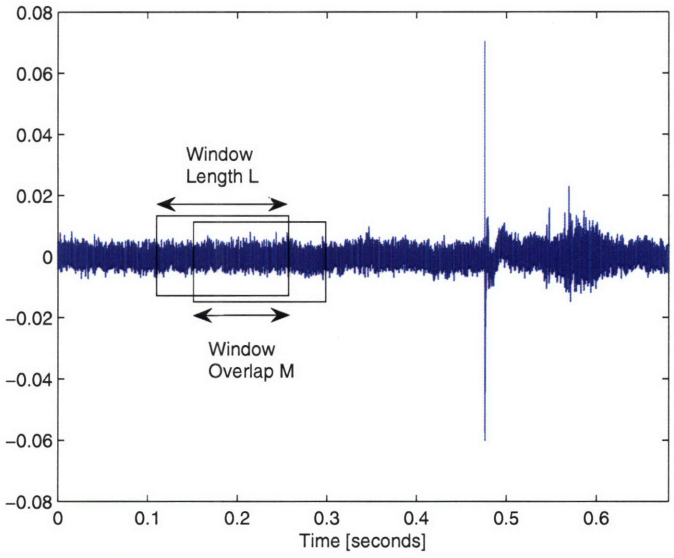


FIGURE 3-6: Linear predictive model coefficients are computed for windowed sections of the data.

3.2.3 Model Order

A good first step to exploring linear prediction for footstep detection is to examine for various model orders how well a linear predictive model can model a footstep signal. This is done by looking at the prediction error energy. Figure (3-7) shows the prediction error energy plotted versus the model order for a $7,000$ sample section of the accelerometer data. The $7,000$ sample section of data is just long enough to include one footstep. As seen in Figure (3-7), the most significant drops in error energy occur when p increases from 1 to 3. In addition, the plot shows that when modeling this entire footstep with linear prediction, near best possible performance is attained with a model order of 16. This means that only small gains are made by increasing the model order beyond 16 when characterizing this footstep with linear prediction coefficients.

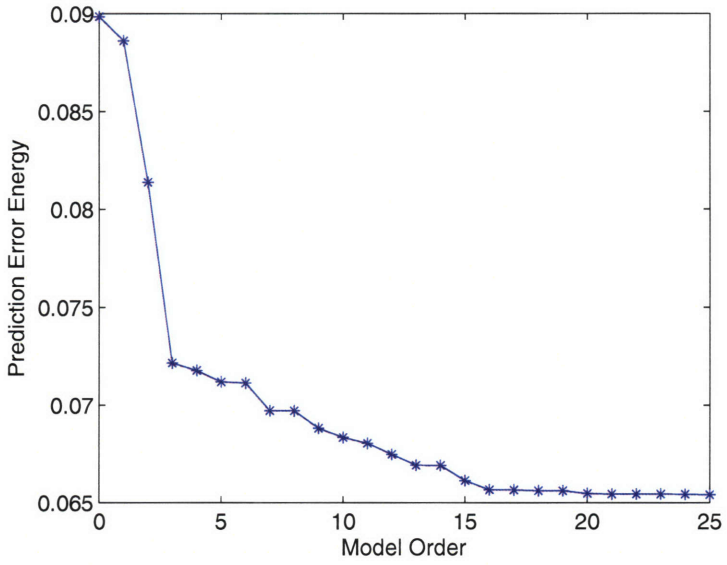


FIGURE 3-7: Prediction Error Energy vs. Model Order for Accelerometer Footstep

Figure (3-8) shows the prediction error energy plotted versus model order for a section of microphone data just long enough to include one footstep. According to this plot, $p = 3$ could be considered an efficient choice for model order when modeling a footstep in microphone data.

In Figure (3-7) one could also argue that $p = 3$ is an efficient choice for model order since the largest drops in error energy occur when going from $p = 1$ to $p = 3$. Using a third order linear predictive model has the added advantage that the coefficients can be viewed as a vector in three-dimensional space. It is for this reason that in this

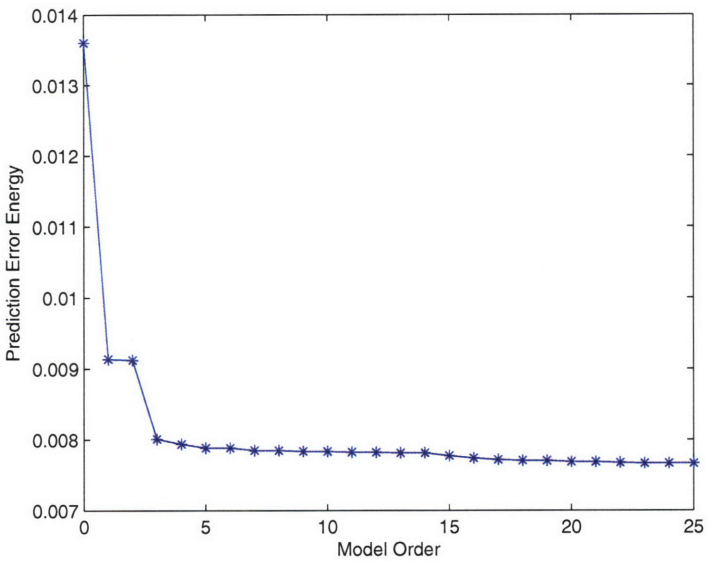


FIGURE 3-8: Prediction Error Energy vs. Model Order for Microphone Footstep

chapter a third order model is primarily used for analysis of the accelerometer and microphone data.

3.3 Visualizing Accelerometer Footsteps in Three-Dimensional Coefficient Space

The third order linear predictive model has the advantage that the movement of its coefficients over time can be easily visualized in three-dimensional space. In this section, a third order linear predictive model is used to analyze accelerometer footstep signals. Several issues, such as window length effects and consistency of coefficient behavior across different footsteps, are considered. First, the regular footsteps are analyzed and then the differences between the linear prediction results seen in the regular and stealthy footsteps are discussed. All accelerometer data considered in this section have been filtered for noise reduction using the high-pass filter discussed in Chapter 2.

3.3.1 Identifying First and Second Footstep Components in Coefficient Space

An accelerometer recording of a footstep is shown in Figure (3-9). In Chapter 2, two components of a footstep signal could be seen in the short-time Fourier transform plots. These two components can also be seen in Figure (3-9). The part of the footstep signal referred to as the "first component" in Chapter 2, is shown in black.

The "second component" is shown in blue. The noise regions are shown in red. The edges of the regions in Figure (3-9) were chosen simply by examining the recording visually and noting approximately where the behavior of the signal transitioned.

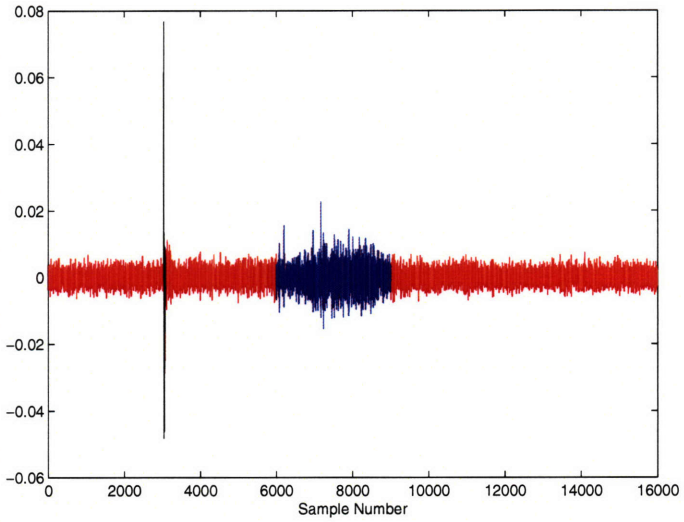


FIGURE 3-9: Accelerometer Recording of a Single Footstep

As can be seen in Figure (3-9), the footstep is about 7,000 samples long (160ms). Figure (3-10) shows how the prediction coefficients change as a 1,000 sample long window slides over the data with an overlap of 900 samples. When the window overlaps the region in Figure (3-9) colored in black, the resulting coefficients are plotted as black circles. Likewise, when the window overlaps the blue region in Figure (3-9), the coefficients are plotted as blue asterisks. The same goes for coefficients plotted as red diamonds.

Since the window is 1,000 samples long and there are approximately 2,200 samples between the first and second components, the resolution in the time domain is fine enough that the two components of the footstep can by analyzed individually in coefficient space. The coefficient plot shows significant separation in space among the three regions - noise, first and second footstep components. As the window slides along the data, the trajectory of the corresponding prediction coefficients can be followed. In some parts of the plot, the direction the coefficients move as the window slides in the direction of increasing time is indicated with arrowheads.

Before the window overlaps the footstep, the model coefficients are being computed from noise and the corresponding coefficients form a "noise" cluster consisting of the red diamonds in the figure. As the window slides far enough along in time that

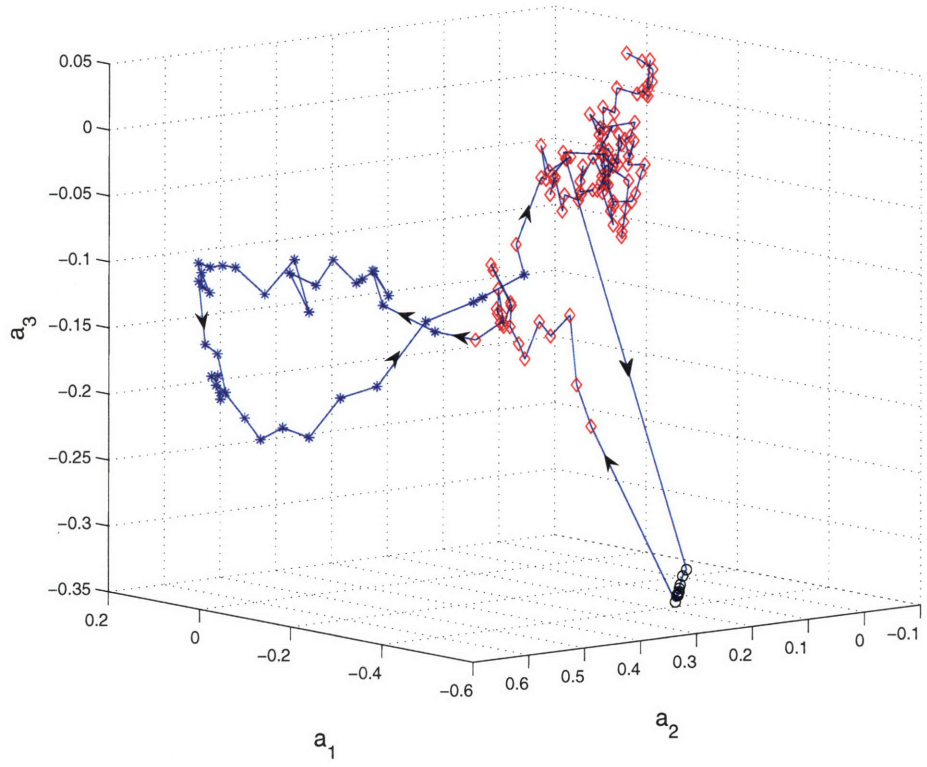


FIGURE 3-10: Identification of first and second footstep components in coefficient space.

the first component of the footstep signal is included, the coefficients move rapidly to a new cluster consisting of the black circles. Once the window slides far enough that it leaves the first component of the footstep, the coefficients move close to the noise cluster. As the window approaches and overlaps more of the second footstep component, the coefficients tend away from the noise cluster. The maximum distance from the noise cluster is achieved when the window fully covers the second footstep component. As the window leaves the second footstep component the coefficients head back to the noise cluster.

3.3.2 Window Length Effects

The window used to generate Figure (3-10) is 1,000 samples long. Similar plots can be created using longer windows to see the effect window length has on the parameter values and their trajectories. The same footstep seen in Figure (3-9), is used to create Figure (3-11). Three different window lengths are used: $L = 1000, 5000$, and 10000.

The corresponding overlaps are $M = 900, 4900$, and 9900, respectively.

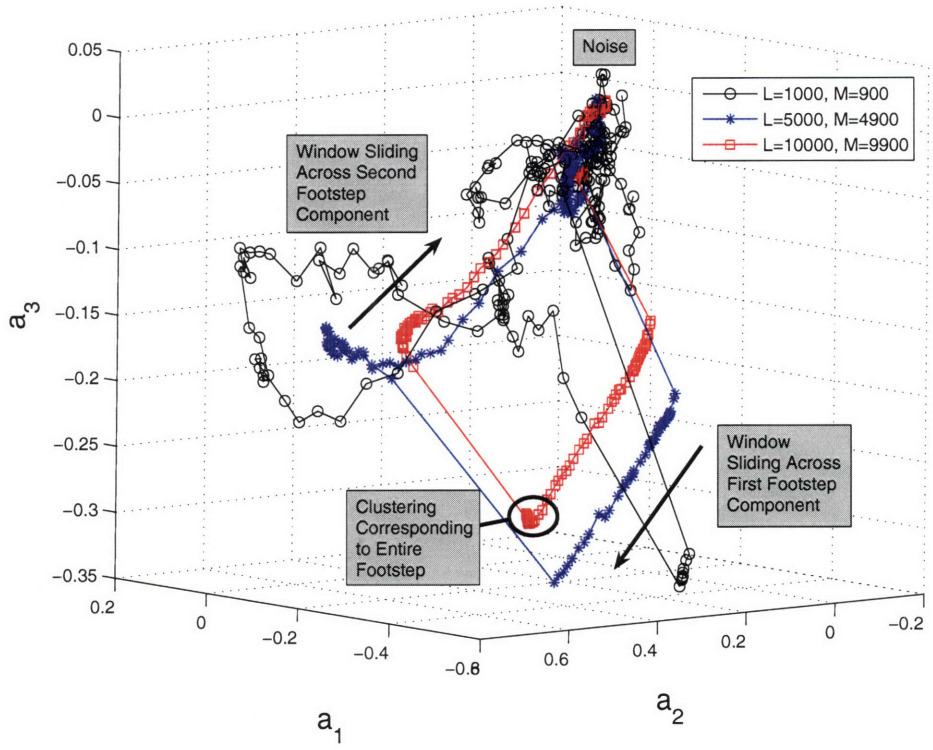


FIGURE 3-11: Multiple window lengths on the same footstep.

Several interesting patterns can be seen in Figure (3-11). First, regardless of window size, each curve traces out a path consisting of three main regions - noise, first footstep component, and second footstep component. In the figure, these three regions are labeled. In addition, the direction the coefficients move as the window slides over the first and second components is indicated with arrows. It can also be seen that when the window is small enough, the coefficients move close to the noise region in between the first and second footstep component regions. On the other extreme, when the window is large enough that it can cover both footstep components at the same time, a new cluster is formed. This new cluster is circled on the $L = 10000$ curve.

The finer resolution in time associated with a shorter window shows up in the figure in other ways. First, a shorter window causes the coefficients to traverse a longer round-trip path as the window slides past a footstep. As can be seen in the figure,

the $L = 1000$ curve's three regions (noise, first and second footstep components) are more spread out in space than the regions associated with the longer windows. As the window length increases, the path the coefficients take in three-dimensional space gets tighter. This can be understood by realizing that the time-dependent activity of the signal can be better resolved with a smaller window. The longer the window, the more samples are used in computing each set of coefficients. Therefore, a longer window results in averaging the signal's behavior over more samples.

The finer time resolution of a shorter window also shows up in the noise cluster. A longer window yields a more tightly clustered noise region. Again, this makes sense when considering that a longer window means computing model coefficients that are averaged over longer sections of data. How tightly clustered the noise coefficients are in space can be measured in the following way. Each data point in the noise cluster corresponds to a vector of prediction coefficients,$\mathbf{a} = \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix}^T$. If there are N sets of coefficients in the noise cluster, $\mathbf{a_1}, \mathbf{a_2}, ..., \mathbf{a_N}$, then the center of the noise cluster can be computed as

$$\bar{\mathbf{a}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{a_i}. \tag{3.9}$$

The square distance from the center of the noise cluster is averaged over all coefficient vectors in the noise cluster. This average is denoted as $\mathcal{V}$ and serves as measure of how tightly clustered a set of coefficients is.

$$\mathcal{V} = \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{a_i} - \bar{\mathbf{a}}\|^2 \tag{3.10}$$

In Figure (3-12), $\mathcal{V}$ is computed for a section of data absent of footsteps. The window length is varied from 1000 to 10000 samples. Regardless of window length, as a window slides across the data, coefficients are generated every 100 samples. N, the number of coefficient vectors generated is constant at 500 for all window sizes. As expected, in Figure (3-12) the chosen metric for parameter clustering, $\mathcal{V}$, decreases as the window length increases.

3.3.3 Consistency of Coefficient Behavior Across Different Footsteps

Also of interest is how similar the coefficient paths are for different footsteps. The coefficients are computed as a window of length $L = 5000$ slides over three different regular footsteps. These coefficients are plotted in Figure (3-13). As can be seen in the figure, the noise sections of all three footstep signals produce coefficients in a single tight cluster. In addition, the coefficient paths for all three footsteps have similar shapes. For all three footsteps, as the window overlaps the first component, the coefficients move to the area of the plot labeled "first component." As the window slides farther along in time and leaves the first component of the footsteps and begins to overlap more of the second component, the coefficients move into the region labeled "second component." The coefficients move back to the noise cluster as the

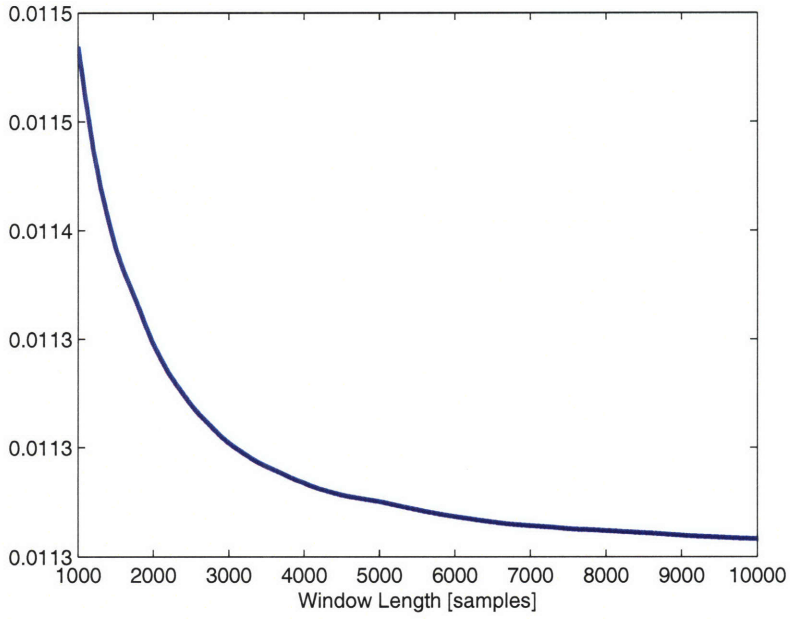


FIGURE 3-12: For various window lengths, $\mathcal{V}$ is computed from a section of data absent of footsteps.

window slides past the second component of the footsteps.

Although the two regions in three-dimensional space containing the coefficients for the first and second components are not as small as the noise cluster, they are a significant distance away from each other and from the noise cluster. With a large number of footstep recordings, a probabilistic model describing the location of the noise and the first and second component clusters could be created. This treatment leads naturally to the classical binary hypothesis testing problem. One hypothesis would be the absence of a footstep while the second hypothesis would be the presence of a footstep. Three regions in three-dimensional coefficient space could be defined:

$$\begin{aligned} R_0 &= Noise \\ R_1 &= First\ Component \\ R_2 &= Second\ Component. \end{aligned}$$

The hypothesis testing decision rule would be based on these regions. A footstep is declared when the movement of coefficients from R_0 to R_1 to R_2 is observed.

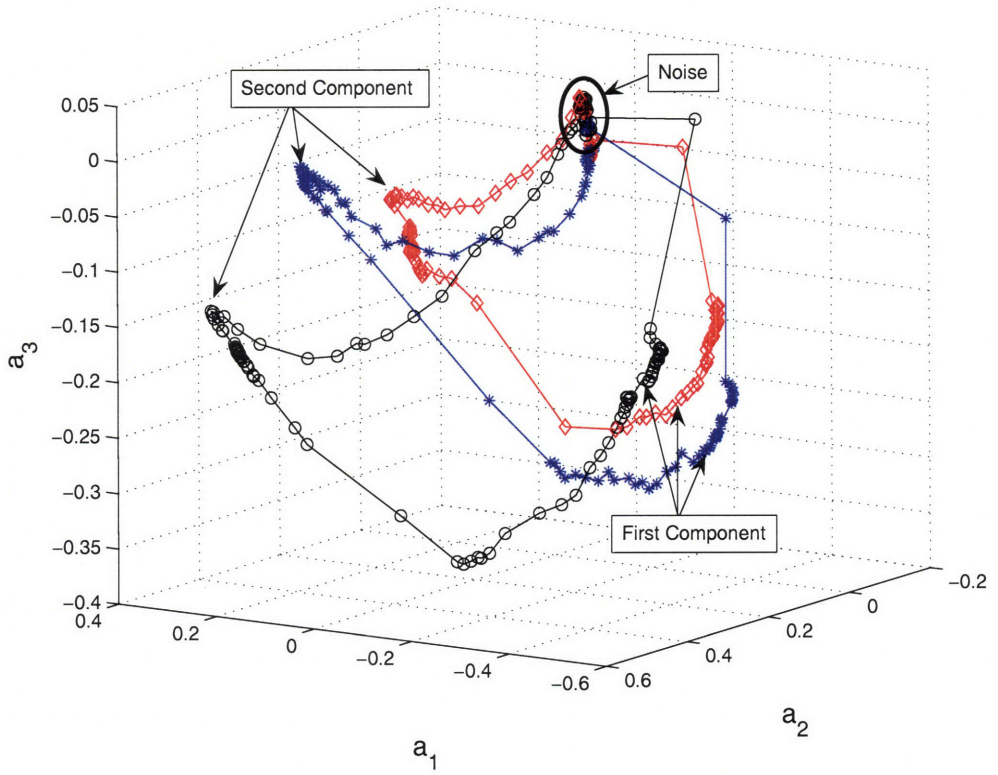


FIGURE 3-13: A comparison of coefficient paths for three different footsteps. Window size of $L = 5000$.

The tradeoff between low probability of false alarm and high probability of detection would be seen in selecting the size of the regions R_0, R_1, and R_2.

3.3.4 Linear Prediction Applied to Stealthy Footsteps

A second set of accelerometer data, the stealthy data set, was recorded in which the walker tried to minimize the amount of footstep signal produced while taking footsteps. As mentioned in Chapter 2, a stealthy footstep tends to have the same basic structure of a regular footstep. That is, the stealthy footstep contains first and second footstep components similar to those seen in the regular footsteps. In both data sets, the first component is narrow in time and evenly distributed over all frequencies while the second component is spread out more in time and fills a frequency band of around 8kHz.

Figure (3-14) shows regular and stealthy footsteps. From this time domain plot, two main differences can be seen in the regular and stealthy footsteps. First, the time between the first and second components is greater for the stealthy footstep. Second for the stealthy footstep, the ratio of second footstep component energy to first footstep component energy is larger.

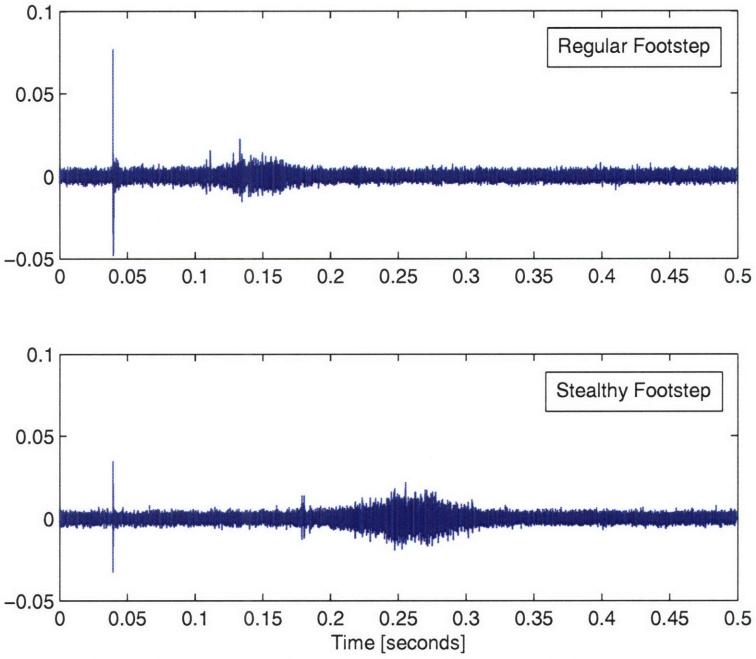


FIGURE 3-14: Accelerometer Footstep Data. Top: Regular Walking Style. Bottom: Stealthy Walking Style.

These differences, particularly the increased relative energy in the second footstep component, can be seen in a plot of the third order linear prediction coefficients. Figure (3-15) plots the coefficient values as a $3,000$ sample long window slides over a regular footstep (black asterisks) and a stealthy footstep (blue diamonds). The coefficient values corresponding to the first and second components and the noise sections of the regular footstep are labeled with arrows. The coefficients from the noise sections of both the regular and stealthy footsteps form a single cluster. However, the coefficient path for the stealthy footstep is of a different shape than the path for the regular footstep. From Figure (3-15) it is difficult to tell whether the stealthy footstep path contains distinct first and second footstep clusters similar to those found in the coefficient path for the regular footstep.

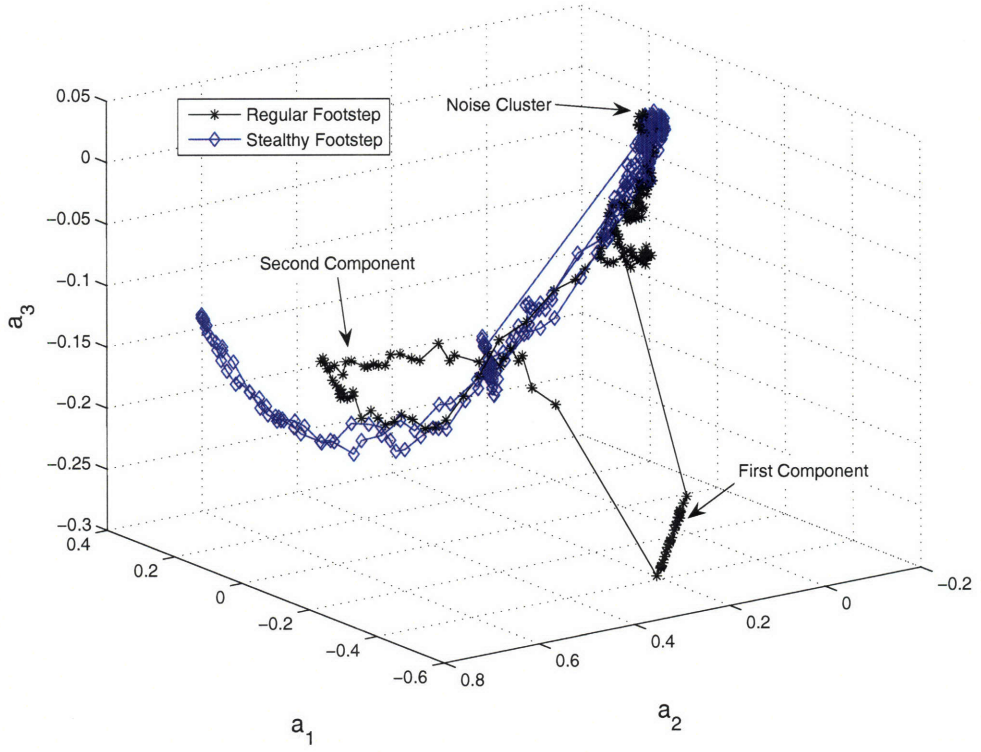


FIGURE 3-15: Comparison of regular and stealthy footstep linear prediction coefficient paths. Regions of the regular footstep are labeled with arrows.

Figure (3-16) shows a time domain plot of the stealthy footstep used to generate Figure (3-15). The first and second components as well as the noise sections are shown in different colors. Figure (3-17) shows the coefficient path that corresponds to sliding a 3000 sample long window across the stealthy footstep. When any portion of the window overlaps the first component, the resulting coefficients are plotted in red in Figure (3-17). Likewise, coefficients computed from sections overlapping the second component are plotted in blue in Figure (3-17). All other coefficients are plotted in black.

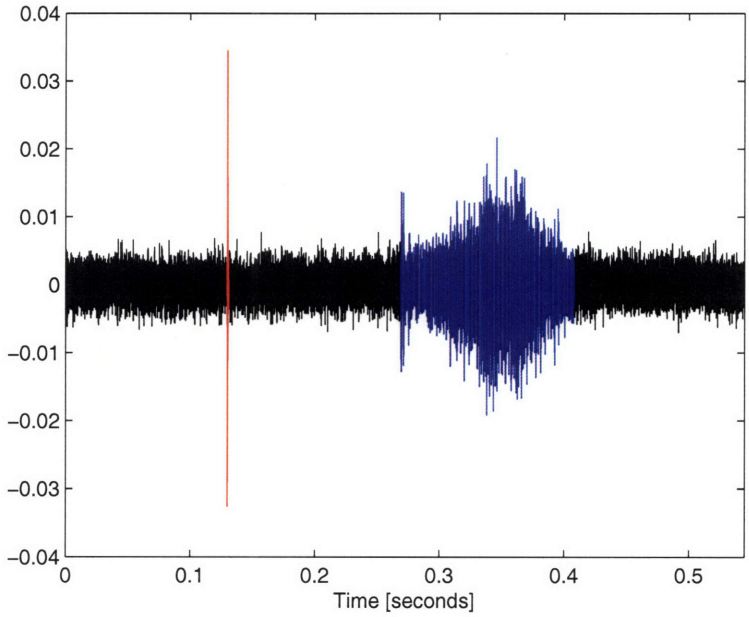


FIGURE 3-16: Stealthy footstep recorded with accelerometer. First component is shown in red, second component in blue.

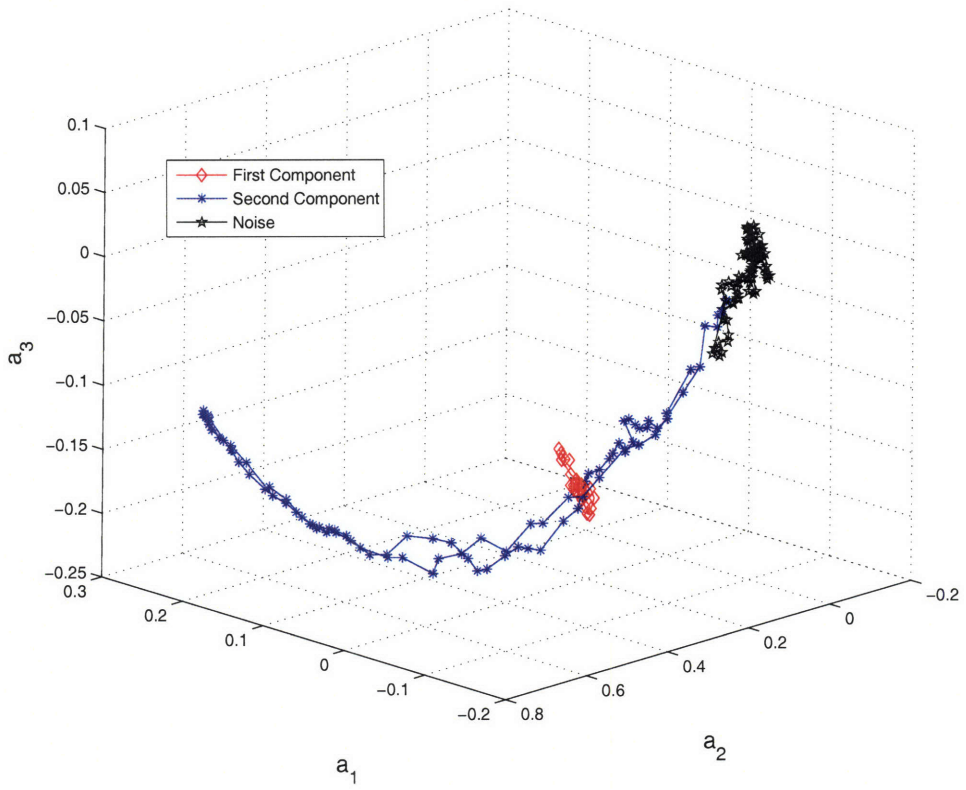


FIGURE 3-17: Coefficient path for stealthy accelerometer footstep.

These plots reveal exactly how the first component affects the parameter path. The first component simply generates a small parameter cluster situated on the path formed by the second component. As the window slides across the second component, the coefficients trace out an arc and end up a considerable distance from the noise cluster before turning around and returning to the noise cluster. The tight cluster of coefficients corresponding to the first component as well as the large arc produced by the second component could be exploited in a binary hypothesis testing framework used for the detection of stealthy footsteps.

3.4 Microphone Footsteps in Three-Dimensional Linear Prediction Coefficient Space

The previous section studies the behavior of linear prediction coefficients computed from accelerometer recordings of footsteps. As mentioned previously, two data sets, regular and stealthy, were recorded with a single sensor unit consisting of both an

accelerometer and a microphone. This section explores the behavior of third order linear prediction coefficients computed from footstep sections of the microphone data. Both regular and stealthy data sets are examined in this section. Data sets are pre-filtered with the noise reducing filter discussed in Section (3.2.1).

3.4.1 Microphone Recordings of Regular Footsteps

Figure (3-18) shows time domain plots of three different regular footsteps recorded with a microphone. The separation of distinct first and second components seen in the accelerometer footsteps is not apparent in the plots of the microphone footsteps. From the time domain plots, it appears the microphone footsteps start with a large spike in amplitude followed by smaller amplitude activity that dies off after about 7000 samples (160ms).

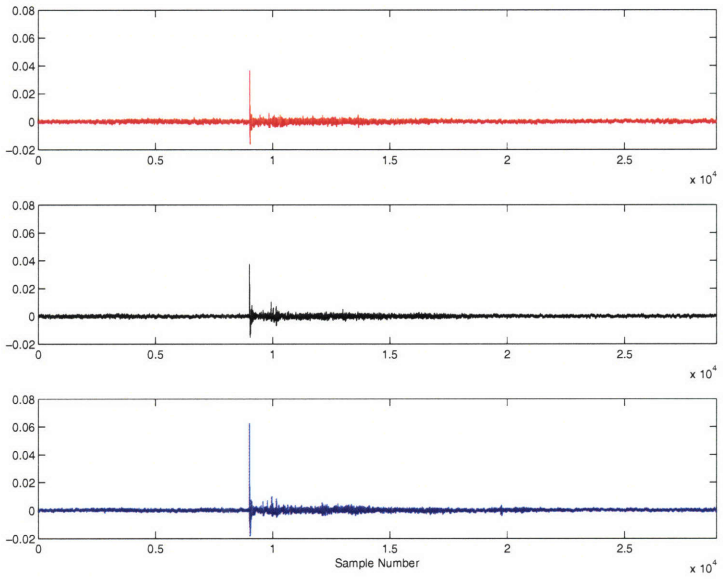


FIGURE 3-18: Three different regular microphone footsteps. The corresponding coefficient paths are shown in Figure (3-19)

A window of length 8000 samples is passed over the three sections of microphone data shown in Figure (3-18). The linear prediction coefficients are generated every 100 samples and the resulting coefficient paths are plotted in Figure (3-19(a)). At the end of this section, the reason for using an 8000 sample window is discussed. Figure (3-19(b)) is a rotated view of the coefficient paths. The arrowheads in Figure (3-19(a)) indicate the direction corresponding to increasing time. As can be seen in the

plots, windows that have not yet reached the main spike in the footsteps produce coefficients that are tightly clustered. As the window slides along in time and overlaps the main spike, the coefficients for all three footsteps jump to the region labeled P_1 in the plots. When the window finally leaves the main spike, the coefficients jump back close to where they started out. The coefficients, however, do not return to the tight cluster where they started out. Instead, the coefficients exhibit significant drifting as the windows slide over the later parts of the data sections in Figure (3-18).

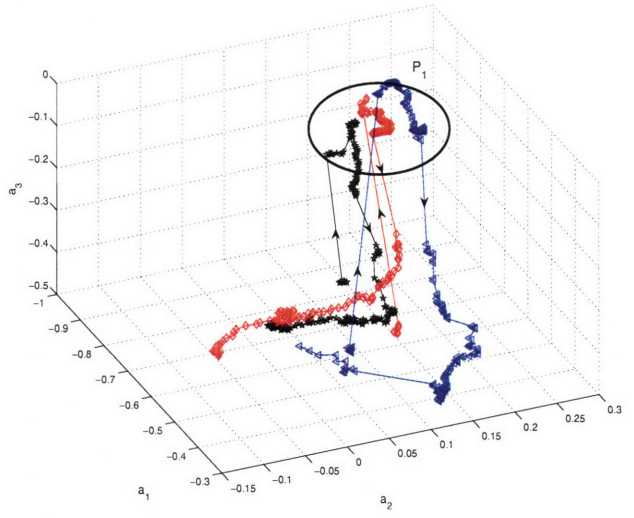
It appears the coefficients may not return to a tight noise cluster in between footsteps as was the case for the accelerometer data. The coefficient path for the section of data shown in Figure (3-20) is used to understand the behavior of the coefficients produced from noise as well as the behavior of coefficients in between footsteps.

An 8000 sample long window is passed over the data in Figure (3-20). The coefficient path (without markers for data points) is shown in Figure (3-21). The first thing to notice about the coefficient path is the noise cluster. As the window slides over the noise and approached the first footstep, the coefficients leave the noise cluster. The coefficients jump to the area labeled "first step main spike" when the window advances far enough along in time to overlap the first footstep's main spike. Once the window leaves the first step's main spike, the coefficients move back in the direction from which they came. Although the coefficients tend toward the noise cluster, they never actually get there before the window begins sliding over the second footstep.

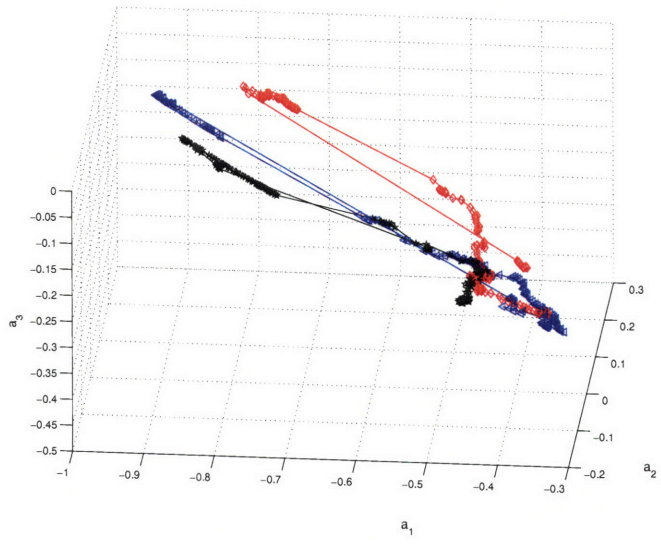
As evidenced by the plot, the coefficients do not return to the noise cluster between footsteps. This behavior can be understood by taking a close look at the footsteps in Figure (3-20). The spikes seen in Figure (3-20) do not mark the beginning of the footstep signals. The footstep signals actually begin before these spikes which is not the case for the accelerometer footstep recordings. The microphone footstep signals are longer in time than the accelerometer footstep signals. As a result, by the time the first footstep in Figure (3-20) has died off, the second footstep signal is starting. As the window slides over the late part of the first step, the coefficients drift toward the noise cluster. The window begins to overlap the beginnings of the second step before the coefficients reach the noise cluster causing them to drift in a different direction.

The speed of sound is generally much higher in solids than in gases. For example, the speed of sound in air is 343 m/s (at 20^o C) while it is roughly 3100 m/s in concrete [4]. Additionally, the walls in a room produce reflections of acoustic signals but are less likely to affect seismic signals. The slower speed of sound in air along with acoustic reflections causes the acoustic impulse response to be longer in time than the seismic impulse response.

The longer duration footstep signals in the microphone data can be explained by the acoustic impulse response being longer than the seismic impulse response. Additionally, the longer impulse response for the acoustic case may be the reason separate first and second footstep components are not as clear in the microphone data. A longer acoustic impulse response means the first footstep component decays more slowly. A sufficiently long acoustic impulse response results in overlap between the first and second footstep components.



(a)



(b)

FIGURE 3-19: Linear prediction coefficient paths for three regular footsteps recorded with a microphone. Plots (a) and (b) are different views of the same data.

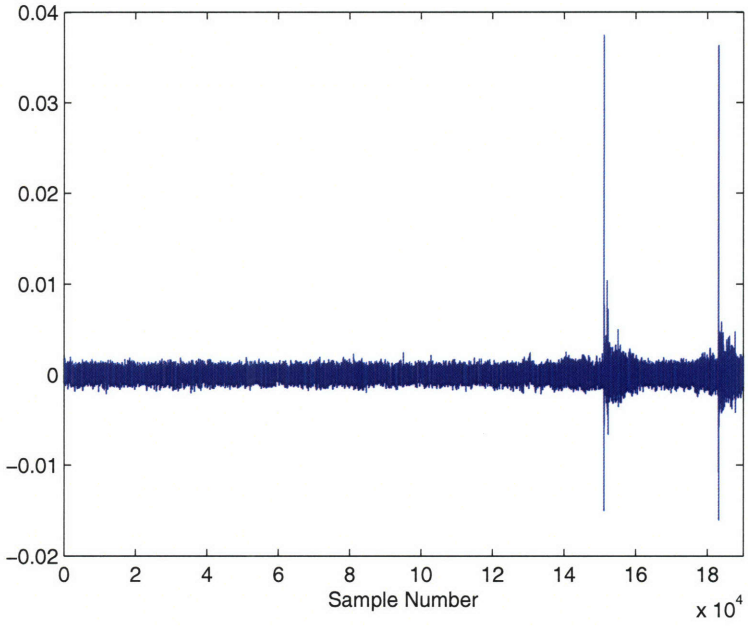


FIGURE 3-20: Section of microphone data used to investigate coefficient behavior between footsteps.

As a window slides over a series of footsteps, the coefficients produced from the samples between the main footstep spikes tend to drift over a large area. In addition, this drifting is not very regular across different steps. Therefore, an algorithm designed to detect footsteps in microphone data may perform best by looking for the large movement of coefficients associated with the main spikes of the footsteps. For example, a decision rule for declaring the presence of footsteps could be based on region P_1 in Figure (3-19(a)). The presence of some n consecutive coefficient vectors in region P_1 may be a suitable basis for declaring the presence of footsteps.

Window Length

In the linear prediction analysis of regular microphone footsteps discussed above, a window length of 8000 samples is used. The reason for choosing 8000 samples is because this length gives tight main spike clustering and consistent coefficient paths across different footsteps. The method used to choose the window length was to plot and examine the coefficient paths for several window lengths.

Figure (3-22) shows the coefficient paths produced from the three footsteps seen in Figure (3-18) using a window length of 3000 samples. As can be seen in Figure

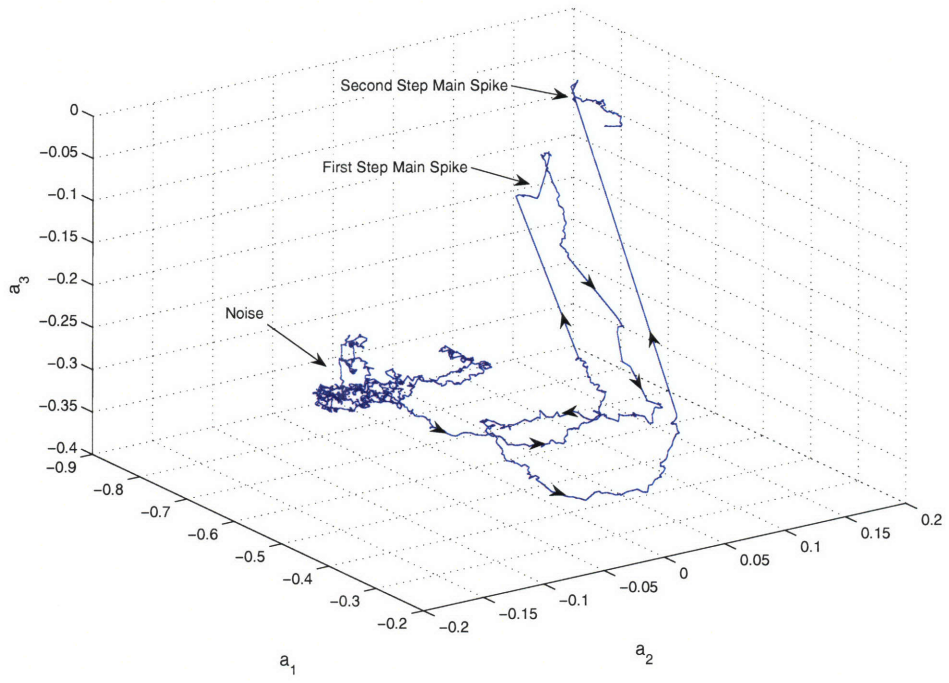


FIGURE 3-21: Coefficient path produced from the data section in Figure (3-20).

(3-22), smaller windows do not produce main spike clustering as tight as that seen in the coefficient paths produced using the 8000 sample window. In addition, smaller windows yield more irregular movement of coefficients computed from the samples between subsequent footsteps' main spikes.

If the window length is increased too far, the window overlaps significant portions of subsequent footsteps. This means long windows do not have the ability to resolve single footsteps. Without the ability to resolve single footsteps, the movement to and from region P_1 in Figure (3-19(a)) will not be seen. A window length of 8000 samples is found to be a good compromise between consistently tight main spike clusters and fine time resolution.

3.4.2 Microphone Recordings of Stealthy Footsteps

This section examines the behavior of the coefficient paths for stealthy footsteps recorded with a microphone. The coefficient paths for approximately 20 stealthy footsteps were examined. Figure (3-23) is representative of the majority of these 20 footsteps. Figure (3-23) plots the coefficient path for one of the data sections in

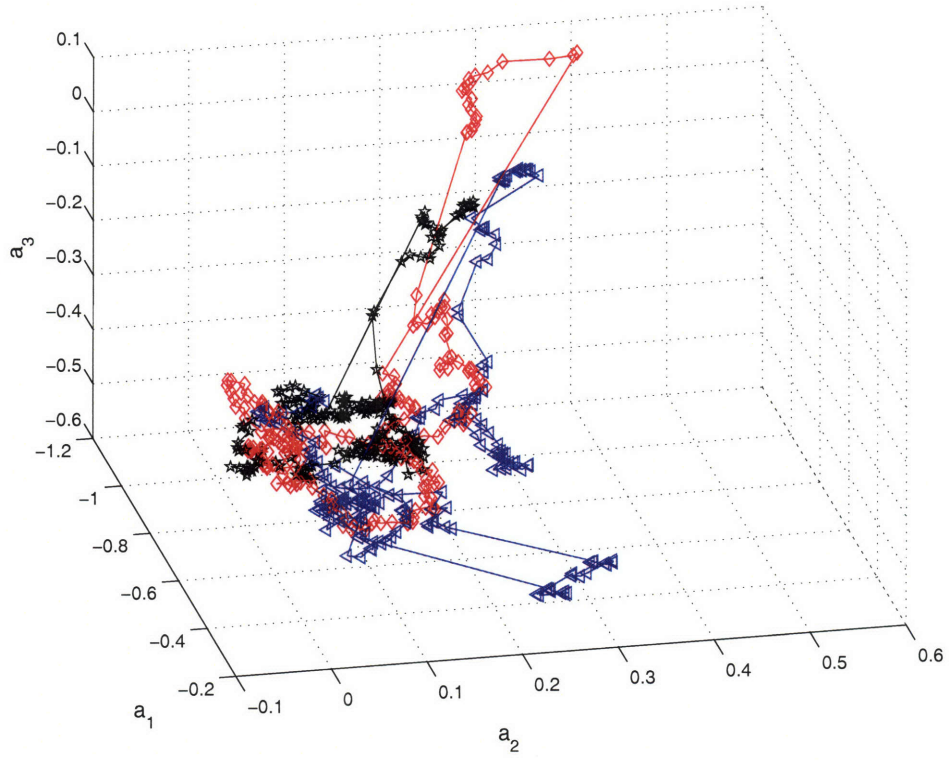


FIGURE 3-22: Coefficient paths generated from data sections in Figure (3-19) using $L = 3000$ sample long window.

Figure (3-18) as well as the coefficient path for a data section containing a stealthy footstep. The stealthy coefficient path, shown in red, does not contain the large jump in coefficient values associated with a footstep's main spike as is seen in the regular footstep coefficient path. The majority of the examined 20 stealthy footstep paths are located in the same region in three dimensional coefficient space as the stealthy path shown in Figure (3-23). However, no clear pattern is apparent among these coefficient paths. Of the 20 stealthy footsteps that were examined, a few were rather large in amplitude. The coefficient paths for these stronger signals closely resembled the coefficient paths for the regular footsteps.

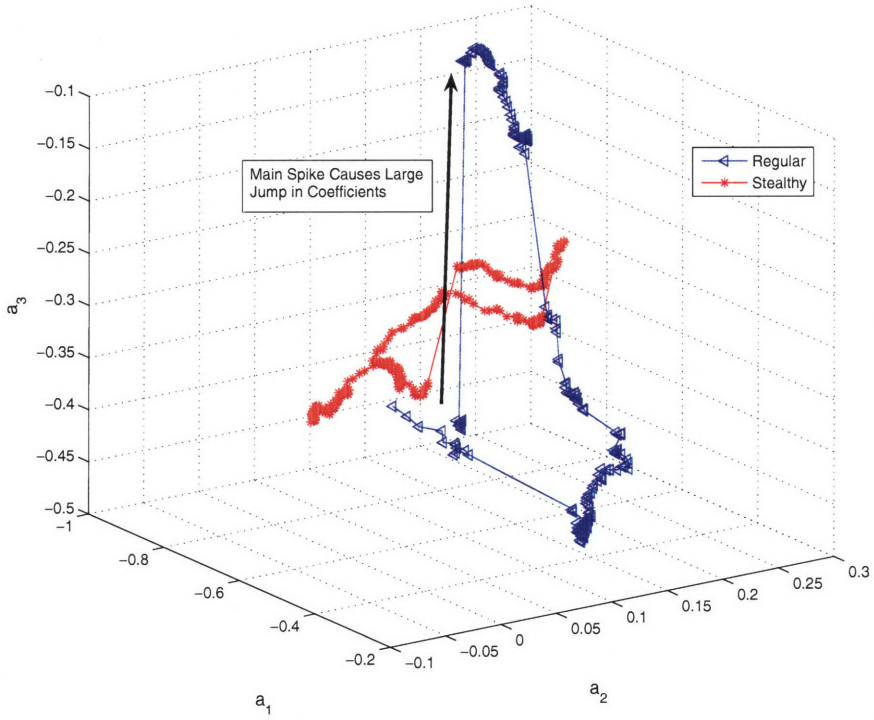


FIGURE 3-23: Comparison of regular and stealthy footstep coefficient paths for microphone data.

The largest amplitude footstep signals in the stealthy data set stand a very good chance of being detected by an algorithm designed to detect regular footsteps. However, since the majority of the stealthy footsteps do not yield consistent coefficient paths, designing an algorithm to detect stealthy footsteps looks to be more challenging.

3.5 Summary

This chapter considers linear predictive modeling of accelerometer and microphone data for footstep detection. First, the chapter explains linear prediction for deterministic signals as well as how linear prediction can be applied to windowed segments of data. The analysis of footstep coefficient paths generated from the regular accelerometer data shows substantial promise for using third order linear prediction coefficients to detect footsteps in accelerometer data.

The stealthy accelerometer data set is used to investigate how stealthy walking would affect the footstep coefficient paths. It is apparent that stealthy footsteps produce

significantly different coefficient paths than the regular footsteps. Nevertheless, there is consistency among these stealthy coefficient paths. From the plots of the stealthy coefficient paths, it seems likely that an algorithm could be designed to successfully identify stealthy footsteps in accelerometer data.

Microphone footstep data is also considered in this chapter. Before applying linear predictive modeling to the microphone data, a method of noise reduction is presented. From the analysis in this chapter, it seems that using linear predictive modeling to detect footsteps is more difficult for microphone data than for accelerometer data. Nevertheless, the analysis in this chapter suggest that a footstep detection scheme for microphone data is possible. However, examination of the coefficient paths produced from stealthy microphone data provides little hope for using linear prediction coefficients to detect these weaker signals.

CHAPTER 4

Blind Channel Identification using Least Squares

4.1 SIMO Channel

A single-input multiple-output (SIMO) channel consisting of q linear time-invariant (LTI) filters is shown in Figure 4-1. The channel outputs, $y_i[n]$, are formed by adding noise to the outputs of the LTI filters. Each LTI filter is characterized by its impulse response, $h_i[n]$.

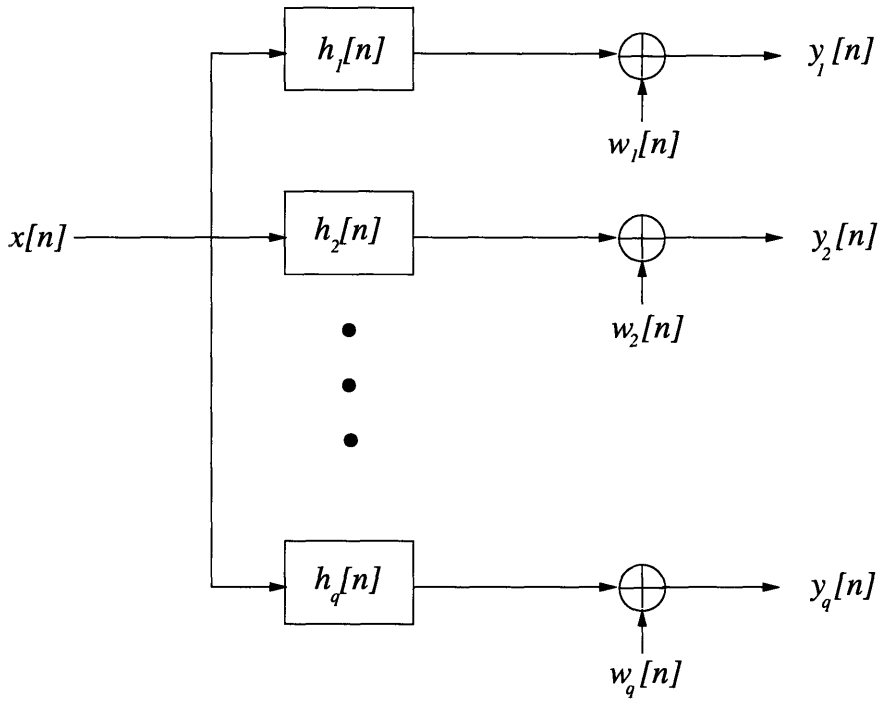


FIGURE 4-1: SIMO Channel

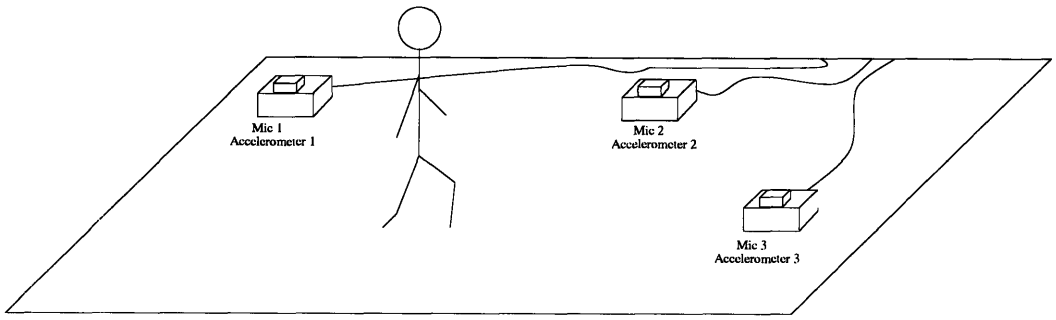


FIGURE 4-2: Six sensors measuring footsteps.

This SIMO channel can be used to model multiple sensors capturing the same signal, $x[n]$. The method by which each sensor captures $x[n]$ is modeled as an LTI filter. Background and sensor noises can be modeled as random noise processes, $w_i[n]$. For example, Figure (4-2) depicts a total of six sensors recording footsteps. A SIMO channel with $q = 6$ LTI filters can be used to model the scenario in this figure. The footstep signal is modeled as $x[n]$, the acoustic and seismic transfer functions from footstep to sensor are modeled as the six LTI filters, and the ambient noise in the room along with sensor electronic noise can be modeled as $w_i[n]$ for each sensor.

Blind channel identification refers to using the outputs, $y_i[n]$ for $1 \leq i \leq q$, to determine the filter impulse responses, $h_i[n]$ for $1 \leq i \leq q$, without knowledge of the input signal, $x[n]$.

4.2 Least Squares Algorithm

In the absence of noise, for any $1 \leq i, j \leq q$,

$$y_i * h_j = (x * h_i) * h_j = (x * h_j) * h_i = y_j * h_i. \tag{4.1}$$

For a finite window of data, Equation (4.1) can be written in matrix form as

$$\mathbf{Y_i[N]h_j} = \mathbf{Y_j[N]h_i} \tag{4.2}$$

where $\mathbf{Y_i[N]}$ and $\mathbf{h_i}$ are defined below.

$$\mathbf{Y_i[N]} = \begin{bmatrix} y_i[K] & y_i[K+1] & \cdots & y_i[2K] \\ y_i[K+1] & y_i[K+2] & \cdots & y_i[2K+1] \\ \vdots & \vdots & \ddots & \vdots \\ y_i[N-K] & y_i[N-K+1] & \cdots & y_i[N] \end{bmatrix} \tag{4.3}$$

$$\mathbf{h_i} = \begin{bmatrix} h_i[K] \\ \vdots \\ h_i[0] \end{bmatrix} \tag{4.4}$$

N is the last sample number in the window of data while K is the maximum order of all of the LTI filters. Equation (4.2) can be written for each possible pairing of filters in the channel. All of these equations can then be written in block matrix form with all of the terms brought to the left side:

$$\mathbf{Yh} = \mathbf{0}. \tag{4.5}$$

For example, with three filters ($q = 3$) Equation (4.5) would be written as

$$\begin{bmatrix} \mathbf{Y_2[N]} & \mathbf{-Y_1[N]} & \mathbf{0} \\ \mathbf{Y_3[N]} & \mathbf{0} & \mathbf{-Y_1[N]} \\ \mathbf{0} & \mathbf{Y_2[N]} & \mathbf{-Y_1[N]} \end{bmatrix} \begin{bmatrix} \mathbf{h_1} \\ \mathbf{h_2} \\ \mathbf{h_3} \end{bmatrix} = 0. \tag{4.6}$$

With noise added to the output of the LTI filters, $\mathbf{Y}$ will be full column rank and there will be no solution to Equation (4.5) except $\mathbf{h} = (0)$. In this noisy case, the one approach to estimating the channel coefficients is to minimize $||\mathbf{Yh}||$ such that $||\mathbf{h}|| = 1$. That is, the estimate for the channel, $\hat{\mathbf{h}}$, is given by the right singular vector associated with the minimum singular value of the matrix $\mathbf{Y}$. This method for estimating the channel has received significant attention. The asymptotic (large $\mathbf{N}$) performance of the channel estimate has been derived for the case where the noise processes, $w_i[n]$, are i.i.d. zero mean Guassian random processes [5].

4.3 Alternative Estimate: Pinning a Channel Coefficient

An alternative approach can be taken to estimate the channel from Equation (4.5). When trying to find the best channel estimate, $\hat{\mathbf{h}}$, in the least squares sense, a constraint must be placed on $\hat{\mathbf{h}}$ to avoid the degenerate solution $\hat{\mathbf{h}} = 0$. The approach discussed in the previous section uses the constraint that $||\mathbf{h}|| = 1$. However, a different constraint could be used to keep $\hat{\mathbf{h}}$ from going to $\mathbf{0}$. The first element of $\hat{\mathbf{h}}$ could be "pinned" to 1. This allows Equation (4.5) to be rewritten in the canonical least squares form, $\mathbf{A\hat{x}} = \mathbf{b}$. Denoting the i^{th} column of $\mathbf{Y}$ as $\mathbf{y_i}$, we have

$$\mathbf{Y}\hat{\mathbf{h}} = \mathbf{0} \tag{4.7}$$

$$\begin{bmatrix} | & | & & | \\ \mathbf{y_1} & \mathbf{y_2} & \cdots & \mathbf{y_{q(K+1)}} \\ | & | & & | \end{bmatrix} \begin{bmatrix} 1 \\ \hat{h}_1[K-1] \\ \vdots \\ \hat{h}_1[0] \\ \hat{\mathbf{h}}_\mathbf{2} \\ \vdots \\ \hat{\mathbf{h}}_\mathbf{q} \end{bmatrix} = \mathbf{0}. \tag{4.8}$$

Equation (4.8) can be rearranged to give

$$\begin{bmatrix} | & & | \\ \mathbf{y_2} & \cdots & \mathbf{y_{q(K+1)}} \\ | & & | \end{bmatrix} \begin{bmatrix} \hat{h}_1[K-1] \\ \vdots \\ \hat{h}_1[0] \\ \hat{\mathbf{h}}_\mathbf{2} \\ \vdots \\ \hat{\mathbf{h}}_\mathbf{q} \end{bmatrix} = \begin{bmatrix} | \\ -\mathbf{y_1} \\ | \end{bmatrix}. \tag{4.9}$$

We can define the reduced matrices in Equation (4.9) as follows: $\mathbf{Y_{red}}$ is the result of removing the first column from the matrix $\mathbf{Y}$ and $\hat{\mathbf{h}}_\mathbf{red}$ is the result of removing the first element from $\hat{\mathbf{h}}$. We can then write Equation(4.9) as $\mathbf{Y_{red}}\hat{\mathbf{h}}_\mathbf{red} = -\mathbf{y_1}$. The least squares solution to Equation (4.9) is $\hat{\mathbf{h}}_\mathbf{red} = -(\mathbf{Y_{red}}^\mathbf{T}\mathbf{Y_{red}})^{-1}\mathbf{Y}_\mathbf{red}^\mathbf{T}\mathbf{y_1}$. To obtain the channel estimate, we append a 1 to the beginning of the vector $\hat{\mathbf{h}}_\mathbf{red}$. Since any solution to the original problem (Equation (4.5)) remains a solution when scaled, we normalize the least square estimate to have unit magnitude. This allows easy comparison to the singular value decomposition (SVD) solution.

In Equation (4.8), the first element of $\hat{\mathbf{h}}$ is pinned to one. However, the choice of element to pin is arbitrary. Any one of the elements of $\hat{\mathbf{h}}$ could be pinned to 1. However, it is conceivable that a problem may occur if the actual value of the element being pinned to 1 is very small.

4.4 Comparison of the SVD and Pinning Channel Estimation Methods

We refer to the two methods to estimate the channel from Equation (4.5) as the SVD method and the pinning method. For some data matrices, $\mathbf{Y}$, these two methods yield the same $\hat{\mathbf{h}}$. However, this is not always the case. To measure the performance of a channel estimate, we use the sum of the squared errors:

$$(\hat{\mathbf{h}} - \frac{\mathbf{h}}{||\mathbf{h}||})^T(\hat{\mathbf{h}} - \frac{\mathbf{h}}{||\mathbf{h}||}) \tag{4.10}$$

.

We compared the performance of these two channel estimation methods by comput-

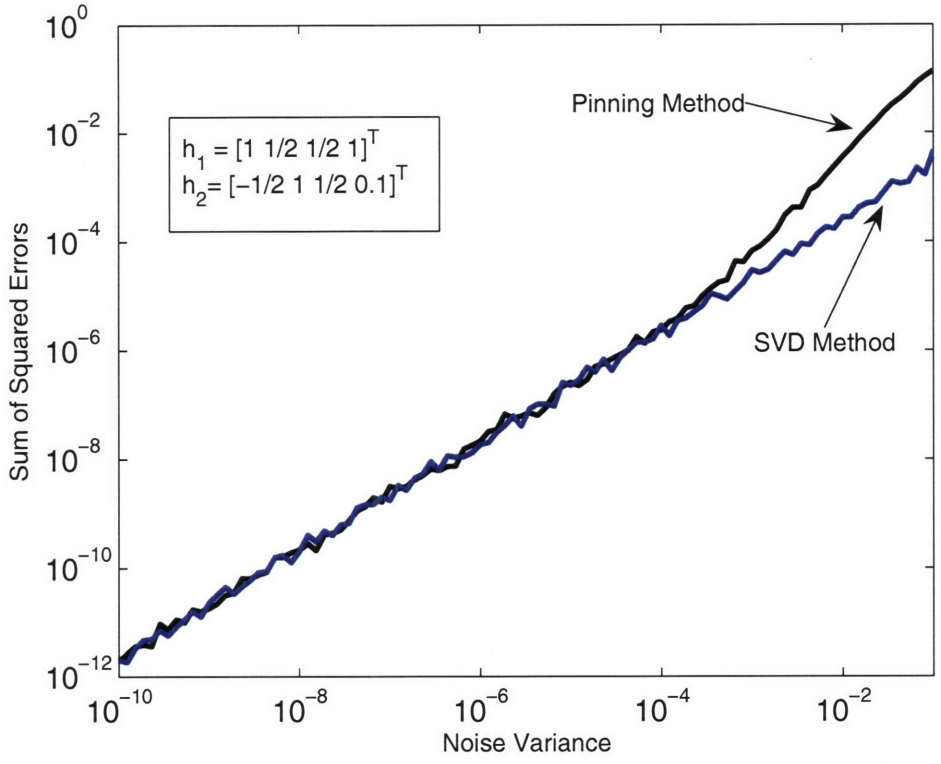


FIGURE 4-3: Performance Comparison of SVD and Pinning Methods

ing Equation (4.10) for a particular two-output channel. The input signal, $x[n]$, was zero mean white Gaussian noise with variance, ${\sigma_x}^2 = 1$. The noise processes, $w_i[n]$, were zero mean i.i.d. Gaussian processes. $\mathbf{N}$, the length of the data window was 1000. For each noise variance, we repeated the experiment 10 times to obtain the sample mean of Equation (4.10). The sample mean of Equation (4.10) plotted versus noise variance as well as the particular filter coefficients used in this experiment can be seen in Figure (4-3). As can be seen in Figure (4-3), the two methods show comparable performance until a noise variance of around 10^{-3}. At this point, the performances begin to diverge and the SVD method is clearly more accurate. It is important to note that $h_1[3]$, the coefficient that is being pinned to 1 actually is 1 in this case.

A second plot is shown in Figure (4-4) in which $h_1[3] = 0.1$. As can be seen in this plot, the SVD method starts outperforming the pinning method at a smaller noise variance, roughly 10^{-6}. Therefore, we can see that how close the pinned element

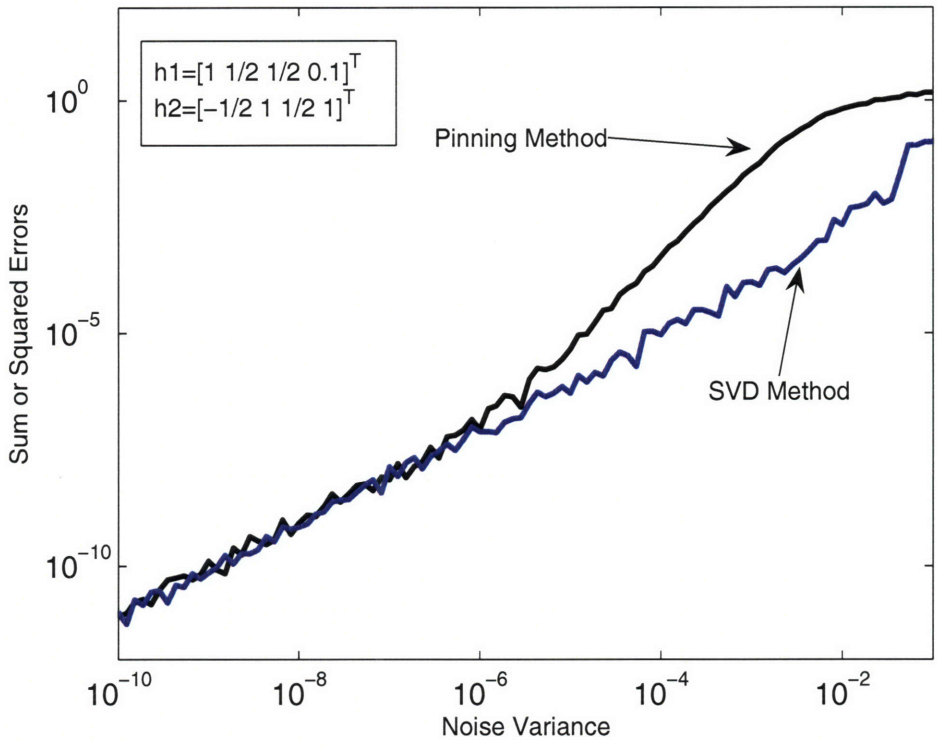


FIGURE 4-4: Performance Comparison of SVD and Pinning Methods with $h_1[K] = 0.1$

actual is to 1, has an effect on the performance of the pinning method.

While the SVD method outperforms the pinning method for large enough noise variance, there are some reasons to consider the pinning method. Since the pinning method allows Equation (4.5) to be cast into the least squares form $\mathbf{A}\hat{\mathbf{x}} = \mathbf{b}$, recursive least squares can be applied to the problem of estimating the channel. With recursive least squares, we can recursively update the channel estimate for each new data point. That is, we can write the channel estimate based on output data up to time k as

$$\hat{\mathbf{h}}(k) = \hat{\mathbf{h}}(k-1) + \mathbf{f}. \tag{4.11}$$

Where $\mathbf{f}$ is a vector function of the output data at time k and the previous channel

estimate, $\hat{\mathbf{h}}(k-1)$.

The pinning method also allows for a more tractable analysis of the probability density function (PDF) of the channel estimate as compared to the PDF analysis associated with the SVD method. However, the derivation of the PDF of the channel estimate is still a challenging problem even for the pinning method. Some of the issues that arise when deriving the PDF can be seen in the example given in the next section.

4.5 Deriving the Distribution of the Channel Estimate

In this example, we consider a two-output channel that consists of single tap filters. We derive the PDF of the channel estimate. The second element of $\mathbf{h}$ is pinned to one and therefore deterministic. Thus, the PDF of the channel estimate consists of the PDF of the estimate of the first filter's single tap, $\hat{h}_1[0]$. For this example, Equation (4.5) reduces to

$$\begin{bmatrix} | & | \\ \mathbf{y_2} & -\mathbf{y_1} \\ | & | \end{bmatrix} \begin{bmatrix} \hat{h}_1[0] \\ 1 \end{bmatrix} = \mathbf{0}. \tag{4.12}$$

Where $\mathbf{y_i} = \begin{bmatrix} y_i[0] & y_i[1] & \cdots & y_i[N] \end{bmatrix}^T$. It follows that the least squares solution is

$$\hat{h}_1[0] = \frac{\mathbf{y_2}^T \mathbf{y_1}}{\mathbf{y_2}^T \mathbf{y_2}}. \tag{4.13}$$

Equation (4.13) may be better understood by rewriting it as

$$\hat{h}_1[0] = \frac{\sum_{n=0}^{N} (h_2[0]x[n] + w_2[n])(h_1[0]x[n] + w_1[n])}{\sum_{n=0}^{N} (h_2[0]x[n] + w_2[n])^2}. \tag{4.14}$$

With Gaussian $x[n]$, $w_1[n]$, and $w_2[n]$, if σ_x^2 is the variance of $x[n]$ and $\sigma_{w_2}^2$ is the variance of $w_2[n]$, then the denominator of Equation (4.14) divided by

$$\sigma_{y_2}^2 = h_2[0]^2 \sigma_x^2 + \sigma_{w_2}^2 \tag{4.15}$$

is a chi-squared random variable with $N+1$ degrees of freedom. A chi-squared random variable with n degrees of freedom is the sum of n independent squared Gaussian random variables each with mean 0 and variance 1.

The numerator, for sufficiently large N, can be approximated as a Gaussian random variable. Therefore, the channel estimate can be approximated as the ratio of a Gaussian random variable and a chi-squared random variable. However, deriving the PDF of this ratio is complicated by the strong correlation between the numerator and denominator due to the presence of $x[n]$ in both.

A new random variable $\tilde{h}_1[0]$ can be defined by the following equation

$$\hat{h}_1[0] = \underbrace{\frac{h_1[0]h_2[0]\sigma_x^2}{h_2^2[0]\sigma_x^2 + \sigma_{w_2}^2}}_{a} + \tilde{h}_1[0]. \tag{4.16}$$

The constant in Equation (4.16) is defined as a. Since $\hat{h}_1[0]$ and $\tilde{h}_1[0]$ are related by a constant offset, the PDF of $\hat{h}_1[0]$ is obtained by shifting the PDF of $\tilde{h}_1[0]$ by a.

An expression for $\tilde{h}_1[0]$ can be obtained by subtracting a from $\hat{h}_1[0]$:

$$\begin{aligned} \tilde{h}_1[0] &= \frac{\mathbf{y}_2{}^T\mathbf{y}_1}{\mathbf{y}_2{}^T\mathbf{y}_2} - a \\ &= \frac{\mathbf{y}_2{}^T(\mathbf{y}_1 - a\mathbf{y}_2)}{\mathbf{y}_2{}^T\mathbf{y}_2}. \end{aligned}$$

Defining $(\mathbf{y}_1 - a\mathbf{y}_2)$ as $\mathbf{y_n}$ yields

$$\tilde{h}_1[0] = \frac{\mathbf{y}_2{}^T\mathbf{y}_n}{\mathbf{y}_2{}^T\mathbf{y}_2}. \tag{4.17}$$

The random vectors $\mathbf{y}_2$ and $\mathbf{y}_n$ are jointly Gaussian. In addition they are uncorrelated:

$$\begin{aligned} \mathrm{E}[\mathbf{y}_2\mathbf{y}_n{}^T] &= \mathrm{E}[\mathbf{y}_2(\mathbf{y}_1 - a\mathbf{y}_2)^T] \\ &= \mathrm{E}[\mathbf{y}_2\mathbf{y}_1{}^T] - a\mathrm{E}[\mathbf{y}_2\mathbf{y}_2{}^T] \\ &= h_1[0]h_2[0]\sigma_x^2\mathbf{I} - \frac{h_1[0]h_2[0]\sigma_x^2}{(h_2[0])^2\sigma_x^2 + \sigma_{w_2}^2}((h_2[0])^2\sigma_x^2 + \sigma_{w_2}^2)\mathbf{I} \\ &= \mathbf{0}. \end{aligned}$$

Thus, $\mathbf{y}_2$ and $\mathbf{y}_n$ are independent.

The mean of $\mathbf{y}_n$ is $\mathbf{0}$ since the mean of $\mathbf{y}_1$ and the mean $\mathbf{y}_2$ are both $\mathbf{0}$. This along with the independence of $\mathbf{y}_2$ and $\mathbf{y}_n$ means the numerator and denominator of Equation (4.17) are uncorrelated:

$$\begin{aligned} \mathrm{E}[\mathbf{y}_2{}^T\mathbf{y}_n] &= \mathrm{E}[\mathbf{y}_2{}^T]\mathrm{E}[\mathbf{y}_n] \\ &= 0 \\ \mathrm{E}[(\mathbf{y}_2{}^T\mathbf{y}_2)(\mathbf{y}_2{}^T\mathbf{y}_n)] &= \mathrm{E}[\mathbf{y}_2{}^T\mathbf{y}_2\mathbf{y}_2{}^T]\mathrm{E}[\mathbf{y}_n] \\ &= 0 \\ &= \mathrm{E}[\mathbf{y}_2{}^T\mathbf{y}_2]\mathrm{E}[\mathbf{y}_2{}^T\mathbf{y}_n]. \end{aligned}$$

Although the numerator and denominator of Equation (4.17) are uncorrelated, they are not jointly Gaussian and they are not independent. However, in deriving the PDF of $\tilde{h}_1[0]$, independence is assumed. At the end of the derivation, histograms are used

to show that this approximation is reasonable.

The numerator in Equation (4.17), ${\mathbf{y}_2}^T\mathbf{y}_n$, is the sum of $N+1$ independent random variables. In deriving the PDF of $\tilde{h}_1[0]$, ${\mathbf{y}_2}^T\mathbf{y}_n$, is approximated as a Gaussian random variable. The random variable $\mathbf{y}_2^T\mathbf{y}_n$ is zero mean. Using the independence of $\mathbf{y}_2$ and $\mathbf{y}_n$ and the properties of the trace, the variance can be calculated as

$$\begin{aligned} var(\mathbf{y}_2^T\mathbf{y}_n) &= \mathrm{E}[\mathbf{y}_2^T\mathbf{y}_n\mathbf{y}_n^T\mathbf{y}_2] \\ &= \mathrm{E}[trace(\mathbf{y}_2^T\mathbf{y}_n\mathbf{y}_n^T\mathbf{y}_2)] \\ &= \mathrm{E}[trace(\mathbf{y}_n\mathbf{y}_n^T\mathbf{y}_2\mathbf{y}_2^T)] \\ &= trace(\mathrm{E}[\mathbf{y}_n\mathbf{y}_n^T]\mathrm{E}[\mathbf{y}_2\mathbf{y}_2^T]) \\ &= \{(h_1[0]^2\sigma_x^2+\sigma_{w_1}^2)(h_2[0]^2\sigma_x^2+\sigma_{w_2}^2)-(h_1[0]h_2[0]\sigma_x^2)^2\}(N+1) \\ &= \sigma_{num}^2 \end{aligned}$$

For sufficiently large N, $\mathbf{y}_2^T\mathbf{y}_n/\sigma_{num}$ is a zero mean, unit variance Gaussian random variable.

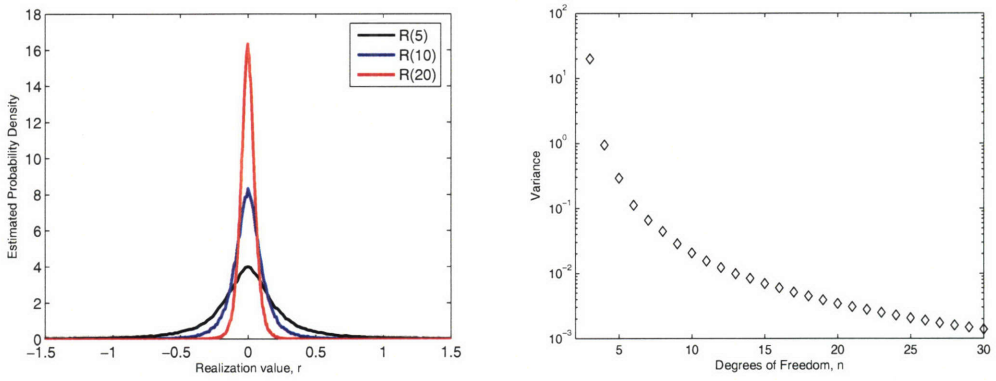
As given in Equation (4.15), the denominator of Equation (4.17), ${\mathbf{y}_2}^T\mathbf{y}_2$, when divided by $h_2[0]^2\sigma_x^2+\sigma_{w_2}^2$ is a chi-squared random variable with $N+1$ degrees of freedom. If $h_2[0]^2\sigma_x^2+\sigma_{w_2}^2$ is defined as $\sigma_{y_2}^2$, then the random variable

$$\frac{\sigma_{y_2}^2}{\sigma_{num}}\tilde{h}_1[0] \tag{4.18}$$

for large N has the distribution of a zero mean, unit variance random variable divided by a chi-squared random variable with $N+1$ degrees of freedom. The Gaussian and chi-squared random variables in this ratio are uncorrelated but may not be independent. However, in this derivation, the distribution of the random variable in Equation (4.18) is approximated as the ratio of two independent random variables. Since a name for this distribution could not be found in the literature, a random variable that is the ratio of a zero mean, unit variance random variable and an independent chi-squared random variable with n degrees of freedom from here on will be referred to as an $R(n)$ random variable with n degrees of freedom and an associated $R(n)$ distribution.

Figure (4-5(a)) shows estimates of the PDF for $R(5)$, $R(10)$, and $R(20)$. These estimates were formed by sampling random number generators and computing histograms. Figure (4-5(b)) plots the variance of the distribution of $R(n)$ for n from 3 to 30.

Simulations are performed in which histograms are generated in order to examine how well the distribution of $\tilde{h}_1[0]$ matches the distribution of an R random variable scaled by $\sigma_{num}/\sigma_{y_2}^2$. The following parameter values are used in the simulations: $\hat{h}_1[0]=0.8, \hat{h}_2[0]=0.5, \sigma_x^2=1, \sigma_{w_1}^2=\sigma_{w_2}^2=0.02$. The histograms are computed by sampling from random number generators 50,000 times. Figure (4-6) shows the results of the simulations. The black lines represent PDFs of the scaled R distribution



(a) (b)

FIGURE 4-5: The R distribution. (a) PDF estimates of $R(5)$, $R(10)$, and $R(20)$. (b) Variance of the distribution of $R(n)$ for n from 3 to 30.

estimated from histograms. The blue bar graphs are histograms of $\tilde{h}_1[0]$. Four plots are shown for $N = 3, 10, 30, 100$. As can be seen from the plots, the match between the distribution for $\tilde{h}_1[0]$ and the scaled R distribution improves as N increases. Additionally, Figure (4-6(a)) shows that the distributions match well even at low values of N.

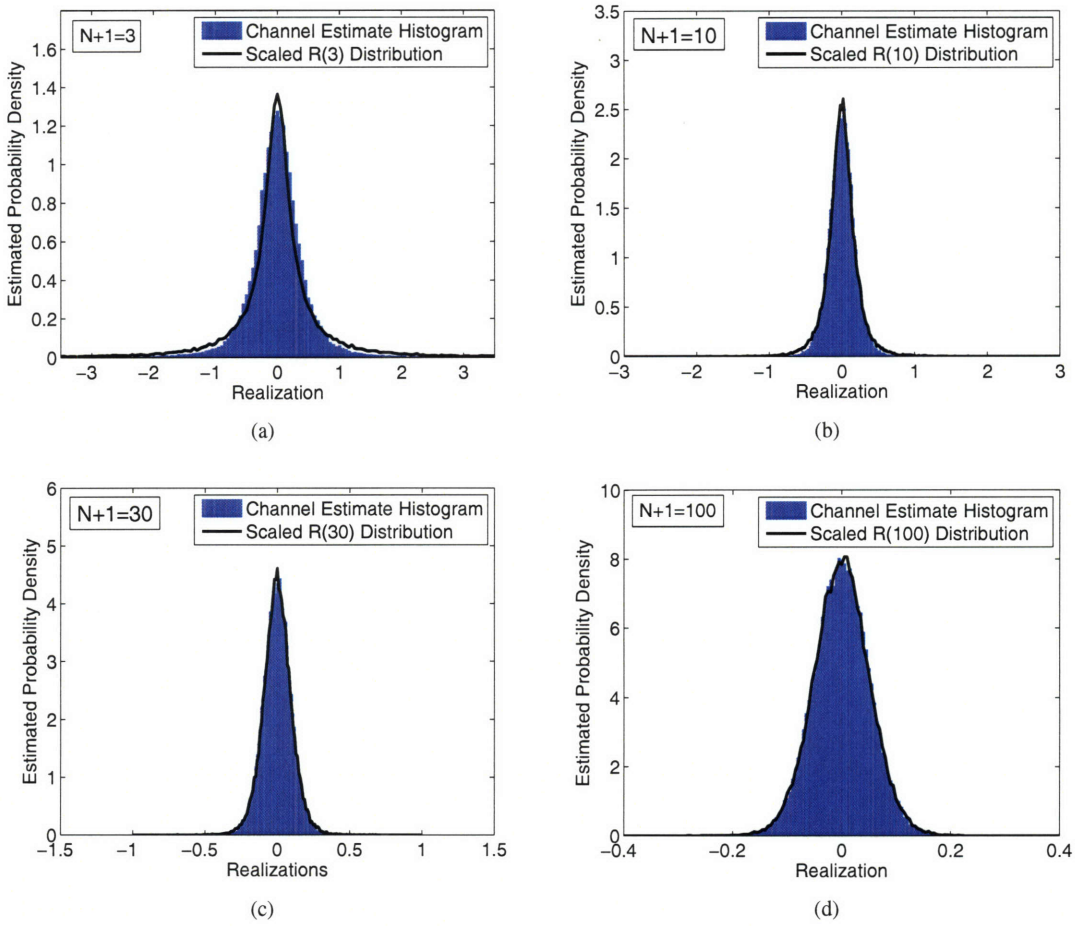


FIGURE 4-6: Histogram plots comparing the estimated probability density for $\tilde{h}_1[0]$ and an R random variable scaled by $\frac{\sigma_{num}}{\sigma^2_{y_2}}$. (a) $N+1=3$ (b) $N+1=10$ (c) $N+1=30$ (d) $N+1=100$.

There are two factors that can potentially keep $\tilde{h}_1[0]$ from being exactly an $R(n)$ random variable scaled by $\sigma_{num}/\sigma^2_{y_2}$. First, the $R(n)$ random variable is the ratio of independent random variables where as $\tilde{h}_1[0]$ is the ratio of uncorrelated random variables. Second, the numerator in Equation (4.17), $\mathbf{y}_2^T\mathbf{y}_n$, scaled by $1/\sigma_{num}$ approaches the distribution of a zero mean, unit variance, Gaussian random variable for large N. However, for smaller values of N this approximation may break down.

Figure (4-7) compares the histograms of $\mathbf{y}_2^T\mathbf{y}_n/\sigma_{num}$ to the PDF of a zero mean, unit variance Gaussian random variable for the same values of N used in Figure (4-

6). As can be seen in Figure (4-7), the distribution for $\mathbf{y}_2^T\mathbf{y}_n/\sigma_{num}$ for small N has more probability mass at small absolute values, less mass at intermediate values and thicker tails than the Gaussian. On the other hand, Figure (4-6) shows the opposite effect in the approximation of the density of $\hat{h}_1^{(N)}[0]$ with an $R(N)$ random variable for small N. It is concluded that the lack of independence of the numerator and the denominator of Equation (4.17) is the dominant effect in the discrepancy in the approximation for small N. All effects are alleviated and the approximation becomes quite good for even modest values of N, say $N = 30$.

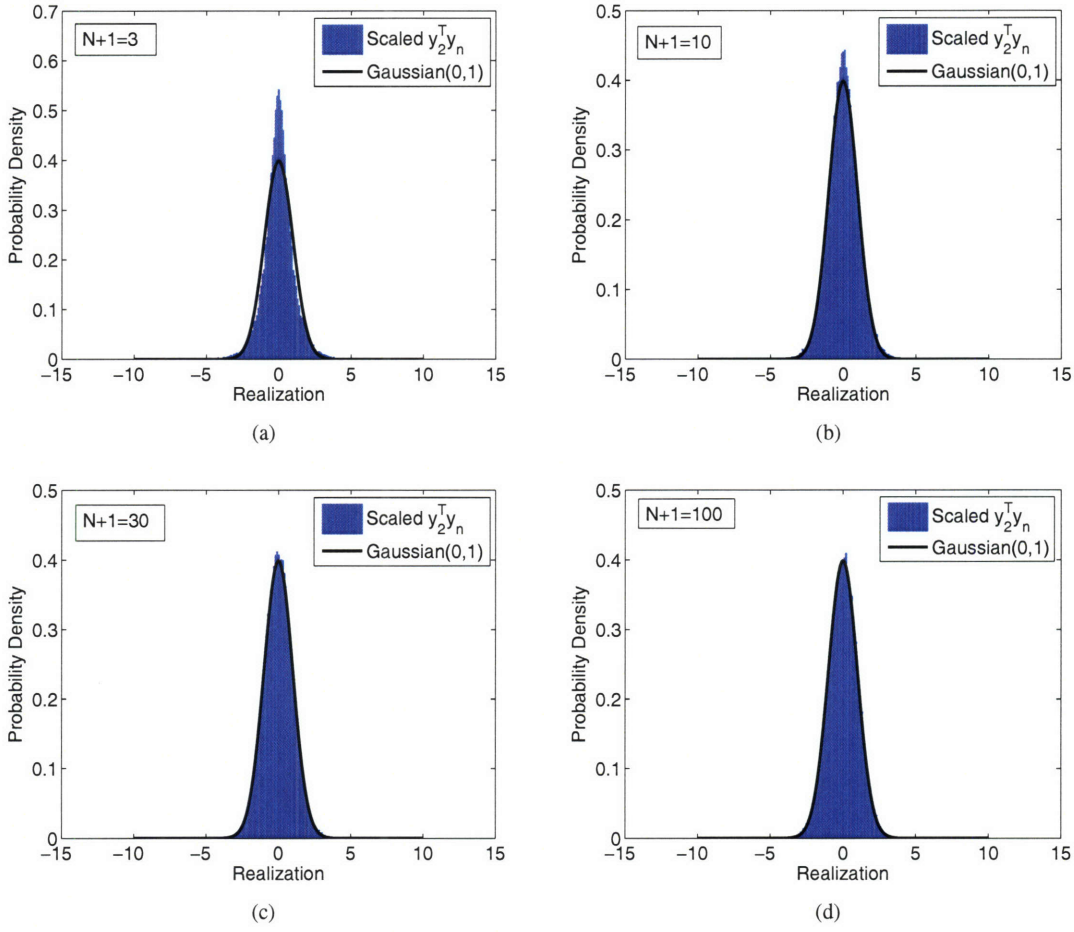


FIGURE 4-7: Plots comparing the histograms of $\frac{\mathbf{y}_2^T\mathbf{y}_n}{\sigma_{num}}$ to the PDF of a zero mean, unit variance Gaussian random variable. Each plot corresponds to a different value for N. (a) $N+1=3$ (b) $N+1=10$ (c) $N+1=30$ (d) $N+1=100$.

An approximate distribution for $\hat{h}_1[0]$ can be obtained from the approximate distribution for $\tilde{h}_1[0]$. The random variable $\hat{h}_1[0]$ can be written in terms of N as

$$\hat{h}_1^{(N)}[0] \approx a + \frac{\sigma_{num}}{\sigma_{y_2}^2} R(N+1) \tag{4.19}$$

where $R(N)$ is a random variable with an R distribution with N degrees of freedom. R has $N+1$ degrees of freedom in Equation (4.19) since the random variable $\hat{h}_1^{(N)}[0]$ involves data of length $N+1$, i.e. $y_i[0], ...y_i[N]$ for $i = 1, 2$. The random variable $\hat{h}_1^{(N)}[0]$ has mean

$$\begin{aligned} mean(\hat{h}_1^{(N)}[0]) &= a \\ &= \frac{h_1[0]h_2[0]\sigma_x^2}{h_2^2[0]\sigma_x^2 + \sigma_{w_2}^2} \end{aligned}$$

and approximate variance

$$\begin{aligned} var(\hat{h}_1^{(N)}[0]) &= \left(\frac{\sigma_{num}}{\sigma_{y_2}^2}\right)^2 var(R(N+1)) \\ &= \left[\frac{h_1[0]^2\sigma_x^2 + \sigma_{w_1}^2}{h_2[0]^2\sigma_x^2 + \sigma_{w_2}^2} - \frac{(h_1[0]h_2[0]\sigma_x^2)^2}{(h_2[0]^2\sigma_x^2 + \sigma_{w_2}^2)^2}\right](N+1)var(R(N+1)). \end{aligned}$$

For the parameters in the simulation,

$$\begin{aligned} mean(\hat{h}_1^{(N)}[0]) &\approx 1.48 \\ var(\hat{h}_1^{(N)}[0]) &\approx 1.85(N+1)R(N+1). \end{aligned} \tag{4.20}$$

4.6 Summary

In the first section of this chapter, the SIMO channel is presented. Additionally, the use of the SIMO channel to model multiple sensors measuring the same signal is explained with the example in Figure (4-2). Section 2 explains the least squares algorithm to estimate the channel impulse responses blindly, i.e. with knowledge of the output signals but no knowledge of the input signal. This algorithm, termed the SVD algorithm, yields the right singular vector of a matrix as the channel estimate which makes analyzing the performance of the channel estimate difficult. In the next section, an alternative method to blindly estimate the channel, the pinning method, is presented. This pinning method does not perform as well as the SVD algorithm at low signal to noise ratio (SNR). However, the pinning method has a couple of advantages over the SVD method. First, the pinning method allows for a recursive estimation of the channel. Second, the pinning method yields a more tractable performance analysis of the channel estimate. The last section of this chapter carries

out the analysis of the performance of the pinning method for the case of a SIMO channel with two, single tap filters.

CHAPTER 5 *Conclusion*

This thesis considers the general problem of detecting footsteps with acoustic and seismic sensors. Each of the three main chapters looks at a different aspect of this problem. Chapter 2 explains two signal processing systems to help a human operator detect footsteps when listening to accelerometer data. Chapter 3 establishes the feasibility of using linear prediction coefficients for automatic footstep detection in accelerometer and microphone data. Lastly, Chapter 4 explains a modification to the least squares method for blind channel identification.

5.1 Digital Processing to Aid the Listener in Footstep Detection

The second chapter of this thesis considers two signal processing systems to help a human operator recognize footsteps when listening to accelerometer data. The inspiration for designing these two systems comes from comparing spectrograms of the accelerometer data to equal loudness curves for the human ear. The equal loudness curves show the relative sensitivity of the human ear for sounds of various frequencies. The spectrograms show that a significant portion of the total footstep signal energy is located at frequencies to which the ear is not very sensitive. The two systems presented in Chapter 2 essentially move the footstep signal energy from low-sensitivity frequency ranges to high-sensitivity frequency ranges. Listening to data files before and after the footstep enhancing processing proves that the processing systems make the footsteps significantly easier to recognize. However, the performance of these systems is difficult to quantify. These before and after data files are made available to the reader through [1].

5.2 Linear Predictive Modeling to Identify Footsteps

In Chapter 3, linear predictive modeling is shown to be a feasible means for detecting footsteps in accelerometer and microphone data. Plotting the third order linear prediction coefficients for windowed sections of data shows the time evolution of the coefficients in three-dimensional coefficient space. As the window slides over footstep signals, the coefficients move between separable regions in space. The presence of this distinctive signature demonstrates the feasibility of using well-known hypothesis testing procedures to produce a footstep detection system with quantifiable detection and false alarm probabilities. Much more data would be needed to further develop such a system.

5.3 Blind Channel Identification Using Least Squares

Chapter 4 explains how multiple sensors capturing the same signal can be modeled as a single-input, multi-output (SIMO) channel. The SVD method for blindly estimating the channels is explained. The difficult performance analysis associated with the SVD method motivates the pinning method developed here. The SVD method outperforms the pinning method at low signal to noise ratio. However, the pinning method has a performance analysis that is more tractable than the analysis for the SVD method. This analysis is demonstrated by deriving the distribution for the channel estimate is derived for a channel consisting of two, single-tap filters.

APPENDIX A

Digital Processing Implementations in MATLAB

A.1 Method 1 Processing

```
function output=method1(input)
% This MATLAB function is an implementation
% of the first method of processing designed
% to aid the listener in detecting footsteps
% (Chapter2).  The 0-11.025kHz range of the
% output signal contains the superposition of
% the input signal's 0-11.025kHz and 11.025
% -22.05kHz bands.  The output does not contain
% any energy beynod 11.025kHz.  Input is a
% column vector.

% The FIR1 function was used to design FIR
% filters via the windowing method.

% Highpass filter the input to obtain
% 11.025-22.05kHz band.
high_half=filter(fir1(300,0.5,'high'),1,input);

% Create vector of cosine values.
```

```
% n=1:length(input);
cos_vec=2*cos(0.5*pi*n);

% Shift by multiplying by cosine vector.
high_half_shifted=cos_vec'.*high_half;

% Sum contains the desired output in the
% 0-11.025kHz range but undesired content
% at higher frequencies.
sum=high_half_shifted+input;

% Lowpass filter to remove undesired content.
output=filter(fir1(300,0.5),1,sum);
```

A.2 Method 2 Processing

```
% function output=method2(input)
% This MATLAB function is an implementation of
% the second method of processing designed to
% aid the listener in detecting footsteps
% (Chapter2).  The 1-6kHz range of the output
% signal contains the superposition of the
% input signal's 6-11kHz and 11-16kHz bands.
% The output does not contain any energy outside
% of the 1-6kHz bands.  Input is a column vector.
%
% The FIR1 function was used to design FIR
% filters via the windowing method.

% 6-11kHz band of input.
c=fir1(300,[6/22.05 11/22.05]);
six_to_11=filter(c,1,input);

% 11-16kHz band of input.
d=fir1(300,[11/22.05 16/22.05]);
elev_to_16=filter(d,1,input);

% Create a vector of cosine values.
n=1:length(input); cos_vect=cos(pi*(5/22.05)*n);

% Shift the 6-11kHz band down to 1-6kHz.
a=fir1(50,9/22.05);
six_to_11_sh=filter(a,1,cos_vect'.*six_to_11);

% Shift the 11-16kHz band down to 1-6kHz.
```

```
cos_vect=cos(pi*(10/22.05)*n);
b=fir1(50,12/22.05);
elev_to_16_sh=filter(b,1,cos_vect'.*elev_to_16);

% The output is the sum.
output = elev_to_16_sh + six_to_11_sh;
```

APPENDIX B *Available Data Files*

Several data files discussed in Chapter 2 can be found at
[1]. The following is a description of each of the files.

Original Regular Accelerometer.wav
The pre-processed recording of regular walking.

Original Stealthy Accelerometer.wav
The pre-processed recording of stealthy walking.

Method 1 Regular Accelerometer.wav
The recording of regular walking processed with the first method discussed in Chapter 2.

Method 1 Stealthy Accelerometer.wav
The recording of stealthy walking processed with the first method discussed in Chapter 2.

Method 2 Regular Accelerometer.wav
The recording of regular walking processed with the second method discussed in Chapter 2.

Method 2 Stealthy Accelereometer.wav
The recording of stealthy walking processed with the second method discussed in

Chapter 2.

Bibliography

[1] Accelerometer and microphone data files discussed in Chapter 2 and listed in Appendix B may be available by contacting crohrs@mit.edu.

[2] Simon Haykin. *Adaptive Filter Theory*. Information and Systems Sciences Series. Prentice-Hall, Englewood Cliffs, New Jersey, 1986.

[3] Kenneth M. Houston and Daniel P. McGaffigan. Spectrum analysis techniques for personnel detection using seismic sensors. *Proceedings of the SPIE*, 5090:162–173, 2003.

[4] L.E. Kinsler and A.R. Frey. *Fundamentals of Acoustics*. John Wiley and Sons, second edition, 1962.

[5] Joonsung Lee. Acoustic signal estimation using multiple blind observations. Master of science, Massachusetts Institute of Technology, 77 Massachusetts Avenue Cambridge, MA 02139, January 2006.

[6] Alan V. Oppenheim, Ronald W. Schafer, and John R. Buck. Parametric signal modeling for discrete-time signal processing, third edition. 2004.

[7] Yasuhiro Shoji, Akitoshi Itai, and Hiroshi Yasukawa. Personal identification using footstep detection. *Proceedings of 2004 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, pages 43–47, Nov 2004.

[8] George Succi, Daniel Clapp, Robert Gampert, and Gervasio Prado. Footstep detection and tracking. *Proceedings of SPIE*, 4393:22–26, 2001.

[9] Eric W. Weisstein. Kurtosis. From Mathworld - A Wolfram Web Resource. http://mathworld.wolfram.com/Kurtosis.html.