Estimating glottal voicing source characteristics by measuring and modeling the acceleration of the skin on the neck

by

Harold Arthur Cheyne II

B.S. Electrical Engineering
Tufts University, 1993

Submitted to the Harvard-MIT Division of Health Sciences and Technology Speech and Hearing Biosciences and Technology Program in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy in Speech and Hearing Biosciences and Technology

at the
Massachusetts Institute of Technology
February 2002

©2002 Massachusetts Institute of Technology. All rights reserved.

Signature of Author____

Harvard-MIT Division of Health Sciences and Technology
Speech and Hearing Sciences Program
November 1, 2001

Certified by____

Robert E. Hillman, Ph.D., CCC-SLP
Associate Professor and Director Voice Disorders Center
Department of Otology and Laryngology, Harvard Medical School
Thesis co-supervisor

____

Kenneth N. Stevens, Sc.D.
Clarence J. Lebel Professor of Electrical Engineering
Thesis co-supervisor

Accepted by____

Martha L. Gray, Ph.D.
Edward Hood Taplin Professor of Medical and Electrical Engineering, HST
Professor of Electrical Engineering and Computer Science, MIT
Co-Director, Division of Health Sciences and Technology

Estimating glottal voicing source characteristics by measuring and modeling the acceleration of the skin on the neck

by
Harold Arthur Cheyne II

Submitted to the Harvard-MIT Division of Health Sciences and Technology Speech and Hearing Biosciences and Technology Program on November 14, 2001 in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Speech and Hearing Biosciences and Technology

ABSTRACT

In the clinical management of voice patients, quantifying vocal function is becoming increasingly important both for corroborating clinicians' subjective impressions during a voice evaluation and for assessing the effectiveness of surgery or voice therapy. Current devices for quantifying vocal function measure acoustic, aerodynamic and electrical parameters during short tasks such as reading. One technique that has shown potential for measuring vocal function but has been mostly used to quantify speech-related behaviors besides phonation is measuring the acceleration of the skin near the larynx.

The acceleration of the skin on the neck between the cricoid cartilage and the sternal notch arises from the airflow pulses that result from vocal-fold vibration. At least two sets of structures play a role in the transformation of the acoustic source resulting from vocal-fold vibration into the measured acceleration: the subglottal system, and the tissues between the subglottal airspace and the accelerometer (e.g., tracheal cartilage, skin, etc.). Advantages of measuring acceleration over current techniques include 1) the structures that filter the glottal pulses vary less over time than the vocal tract and thus they may be adequately modeled as time-invariant, making signal processing potentially easier; 2) environmental acoustic noise has a minimal influence on the measured acceleration; and 3) the accelerometer's size and placement make it more unobtrusive and comfortable for extended recordings than the current techniques.

This thesis work investigates the potential of using the measured acceleration for quantifying vocal function. Simultaneous acceleration, acoustic and aerodynamic recordings on ten subjects with normal voices are made to examine relationships between the acceleration signal and the acoustic or aerodynamic signal. A vocal system model is also developed to provide insights into these relationships. Specifically, estimates of Sound Pressure Level (SPL) and Maximum Flow Declination Rate (MFDR) are calculated from the acceleration using the vocal system model and show high correlation (r^2>0.71 and r^2>0.75 respectively) with their respective values as derived from the acoustic and airflow signals for 9 out of 10 subjects. These results demonstrate the potential of the acceleration signal to provide an alternate, non-invasive means of obtaining measures of vocal function.

Thesis co-supervisor: Robert E. Hillman, Ph.D.
Title: Associate Professor and Director Voice Disorders Center, Department of Otology and Laryngology, Harvard Medical School

Thesis co-supervisor: Kenneth N. Stevens, Sc.D.
Title: Clarence J. Lebel Professor of Electrical Engineering

BIOGRAPHICAL NOTE

Harold Arthur Cheyne II grew up in Mansfield, Massachusetts. He attended Tufts University, majoring in both Electrical Engineering and Music, from 1989-1993. During his senior year at Tufts, he applied to the Department of Electrical Engineering and Computer Science at the Massachusetts Institute of Technology, hoping to combine his two undergraduate majors in graduate work. Although he was accepted to the Department of Electrical Engineering and Computer Science, Nelson Kiang, then the director of the Speech and Hearing Sciences Program, contacted Harold and urged him to consider this one-year-old program instead. After talking to Nelson for a few hours on a Sunday afternoon, Harold decided that the Speech and Hearing Sciences (SHS) Program contained a valuable balance of basic science coursework and clinical experience that was a great way to continue pursuing both his interest in engineering and music. While in the SHS program, Harold specialized in acoustics and speech processing, and combined these two topics in the thesis work presented here.

This thesis work is dedicated to my wife, Deborah Kiely,
my biggest cheerleader.

TABLE OF CONTENTS

1 Introduction 7
- 1.1 Motivation 7
- 1.2 Goals 8
- 1.3 Contributions, or, the gaps this research will fill 9
- 1.4 Description of contents 11

2 Background 12
- 2.1 Accelerometers 12
- 2.2 Models of the vocal system 13
  - 2.2.1 The subglottal system 13
  - 2.2.2 The glottal source 15
  - 2.2.3 The vocal tract and radiation characteristic 15
- 2.3 Measures of vocal function 17

3 Methods 20
- 3.1 Recordings 20
  - 3.1.1 Subjects and Vocal Tasks 24
  - 3.1.2 Editing and calibrating the acceleration, acoustic and aerodynamic signals 25
- 3.2 Signal analysis 27
  - 3.2.1 Overview and nomenclature 27
  - 3.2.2 Vocal system model 27
    - 3.2.2.1 The mass and resistance of the tracheal wall 28
    - 3.2.2.2 The subglottal transfer impedance, $\mathbf{Z_T(f)}$ 29
    - 3.2.2.3 The glottal source, vocal tract and radiation characteristic 31
    - 3.2.2.4 Estimating MFDR and SPL from the acceleration using the model 31

4 Experiment I: Verifying assumptions used in the vocal system model 35
- 4.1 The effect of vertical laryngeal position (VLP) on the acceleration 36
  - 4.1.1 Predictions using the subglottal transfer impedance model 36
  - 4.1.2 Varying VLP through changes in F0 37
  - 4.1.3 Varying VLP by vocal task 6 – “Pat gave the dog a bag” 45
  - 4.1.4 Discussion of the effect of VLP on the acceleration 47
- 4.2 The effect of lung volume (LV) on the acceleration 49
  - 4.2.1 Varying LV by vocal task 4 – Maximum Phonation Duration 49
  - 4.2.2 Discussion of the effect of LV on the acceleration 52

5 Experiment II: Estimating MFDR from the acceleration using the vocal system model 53
- 5.1 Using the model to estimate MFDR from the acceleration and airflow signals 53
- 5.2 Results from Experiment II 65
- 5.3 Discussion of estimating MFDR from the acceleration 71

6 Experiment III: Estimating SPL from the acceleration using the vocal system model 74
- 6.1 Using the model to estimate SPL from the acceleration 74
- 6.2 Results from Experiment III 80
- 6.3 Discussion of estimating SPL from the acceleration 87

7 Supplemental Experiments: Estimating SPL and the degree of glottal closure without using the vocal system model 97
7.1 Relationship between SPL and the acceleration spectral slope 97
7.1.1 Hypothesis and method of measuring acceleration spectral slope 97
7.1.2 Results of comparing SPL to **H1′-A2′** 98
7.1.3 Discussion of comparing SPL to **H1′-A2′** 99
7.2 Relationship between the degree of glottal closure and the bandwidth of **F1′** 101
7.2.1 Hypothesis and the method of measuring the degree of glottal closure and the bandwidth of **F1′** 101
7.2.2 Results and discussion of comparing the minimum flow to the bandwidth of **F1′** 104
8 Conclusions 108
9 Future work 109
10 Acknowledgements 109
11 Appendices
11.1 Appendix A – subject informed consent form 110
11.2 Appendix B - MATLAB® programs written for the data analysis 115
12 Bibliography 197

1 Introduction

Clinicians and researchers interested in quantifying a person's vocal function, loosely defined as the vocal output for a given vocal effort, have traditionally relied on acoustic, aerodynamic, and electroglottographic techniques. Some typical clinical measures using these techniques are reviewed in Hillman, Montgomery and Zeitels (1997). To infer information about vocal function from an acoustic, aerodynamic or electrical signal, the signal must be processed to estimate what goes on at the glottal level. This signal processing can be fairly simple, such as marking the fundamental frequency, which can be done in any of those three signals mentioned. More complex signal processing might include "inverse filtering" the acoustic or aerodynamic signal, or using the electroglottographic signal to infer vocal fold contact area (Childers et al., 1986). Inverse filtering the acoustic or aerodynamic signal, for example, is done to estimate the acoustic or aerodynamic characteristic of the glottis (source). It assumes that the vocal system is linear and time-invariant over a time interval of a glottal pulse, and that the vocal tract and radiation characteristic act as filters on the source such that their transfer functions can be inverted and their effects removed from the acoustic signal.

One method that has not been thoroughly explored for its potential to noninvasively gather information on vocal function is measuring the acceleration of the skin near the larynx. Researchers have measured acceleration signals that arise from vocalization using miniature accelerometers glued to a subject's neck (Horii & Fuller, 1990; Stevens, Kalikow & Willemain, 1975), but those signals have not been investigated in sufficient detail to determine relationships between the measured acceleration and descriptors of vocal function. A miniature accelerometer could provide an attractive method of extracting vocal function information, due to its immunity to environmental noise, extremely small size, durability, and cost.

1.1 Motivation

Quantitative measurements on acceleration signals exist for examining acoustic information arising from respiratory and nasal activity (Pasterkamp, Schafer & Wodicka, 1996; Pasterkamp et al., 1993; Horii, 1983; Horii & Monroe, 1983; Lippmann, 1981; Horii, 1980), but for vocal function mostly qualitative descriptions of acceleration signals have been reported. Only one report (Horii & Fuller, 1990) has been found which describes the use of measured acceleration to compute quantitative vocal function parameters (F0, jitter, shimmer), but it offers no comparison to the corresponding acoustic measures. Furthermore, none of the above investigators who have used miniature accelerometers for measuring vocal, nasal or respiratory activity have presented a model of the relationship between any quantitative measure and a corresponding physiological mechanism.

Obtaining an acceleration signal for measuring vocal function is a noninvasive procedure, like conventional vocal function measures that use acoustic, aerodynamic, or electroglottographic recordings. In addition, an accelerometer has potential advantages over these other techniques. First, an accelerometer is not as influenced by environmental noise as is a microphone. Second, the filtering of the glottal waveform that is ultimately reflected in the acceleration signal is thought to occur primarily in the subglottal system and the tissues between the airway and the accelerometer. Both of these structures are more or less static over time (this issue is discussed further in Section 3.2.1), and certainly vary much less with time than the vocal tract does during voicing. As a consequence, processing the acceleration signal to estimate glottal waveform parameters is potentially easier than doing so with an acoustic or aerodynamic signal, especially if one can assume that the subglottal system and the tissues between the airway and accelerometer are time-invariant. Third, the accelerometer is comfortable and inconspicuous enough to be worn for several hours, making possible extended recordings for analyzing vocal function.

One practical application of this method of measuring acceleration is a "vocal accumulator", a device for recording about one day's worth of a wearer's voice use. Similar devices, using a microphone as input instead of an accelerometer, have been proposed in the past by Ryu et al. (1983), Ohlsson, Brink & Löfqvist (1989), Masuda et al. (1993), and Buekers et al. (1995). A vocal accumulator using acceleration as input would likely improve clinicians' ability to identify voice misuse. The use of a miniature accelerometer in this application has the advantage that the transducer is small and unobtrusive, minimally affecting a person's typical vocal behavior. Such a device would also be useful for tracking post-surgical patients instructed to maintain voice rest.

1.2 Goals

The goals of this thesis research are to 1) measure the relationship between the acceleration signal and both acoustic and aerodynamic signals for a variety of speakers and speaking conditions, and 2) interpret the relationships between these signals by developing a model of the appropriate vocal system components. These two goals are approached in a parallel, rather than sequential, method so that insights from one can be applied to the other immediately. For example, a relationship in the model may be tested with specific measurements.

Measuring the relationship between either acceleration and acoustic or acceleration and aerodynamic signals is accomplished through simultaneous recordings of acceleration, aerodynamic and acoustic signals from five adult female and five adult male subjects with normal voices. A subject's voice is judged normal if that person has had no history of any voice disorder or laryngeal surgery. The recordings include isolated and continuous speech samples at various vocal efforts.

Relationships between the acceleration signal and either acoustic or aerodynamic signals are investigated first by estimating two conventional parameters of vocal function from the acceleration signal: Sound Pressure Level (SPL) and Maximum Flow Declination Rate (MFDR). These two parameters have not previously been determined using a measure of acceleration. The investigation then briefly explores the possibility of obtaining a third vocal function parameter, a measure of the degree of glottal closure during each cycle of glottal vibration. At a minimum, such a parameter would allow intra-subject comparisons, e.g., "Is one subject achieving a greater degree of glottal closure than another?" Incomplete closure of the glottis can occur in either or both the membranous (between the arytenoid cartilages and the anterior commissure) and cartilaginous (along the arytenoid cartilages) portions of the vocal folds. Shown below in Figure 1.1 are three examples of differing degrees of glottal closure, adapted from Stevens (1977). Ideally, this parameter would describe whether or not a subject achieves complete closure of the vocal folds during each fundamental period, and if not, this parameter's value would relate to the area of the glottal aperture or "glottal chink".

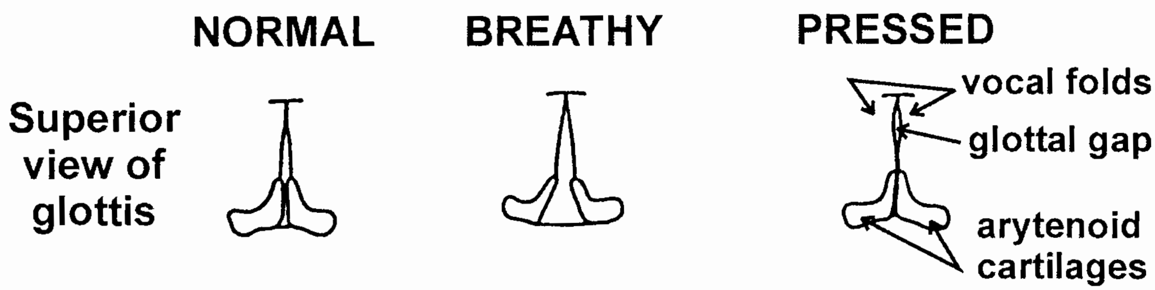


Figure 1.1. Three superior views of the glottis, showing the vocal fold edges, the glottal gap at the time of peak flow through the glottis, and the arytenoid cartilages for three different degrees of glottal adduction: normal, breathy and pressed. A "glottal chink" exists in the cartilaginous portion of the vocal folds (i.e., between the arytenoid cartilages) in normal voice, and increases in area for breathy voice. Breathy voice also shows incomplete closure of the membranous portion of the vocal folds, while pressed voice may show little or no glottal chink or lack of closure in the membranous portion. Adapted from Stevens (1977).

Concurrent with the simultaneous measurement of acceleration, acoustic, and aerodynamic signals is the development of a model of the vocal system to interpret those measurements. The model is an analog system containing both acoustic components, representing the subglottal system, glottis, and supraglottal system, and mechanical components representing the tissues between the subglottal airspace and the accelerometer. Its purpose is to develop quantitative relationships between the measured acceleration and other measurable quantities such as the acoustic SPL. For example, the model should be able to predict the measured SPL given the measured acceleration and some assumption about the vocal-tract transfer function. The model's equations give insight into the glottal mechanisms that produce the measured acceleration and may motivate new vocal function parameters.

1.3 Contributions, or, the gaps this research will fill

This work contributes both to the basic science and clinical knowledge of vocal function. From the basic science perspective, two contributions are made. First, quantifying the relationship between the acceleration signal and the glottal activity will further the understanding of how vocal fold vibrations give rise to the measured acceleration. Second, additional knowledge of vocal system mechanics, especially in the behavior of the tissues between the tracheal airspace and the accelerometer, will be gained through the vocal system model. From the clinical perspective, two other contributions are made. First, measuring the acceleration of the skin on the neck may potentially be a less expensive alternative method for obtaining parameters that describe vocal function as compared to conventional voice clinic equipment. Second, ambulatory monitoring of the acceleration signal as mentioned in Section 1.1 would provide a more accurate picture of a person's typical voice use as compared to patient self-report.

Although the measurement of glottal waveforms with accelerometers has been reported in the literature (Stevens, Kalikow & Willemain, 1975; Horii & Fuller, 1990), the relationship between the measured acceleration and the glottal activity has not been quantified. The measurements of the acceleration of the skin near the larynx provide normal voice data for relating the acceleration signal to normal vocal-fold motion and for exploring the variation in the acceleration signal across normal voices. For example, discussion of how the acceleration signal from subjects with normal voices may vary across vowels, with the degree of glottal closure, with lung volume or with larynx position is presented with the results.

The measurements and proposed model contribute knowledge of normal vocal system mechanics through interpretation of the acceleration waveform based on the proposed model. Current models do not account for motion of the structures superficial to the trachea (e.g., skin, connective tissue, fat), so a model of these structures is combined with current models of the acoustic signal in the larynx, vocal tract and subglottal structures. Refining the model is a continuous process until it adequately describes the relationship between the acoustic or aerodynamic output from the lips and the acceleration signal.

Conventional voice clinic equipment used for obtaining measurements of vocal function include airflow meters, Sound Pressure Level (SPL) meters, and sound-isolation booths. All of these are at least two orders of magnitude more expensive than the accelerometer used in this thesis work. Depending on the accuracy necessary for parameters of vocal function, the accelerometer may be a significantly less expensive method for obtaining those parameters.

This work is also important for the development of devices, like the vocal accumulator mentioned above, to quantify vocal function through measuring acceleration of the skin near the larynx. Ramig & Verdolini (1998) reviewed reports on the frequency of occurrence of voice disorders, and estimate that about 3% to 9% of adults in the United States have voice disorders. A major factor in most voice disorders is how an individual uses his or her voice. Clinical management of voice patients focuses on evaluation and modification of voice use, for which clinicians presently depend on patient self-report and self-monitoring. These self-reports are highly subjective and prone to be unreliable because voice production occurs relatively automatically. A vocal accumulator using an accelerometer could provide an inconspicuous way to monitor and perhaps provide biofeedback on vocal function over several hours.

1.4 Description of contents

Section 2 provides information on past work done in areas relevant to the thesis work. Previous uses of accelerometers are described in Section 2.1. Section 2.2 reviews models of the subglottal system, glottis, and vocal tract, some of which form the starting point for the modeling effort in this work. Section 2.3 discusses some acoustic and aerodynamic measures of vocal function, with particular emphasis on SPL, MFDR, and quantifying glottal closure.

Section 3 describes the methods of the thesis work. The recording apparatus, subjects, and vocal tasks are described in Section 3.1. Data analysis techniques are presented in Section 3.2 as an overview of the section and the nomenclature used (3.2.1), and the presentation of the vocal system model developed for this work (3.2.2).

Section 4 presents and discusses two experiments designed to verify two assumptions made about the vocal system: 1) that vertical larynx position relative to the accelerometer does not significantly affect the measured acceleration (4.1), and 2) that lung volume does not significantly affect the measured acceleration (4.2).

Section 5 outlines and discusses the results for estimating MFDR from the acceleration. It begins with a description of the signal processing (5.1), then a summary of the results and subject-by-subject plots of the MFDR estimates (5.2), followed by a discussion of the strengths and weaknesses of the approach (5.3).

Section 6 outlines and discusses the results for estimating SPL from the acceleration. Like Section 5, it begins with a description of the signal processing (6.1), then summarizes and shows the SPL estimates for each subject (6.2), and then discusses those results (6.3).

Section 7 briefly discusses two experiments and their results for directly estimating SPL (7.1) and the degree of glottal closure (7.2) from the acceleration without the vocal system model.

Section 8 summarizes the conclusions reached from each experiment, and revisits the strengths and weaknesses of the vocal system model.

Section 9 suggests future work to be done in terms of model improvements, additional experiments, and additional analyses of the current data set.

Section 10 acknowledges those people who have a significant impact on this work.

Section 11 includes appendices for reference. Appendix A is the consent form used for the recorded subjects. Appendix B lists the MATLAB® programs created for the data analysis.

Section 12 is the bibliography; it lists the references used in this work.

2 Background

2.1 Accelerometers

Stevens, Kalikow & Willemain (1975) used a miniature accelerometer to extract fundamental frequency and nasalization information from speech for a deaf child's speech training device. For deriving fundamental frequency, the accelerometer was mounted "in the midline between the thyroid cartilage and the sternal notch" (p.595), which reportedly produced the maximal output, as opposed to other mounting locations on the neck. The acceleration signal was found to contain a 500 Hz oscillation whose frequency was vowel-independent but speaker-dependent. The authors presumed that this oscillation was a resonance of the subglottal system. For detecting nasalization, the accelerometer was attached to the side of the nose, and it was noted that nasalized vowels produced acceleration amplitudes that were at least 10 dB greater than those for non-nasalized vowels. Stevens et al. (1976) later described how the detection and quantifying of nasalization using the accelerometer would help to both assess and train the speech of deaf children.

Quantifying the degree of nasalization in speech using accelerometers was further developed by Horii (1980), Lippmann (1981), Horii & Monroe (1983) and Horii (1983). First, Horii (1980) developed the Horii Oral-Nasal Coupling (HONC) index, which required two accelerometers, one on the external nose and the other "between the thyroid and sternal notch" (p.256). The HONC index was defined as the ratio of the amplitude of the nose acceleration signal and a constant times the amplitude of the neck acceleration signal. The index was shown to increase for nasalized sounds and decrease for non-nasalized sounds during normal speech, and to decrease in general for one hyponasal speech sample. Lippmann (1981) compared the signal energy in different frequency bands for two accelerometers: the one used by Stevens et al., and a much less expensive model. He concluded that for detecting nasalization, the inexpensive accelerometer can perform as well as the more expensive model, so clinicians should not avoid the technique due to cost. Later, Horii (1983) reported a high correlation between listener perceptions of the degree of hypernasality in speech samples and 1) the mean HONC index, and 2) the relative frequency of the HONC index being above –12 dB (re the HONC index for [m]). He also proposed a visual and auditory feedback method for training hypernasal speakers using the signals from the two accelerometers, but not computing the HONC index (Horii & Monroe, 1983).

Horii & Fuller (1990) recorded patients preceding intubation and following extubation with a miniature accelerometer to investigate the short-term effects of intubation on the variability of the voice (e.g., jitter and shimmer). The accelerometer was chosen for its insensitivity to environmental acoustic noise, which has the potential to significantly affect frequency and amplitude variability measures. Placement of the accelerometer was similar to that in Stevens et al. (1975). They found that following extubation, the spectral slope of the acceleration significantly decreased and the jitter and shimmer in the acceleration signal increased.

Accelerometers have also been used to record respiratory activity, by Pasterkamp et al. (1993) and Pasterkamp, Schafer & Wodicka (1996). In their 1993 comparison of accelerometers, contact microphones and air-coupled microphones in recording respiratory sounds from the thorax, Pasterkamp et al. found that accelerometers and contact microphones had the least effect on the spectrum of the desired signal. They also found that the sensitivity of their accelerometer was significantly lower than that of the contact microphone. However, given that normal respiratory signals are much lower in amplitude than normal phonatory signals, this difference in sensitivity will not affect this work. Pasterkamp, Schafer & Wodicka used an accelerometer mounted on the skin over the trachea between the cricoid and sternal notch to show that subjects with obstructive sleep apnea produce significantly larger accelerations, as measured by the energy in the acceleration signal over frequency, than do subjects in the control group.

2.2 Models of the vocal system

2.2.1 The subglottal system

An impedance model of the subglottal system based on human data was first developed by Fant et al. (1972). To test if the extra peaks they detected in speech spectra were subglottal formants, they measured the acoustic input impedance at the tracheostoma of five Japanese subjects who had undergone laryngectomy. Their first model of the subglottal system was implemented on Fant's "electrical vocal-tract analog" (Fant et al., 1972, p.5) and based on measured lengths and areas of the trachea and bronchial tree. It contained a series of three-element T-branches, shown in Figure 2.1. This first model produced subglottal resonances that were too low in frequency compared to those measured from the laryngectomy data, so three other elements were added to compensate for the shift in frequency. The mass and resistive components L_B and R_B, and the capacitive component C_A, were given values such that the input impedance of the subglottal system (i.e., looking to the right of the dotted line separating the "glottal source" and "trachea and bronchial tree" sections in Figure 2.1) had peaks at the frequencies of the observed subglottal resonances from the laryngectomy data. According to Fant et al., these components could represent the mass and resistance of the tissue connecting the tracheal rings, and the capacitance of the alveoli respectively. Although these additions brought the frequencies of the resonances and anti-resonances in the model and data into agreement, it did not account for the increased damping of the subglottal formants with increasing frequency above the second subglottal formant. Figure 2.1 schematically shows the complete vocal system model proposed by Fant et al., which was used to investigate the appearance of subglottal formants in speech. The complete model includes a variable glottal mass L_G and glottal resistance R_G to allow variation in the gap between the vocal folds, a source V_S that provides the excitation to the vocal system, and a vocal tract model that is explained more in Section 2.2.3.

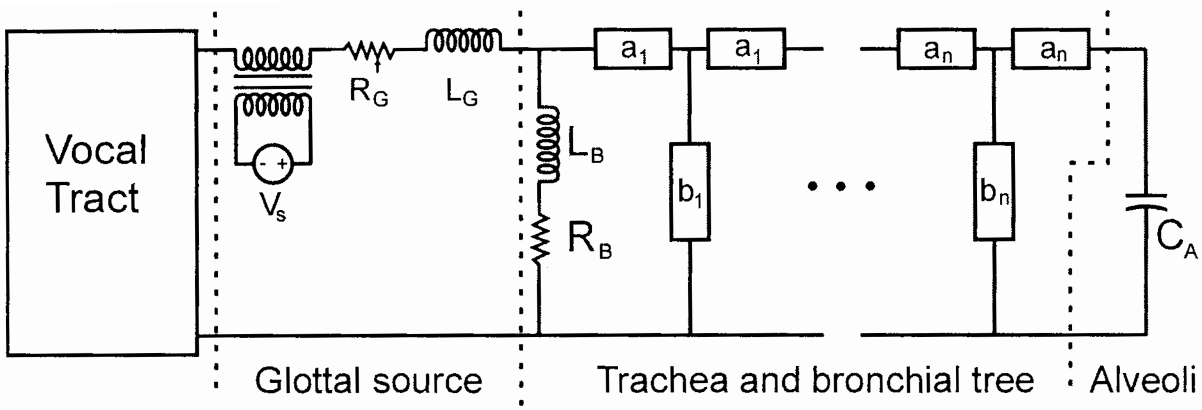


Figure 2.1. Vocal system model, adapted from Fant et al. (1972), consisting of four main components: the vocal tract, which is modeled as a series of three-element T-branches (not shown); the glottal source V_S including a glottal resistance R_G and L_G to allow variable glottal gaps; the trachea and bronchial tree, which is modeled as a series of three-element T-branches with a mass L_B and resistance R_B representing the impedance of the muscle tissue connecting the tracheal and bronchial rings; and the compliance of the alveoli C_A.

Ishizaka, Matsudaira and Kaneko (1976) expanded on the investigation of Fant et al. by further developing an anatomical model based on the five Japanese subjects who had undergone laryngectomy. They reported that the first three resonances in the measured acoustic input impedance at the stoma occur at about 640, 1400 and 2100 Hz, with the impedance having the largest magnitude at the frequency of the second resonance. To model the data, they developed an analog of an acoustic model based on the anatomy of the trachea, bronchi and lungs, which included losses due to non-rigid walls in the system. The trachea and larger bronchi were modeled by seventeen interconnected two-ports, all with nine elements similarly connected but different in element values. Seven other interconnected seven-element two-ports represented the smaller bronchi, while a one-port load impedance represented the alveoli. Although the model has a high complexity, it reproduced many of the features of the measured acoustic input impedance of the subglottal system.

Spectral analyses of direct measurements of the subglottal pressure during normal speech were conducted by Cranen and Boves (1987). They found a discrepancy between the frequency of their measured subglottal spectral peaks (510, 1355, 2290 Hz) and those reported by Fant et al. (1972) and Ishizaka, Matsudaira and Kaneko (1976). They postulated that a glottis that does not close completely during phonation could produce a secondary excitation wave during the "closed-glottis" interval that would result in destructive interference near the frequency of the first subglottal formant, producing the observed decreased frequency of that subglottal formant. A model of the glottis, based on the two-mass model of Ishizaka and Flanagan (1972), was introduced with a bypass leak to support this hypothesis.

2.2.2 The glottal source

A lumped element model of the glottal source that allows a variable glottal impedance is presented in Stevens (1998), and shown below in Figure 2.2. This model is fairly simple in that it has only three components – the glottal impedance Z_G, and two volume velocity sources V_{VG}. However, it provides versatility for a lumped element vocal system model since the glottal impedance can be an open circuit (i.e., $Z_G \rightarrow \infty$), meaning there is no coupling between the subglottal and supraglottal systems. This allows the subglottal and supraglottal systems to be analyzed separately. Or, glottal coupling can be realized with a finite Z_G, perhaps more accurately modeling most speech, in which a small "glottal chink" is present.

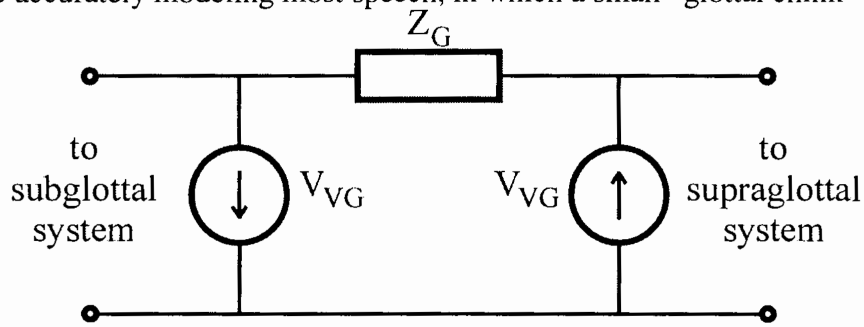


Figure 2.2. Glottal source model with variable glottal impedance Z_G and volume velocity sources V_{VG} exciting both the supraglottal and subglottal systems.

A mathematical glottal waveform model with four parameters was developed by Fant, Liljencrants and Lin (1985), and is known as the LF-model. The LF-model describes the derivative of the glottal flow, or $\frac{dv_{VG}(t)}{dt}$, and its parameters allow control over the spectral effects due to incomplete glottal closure. Modeling these effects is important in light of the findings of Cranen and Boves mentioned above. The degree of closure in the model can be controlled, along with the pulse amplitude, fundamental frequency and duty cycle using the four interdependent parameters of the model. The parameters depend on one another because one constraint on the model is that the area under the flow derivative over one fundamental period must be zero. With this model, they demonstrated the effects of varying the four parameters and fitted one subject's actual flow measurements. Varying the model's parameter values showed that the model could produce flow curves from one extreme, complete closure, to the other extreme of a sinusoidal flow pattern. When compared to the actual flow measurement of a male subject with breathiness, the LF-model showed a reasonable fit.

2.2.3 The vocal tract and the radiation characteristic

The source-filter theory of speech, introduced by Fant (1960), has been used by many investigators for a myriad of purposes from basic acoustic analyses of a speech sound to coding speech waveforms for reducing the amount of data needed to store speech. This theory models the vocal tract with some type of acoustic equivalent circuit. This vocal tract model acts as a filter to the sound generated by the glottal source. Fant proposed treating the vocal tract as a series connection of T-network circuits, just as the trachea and bronchial tree were modeled above. Alternatively, he states that the vocal tract can be modeled using analogous circuits for acoustic horns, which is useful for vocal tract configurations with continuous area changes as it makes for less computation than the T-network model.

Stevens (1998) provides an extensive explanation of vocal tract acoustics from the source-filter theory view. He uses an approximation based on the theory of one-dimensional propagation of sound in tubes. He contends that this approximation holds for frequencies whose wavelengths are much longer than the cross-section of the vocal tract, which is true for frequencies less than 5000 Hz. In conjunction with perturbation theory from electric network theory, Stevens shows that the prominent vocal tract resonance(s) for any phoneme can be estimated from the geometry of the vocal tract for that phoneme.

The acoustic pressure measured at a distance r from the lips is related to the volume velocity output at the mouth by the radiation characteristic R(f). Fant (1960) gives the radiation characteristic for a piston vibrating in a sphere as

$$\frac{p_r}{U_0} = \frac{\rho\omega}{4\pi r} K_T(\omega)$$

Equation 2.1

where

$p_r \equiv$ the sound pressure measured at the distance r,
$U_0 \equiv$ the volume velocity at the mouth,
$\rho \equiv$ the density of air, and
$K_T(\omega) \equiv$ a correction factor for the baffling effect of the sphere.

Fant also provides a plot of the correction factor (in dB) versus frequency, showing the correction factor at 0 dB for low frequencies, then monotonically increasing from about 0 dB at 200 Hz to about 6 dB at 5000 Hz.

Stevens (1998) gives a similar equation for the radiation characteristic, without the correction factor, stating that for "a first approximation . . .the mouth opening can be regarded as a simple source" (p.127):

$$R(f) = \frac{p_r(f)}{U_0(f)} = \frac{j2\pi f\rho}{4\pi r} e^{-j\frac{2\pi fr}{c}}$$

Equation 2.2

Or, if we are considering only the magnitude of the radiation characteristic,

$$|R(f)| = \frac{f\rho}{2r}$$

Equation 2.3

In Equations 2.2 and 2.3, $p_r(f)$ is the pressure at a distance r, c is the speed of sound, and the other parameters are as described above.

2.3 Measures of vocal function

Clinical assessment of voice incorporating acoustic and aerodynamic measures was reviewed by Hillman, Montgomery and Zeitels (1997). They state that typical clinical non-invasive aerodynamic measures include transglottal air flow, estimated from the pressure drop across a known acoustic resistance, and average subglottal pressure, estimated from an oral pressure tube. Typical acoustic measures include average fundamental frequency (F0), average sound pressure level (SPL), jitter, shimmer, and harmonics-to-noise ratio.

For comparing normal and abnormal vocal function, Hillman et al. (1989) selected several aerodynamic measures and one acoustic measure: oral volume velocity, intraoral air pressure, flow open quotient, flow speed quotient, flow closing quotient, AC flow, minimum flow, peak flow, AC/DC ratio of flow, maximum flow declination rate (MFDR), and SPL. In describing their methods, they state that MFDR is "considered an indirect estimate of maximum vocal fold closing velocity and . . . related to the abruptness of vocal fold closure" (p. 380). Although direct measurements of MFDR and vocal fold closing velocity have not been reported, they reference a vocal fold model by Titze (1984) as evidence for a relation between MFDR and vocal fold velocity. Titze's model includes a direct variation of the glottal flow with the vocal fold displacement; therefore it implies that the derivative of the glottal flow varies directly with vocal fold velocity, or more specifically that MFDR varies directly with maximum vocal fold velocity. Furthermore, Titze showed that measures of glottal area by photoglottography and vocal fold contact area by electroglottography on one male subject (presumably normal) could be accurately reproduced with his model. Thus Titze's model, and the relationship between MFDR and vocal fold velocity, appears reasonable for normal male speakers at least. Hillman et al. (1989) also hypothesized that as a subject's MFDR increased, so should the potential for her/him to incur trauma to the vocal folds. Their results supported a link between organic vocal pathology and increased MFDR, as they found that their subjects who had histories of organic vocal pathologies, such as nodules, displayed abnormally high values of MFDR. However, their results could not establish whether the increased MFDR was a contributing factor to the development of the pathology or if it was secondary to the pathology.

Sapienza & Stathopoulos (1994), in their comparison of children's and adult's MFDR values, also state that there is a positive correlation between MFDR and how quickly the vocal folds close, again referencing the model by Titze (1984). They hypothesize that two mechanisms could lead to an increased MFDR, either the vocal folds exhibit "greater recoil" (p. 240) due to high amplitudes of vibration, or due to a larger Bernoulli effect from larger airflows.

Breathy phonation, due to incomplete glottal closure, was one type of phonation studied by Klatt and Klatt (1990) using acoustic measures, perceptual tests of actual speech, and perceptual tests of synthesized speech. They stated that other investigators had compared breathy speech to normal speech and had found that inverse filtering the measured airflow showed breathy speech has increased average flow, a more sinusoidal airflow waveshape, and a relatively longer open period. For potential objective measures of breathiness in speech, they listed the following acoustic cues:

1) H1-H2, where H1 and H2 are the amplitudes of the first and second harmonics in the acoustic spectrum, gives a measure of the relative strength of H1. Klatt and Klatt reference acoustic and perceptual studies showing that the relative strength of H1 increases with the open quotient (the percentage of time the vocal folds are open during one fundamental period), and state that increases in open quotient "might be expected for a breathy voice quality"(p. 827).

2) The amount of noise in the waveform should increase with breathiness, presumably since more turbulent air is exciting the vocal tract when the glottis does not close completely. This could be measured by evaluating the lack of periodicity of the acoustic waveform at high frequencies (e.g., around F3, the third formant).

3) The bandwidth of the first formant (F1) increases with breathiness, since the glottal impedance introduces more loss as glottal closure becomes less complete.

4) The appearance of extra poles and zeros in the spectrum, particularly "extra resonances" around 1650 and 2350 Hz on average, could occur due to the increased tracheal coupling from incomplete glottal closure.

In their perceptual tests using samples of actual speech, Klatt and Klatt report that the first two measures listed above, H1-H2 and the amount of noise, had a significant correlation with the perceived breathiness in the speech. Furthermore, subjects in their perceptual tests of synthesized speech reported the most natural-sounding breathiness was achieved when the synthesis included a large spectral tilt, a greater than normal amount of aspiration noise, a widened bandwidth of F1, and an increase in the open quotient (which affects H1-H2).

Holmberg et al. (1995) made direct comparisons between airflow and acoustic measurements of female speech. One of their goals was to relate the acoustic H1-H2 measure to the degree of glottal abduction. They found a correlation between H1-H2 and their flow adduction quotient, which relates to the amount of time per fundamental period that the vocal folds are closed. Another goal of their study was to relate the DC airflow, which they assumed would be due to a glottal chink, to one of their acoustic measures, but they did not find consistent correlation between the measured DC airflow and any of their acoustic measures.

Hanson (1997), in her study of female voicing source acoustic characteristics, modeled three spectral cues to incomplete glottal closure listed above from the Klatt and Klatt (1990) study, and measured several acoustic parameters corresponding to those changes. She stated that a change in the bandwidth of F1, one of those spectral cues, could "provide an indirect indication of the degree to which the glottis fails to close completely during a cycle of glottal vibration."(p.470). Using a model of the voicing source, glottis, and vocal tract, she described how the bandwidth of F1 would change as a function of the glottal chink area during the closed cycle with the following equation:

$$B_g = \frac{\rho c^2}{\pi A_v l_v R_{ch}(1 + 4\pi^2 f^2 M_{ch}^2 / R_{ch}^2)}$$

Equation 2.4

where

$B_g \equiv$ the contribution of the glottal chink to the first formant bandwidth (Hz), such that $B1 = B_g + B_v$, where B_v is the contribution of the vocal tract to the first formant bandwidth
$\rho \equiv$ the density of air in the vocal tract (g•cm^{-3})
$c \equiv$ the speed of sound (cm•s^{-1})
$A_v \equiv$ cross-sectional area of the vocal tract (assuming uniform cross-sectional area, in cm^2)
$l_v \equiv$ length of the vocal tract (cm)
$R_{ch} \equiv$ glottal resistance due to the glottal chink (dyn•s•cm^{-5})
$f \equiv$ the formant frequency (Hz)
$M_{ch} \equiv$ the acoustic mass of the glottal chink (g•cm^{-4})

The area of the glottal chink, A_{ch}, can be related to the acoustic mass of the glottal chink by the equation

$$M_{ch} = \frac{\rho l_g}{A_{ch}}$$

Equation 2.5

where l_g is the vertical "length" of the glottis.

Hanson also used a lumped-element electrical model of the trachea, glottis, and vocal tract to relate the tilt of the acoustic spectrum to the area of the glottal chink. To measure the spectral tilt from the vowel portion of a word "bVd" in a carrier phrase, where V was one of the vowels {/æ/, /ɛ/, /ʌ/}, Hanson subtracted the amplitude of the harmonic nearest the third formant from the amplitude of the first harmonic (H1 – A3), and corrected for the effect of F1 and F2 on the amplitude A3.

3 Methods

Chapters 1 and 2 provide motivation and background information for investigating the acceleration of the skin on the neck as a method of measuring voice use, in particular Sound Pressure Level (SPL) and Maximum Flow Declination Rate (MFDR). In this chapter, the experimental recording methods for capturing simultaneous acceleration, acoustic and aerodynamic signals are described. Following the recording description, the vocal system model developed for this work is presented, and then an overview of the use of the model in providing estimates of MFDR and SPL from the acceleration is given.

3.1 Recordings

Prior to recording each subject, the experimental procedures and tasks were explained to the subject, and each subject provided written informed consent on the form that appears in Appendix A. All subjects received compensation for their participation in the experiment.

All experimental recordings took place in a sound-isolation booth (Eckoustic Noise Control Products Model XHD Type 3), with the subject seated in an otolaryngological examination chair. Subjects were instructed to make themselves comfortable at the start of the experiment, and then asked to maintain their posture and position in the chair during the experiment. The subject and chair were in direct view of the experimenter through the window of the sound-isolation booth, so any significant change in the subject's posture or position could be observed through the window.

The equipment used for recording the accelerometer, aerodynamic, acoustic and electroglottographic signals is schematized in Figure 3.1. The accelerometer (Knowles BU-7135) was attached at midline about 1cm above the sternal notch using Smith & Nephew Skin-Bond®. The two-channel EGG electrodes (Glottal Enterprises model MC2-1) were placed over the subject's thyroid cartilage and held by an elastic strap with hook-and-loop closure ends. The microphone (Sennheiser MKE2) was attached to a flexible lamp holder that attached to the chair, and positioned at midline at the level of the subject's lips and 15cm away from the subject's lips. Following the initial microphone calibration recording described below, the Rothenberg mask, used for measuring oral air flow, was fitted over the subject's nose and mouth by four adjustable straps that connected to a brace behind the subject's head.

The output of the accelerometer had its DC offset removed by a preamplifier that was custom-made by the author. This preamplifier circuit also provided the DC power for the accelerometer, and the schematic for it appears in Figure 3.2. A frequency response plot of the preamplifier's transfer function appears in Figure 3.3. The Rothenberg mask output was filtered and amplified by a Glottal Enterprises model MS-100. The wire mesh of the Rothenberg mask was heated using the built-in heater current of the device to

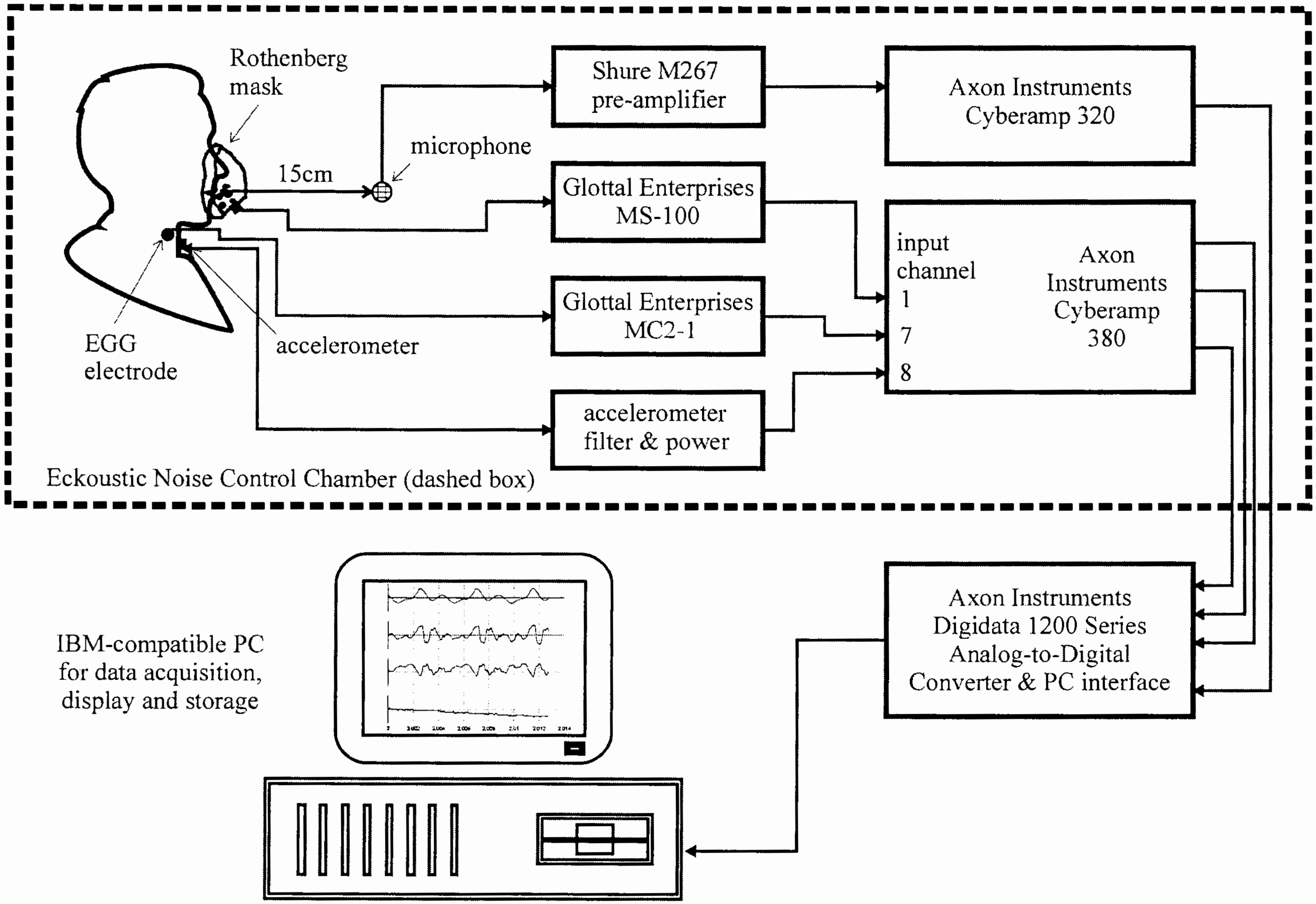


Figure 3.1. Experimental recording block diagram.

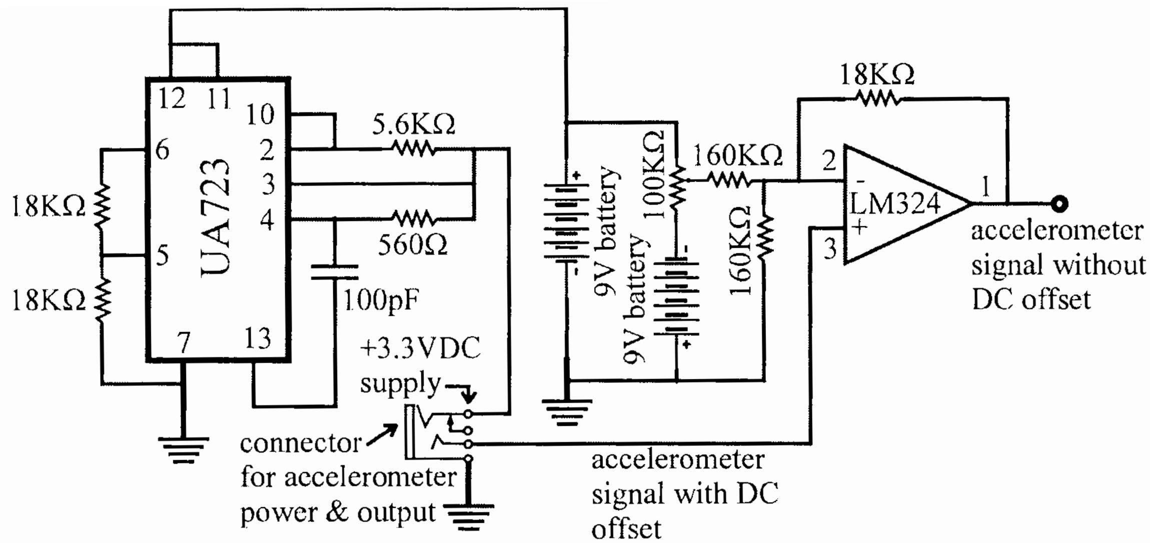


Figure 3.2. Schematic of the custom-made accelerometer power supply and DC offset adjustment preamplifier circuit.

avoid condensation. The output of the microphone was pre-amplified (Shure model M267). The laryngeal tracking signal of the two-channel EGG was recorded to monitor the vertical displacement of the larynx in the neck.

All four signals were amplified and low-pass filtered for anti-aliasing at 6kHz (except for the EGG signal that was low-pass filtered at 20Hz) by an Axon Instruments CyberAmp (model 380 for acceleration, EGG and airflow; model 320 for microphone). The gain settings for each signal were adjusted individually for each subject, and were chosen subjectively by the experimenter to obtain the best Signal-to-Noise Ratio given the Analog-to-Digital (A/D) input range of ±10 Volts. An Axon Instruments Digidata (series 1200) digitized all four signals at 20kHz and was the interface between the digital signals and the IBM-compatible PC that displayed and stored the data. The digitized signals were displayed simultaneously during recording and stored on a hard disk in .ABF file format using Axon Instruments Axoscope software.

Once the subject was comfortably seated and the EGG electrodes, microphone, and accelerometer had been fitted, s/he was asked to sustain a vowel at normal intensity for a few seconds and then at a louder intensity for a few seconds. The initial gain for the microphone and accelerometer signals were chosen based on these speech samples. The microphone signal was then calibrated using a Cooper-Rand artificial larynx source held at the subject's closed lips, while an SPL meter (RION model NL-11) was held near the microphone. Following the microphone calibration, the Rothenberg mask was fitted and informally checked for leaks by having the subject exhale while the experimenter felt around the edge of the mask. The subject was again asked to sustain a normal and a louder intensity vowel to set the initial gain for the airflow signal. Lastly, the subject was instructed to swallow several times, causing the EGG signal to go through its full excursion, and the gain for the EGG signal was set appropriately. No attempt was made to calibrate the EGG signal, because previous authors (Milstein, 1999; Elliot, Sundberg & Gramming, 1992; Rothenberg, 1992) have noted that the absolute value of the vertical laryngeal position (VLP) signal provided by the EGG can be prone to error. However, the relative changes in the EGG signal have been noted as being a reasonable indicator of the direction of VLP change (Milstein, personal communication 2000). In interpreting the results of these experiments, the direction of VLP change (i.e., rostrally or caudally)

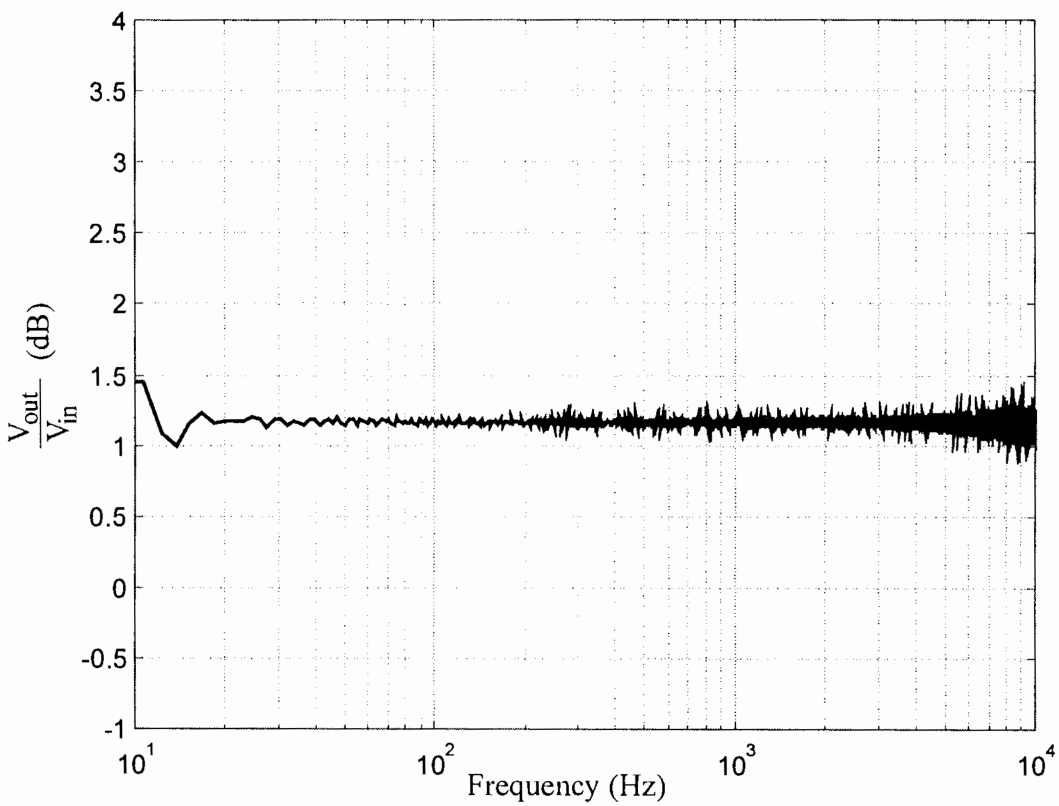


Figure 3.3. Transfer function of the custom-made accelerometer preamplifier for V_{in} = 0.1 Volts. The mean value over the frequency range shown is 1.2 dB. This mean value is maintained ±0.1dB for $0.003 \leq V_{in} \leq 0.5$ Volts.

will be inferred from the change in the EGG signal, with no absolute calibration of the signal with respect to distance.

3.1.1 Subjects and vocal tasks

Ten adult subjects with normal voices, five females and five males, participated in these recordings. As mentioned in Section 1.2, a subject was judged to have normal voice if s/he had no history of laryngeal surgery or disorder.

After the calibrations and gain settings, the subject was instructed to perform the following vocal tasks:

1) sustaining vowels (/a/ as in father, /ae/ as in bat, /i/ as in see, /o/ as in boat, /u/ as in boot) at normal, quieter than normal, louder than normal, as quietly as possible, and twice as loud as normal;
2) saying vowel-consonant-vowel (VCV) triplets in the carrier phrase "Say VCV again";
3) sustaining vowels (/a/, /ae/, /i/, /o/, /u/) using three different voice qualities that are associated with varying degrees of glottal adduction: normal, breathier than normal, more pressed than normal;
4) sustaining /a/ as long as possible (maximum phonation duration);
5) sustaining /a/ while varying pitch from normal to lowest (excluding vocal fry) then to highest (including falsetto);
6) saying the sentence "Pat gave the dog a bag";
7) repeating the first task with the Rothenberg mask removed.

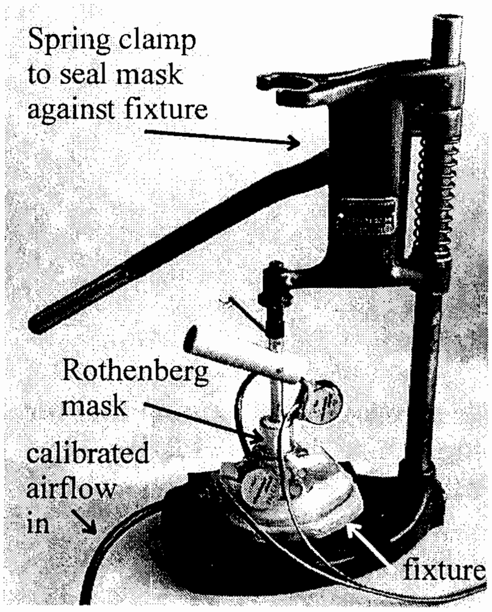


Figure 3.4. Fixture, clamp, and airflow input for Rothenberg mask calibration device.

For each subject, the experimenter completed a check-list to ensure that all tasks were performed, and to note if and why any tasks needed to be repeated. After the subject had completed those tasks, the Rothenberg mask was calibrated by fitting it to the fixture shown in Figure 3.4. A series of constant flows, maintained by a calibrated flow meter (F&P Co., model FP1/4-25-G-5/36) was introduced to the mask while the airflow signal and microphone signal (for recording announcements of the flow values) were recorded. Flow values of 0, 24, 18, 12, 8 and 4 units were used in the flow calibration. These flow units were later converted to values in $\frac{cm^3}{\text{sec}}$ and used for the flow signal calibration.

3.1.2 Editing and calibrating the acceleration, acoustic and aerodynamic signals

The MATLAB® script abf2mat (see Appendix B) was written to automate the process of converting the signals from Axoscope format (.ABF files) to MATLAB format (.MAT files), then editing and calibrating the acceleration, acoustic and aerodynamic signals obtained in the experiments. The acceleration signal has a known correspondence to absolute acceleration, as given in Figure 3.5. This and the subject-dependent gain for the acceleration signal were used to convert the acceleration signal from Volts to $\frac{cm}{\sec^2}$. The acoustic signal was calibrated using the recorded Cooper-Rand sound source. The root-mean-square (RMS) value of a section of the Cooper-Rand excitation was calculated and compared to the recorded SPL meter level to create a conversion factor of Volts to $\frac{dyne}{cm^2}$. For the aerodynamic calibration, a linear least-squares fit was found for the five non-zero constant flow measures described above and the point [0 Volts, 0 cm^3/sec], to give a conversion factor from Volts to $\frac{cm^3}{\sec}$.

After calculating the calibration conversion factors, data from each task were first edited to remove any signals that were unrelated to the task (i.e., silence between tasks, extra instruction given before a task, coughs, etc.), then converted to absolute units using the conversion factors, and then saved as a .MAT file.

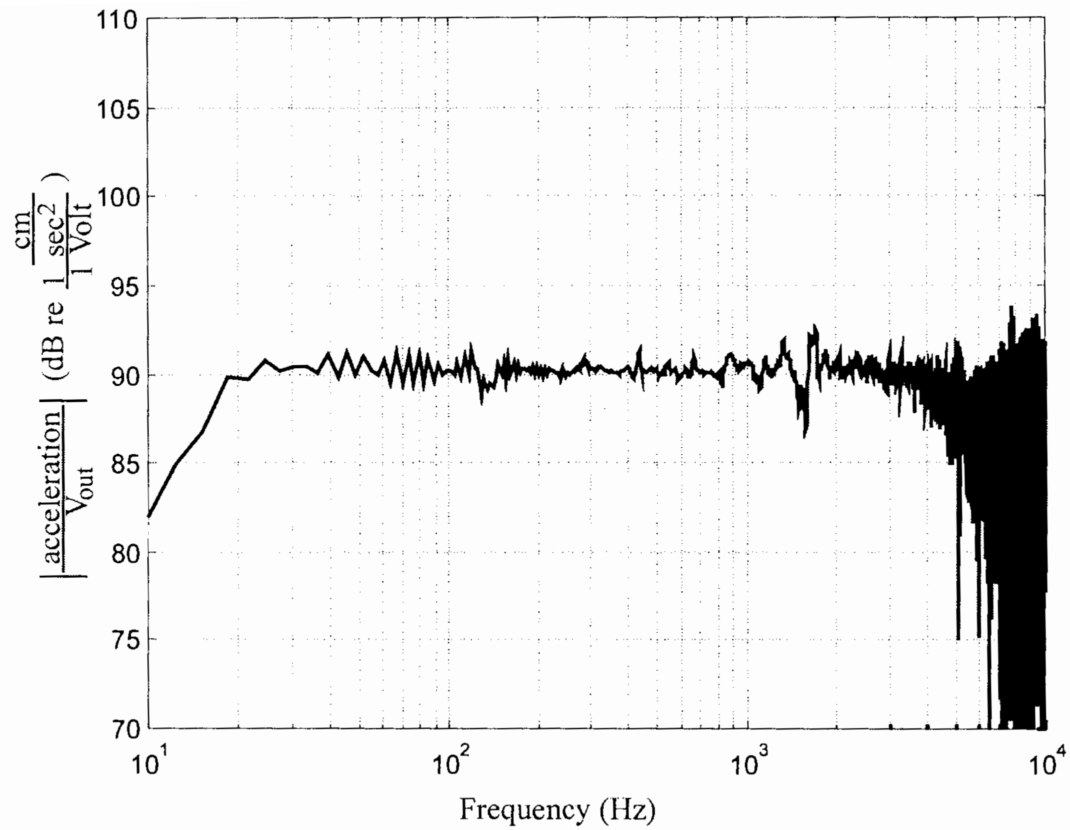


Figure 3.5. The magnitude of the ratio of acceleration to the accelerometer output voltage in cgs units. Note that over the frequency range of interest (70-3000Hz) the ratio is flat at 90 dB ± 3 dB re 1 cm/sec²/Volt.

3.2 Signal analysis

3.2.1 Overview and nomenclature

All signal analysis was accomplished using custom programs within MATLAB®, written by the author unless otherwise noted. These programs appear in Appendix B for reference. Where possible, signal analysis techniques were fashioned after similar techniques in the literature, which are referred to below in the corresponding descriptions of the signal analysis procedures for each experiment.

Throughout the following sections, italics are used to denote the names of signals, spectra, impedances, transfer functions, and model parameters. Uppercase italic letters refer to frequency-domain functions, while lowercase italic letters refer to time-domain functions. Subscripts link italic names to model components. Some particular letters' meanings are

z, Z	for impedances
v_V, V_V	for acoustic volume velocity
f	for frequency (Hz),
BW	for bandwidth (Hz), and
Hn	for spectral harmonic amplitudes.

For example,

$v_{VGA}(t)$	refers to the acceleration-derived glottal volume velocity time signal
$T(f)$	refers to the vocal tract transfer function
$H2$	refers to the amplitude of the second harmonic in the spectrum

And in particular, the following symbols and letters refer to the specific quantities listed.

c	equals the speed of sound, 35400 cm/sec at 37°C, 34500 cm/sec at 22°C
ρ_0	equals the density of air, 0.00112 g/cm^3 at 37°C, 0.00114 g/cm^3 at 22°C

3.2.2 Vocal system model

The vocal system model developed for this work incorporates a combination of some past investigators' models mentioned in Section 2.2, plus a representation of the accelerometer, the subglottal airspace between the glottis and accelerometer mounting point, and the tissues between the subglottal airspace and the accelerometer. This model has five main components as shown in Figure 3.6: the subglottal transfer impedance, the mass and resistance of the skin and accelerometer, the glottal source, the vocal tract, and the radiation characteristic seen looking out of the mouth. It is an electrical analog of the acoustic and mechanical models of its components, with pressure being analogous to voltage and volume velocity being analogous to current. Descriptions of these five main components and how they relate the measured acceleration to the MFDR, SPL, and the degree of glottal closure are discussed in this section.

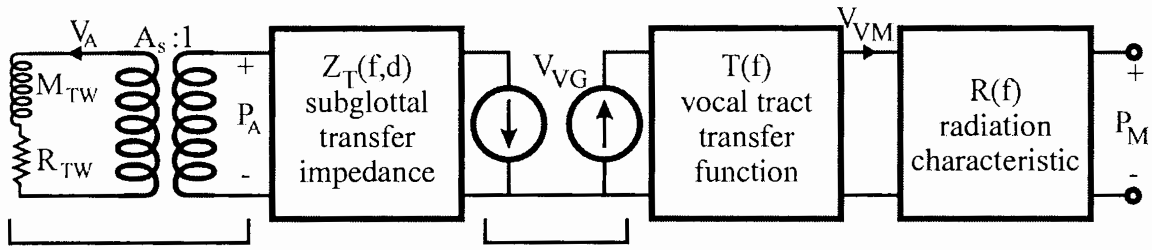


Figure 3.6. The vocal system model, consisting of five main components: 1) the mass and resistance of the tracheal wall, including a mechanical-domain-to-acoustic-domain transformer; 2) the subglottal transfer impedance; 3) the two glottal volume velocity sources needed since complete closure is assumed; 4) the vocal tract transfer function; and 5) the radiation characteristic. Note that the signals to be measured are shown in the model as P_M (microphone pressure signal), V_{VM} (airflow signal), and V_A (velocity of tracheal wall, or the integrated acceleration signal).

3.2.2.1 The mass and resistance of the tracheal wall

Mass $\mathbf{M_{TW}}$ and resistance $\mathbf{R_{TW}}$, the only mechanical parts of the model, incorporate the moving mass and the damping qualities of the accelerometer and all the structures between the tracheal airspace and the accelerometer (e.g., tracheal cartilage, fat, skin). An acoustic-to-mechanical transformer with turns ratio $\mathbf{A_S}$**:1** connects the mechanical mass and resistance to the acoustic model, where $\mathbf{A_S}$ is the area under the accelerometer. This two-element model represents a reasonable approximation over the frequencies of interest (60 to 3000 Hz) given past models of vocal tract wall impedance by Ishizaka, French, and Flanagan (1975), by Fredberg & Hoenig (1978), and by Habib et al. (1994).

Ishizaka, French and Flanagan (1975) measured the mechanical input impedance of the neck wall and modeled the yielding wall of the vocal tract in the neck region as a mass, compliance, and resistance connected in series, with a resonant frequency of 72 Hz. Using their reported values of mass per unit area and resistance per unit area, along with the area under the accelerometer ($\mathbf{A_S}$ = 0.448 cm^2), equivalent values for $\mathbf{M_{TW}}$ and $\mathbf{R_{TW}}$ are 1.1 grams and 1039 grams/second respectively. Because their model's resonant frequency is quite close to the lowest fundamental frequencies of interest, including the resistance $\mathbf{R_{TW}}$ in the vocal system model here may more accurately portray the driving-point admittance $\frac{V_A}{F_A}$ for frequencies around the first harmonic.

Fredberg & Hoenig (1978) proposed an equivalent circuit model for the subglottal system based on a branching network, with parameters derived from reported lung morphometry. In contrast to the findings of Ishizaka, French & Flanagan (1975), their value for the tracheal specific wall inertance of 0.302 grams/cm corresponds to an $\mathbf{M_{TW}}$ of 0.34 grams. However, Habib et al. (1994) refer to the work Fredberg & Hoenig (1978) and state, "Fredberg and Hoenig did not have information available about the wall thickness as a function of airway order" (p.449), and this lack of information may have led to their underestimating the mass.

Habib et al. (1994) constructed an electric analog model of the subglottal airway containing two parallel branches for each section of the airway wall, one representing the cartilaginous portion of the wall and one representing the soft tissue portion. Each branch consisted of a series connection of a resistance, compliance and mass. With an approximate average value for their model's specific airway inertance and resistance at the level of the trachea from seven subjects of 10^{-3} cm $H_2O \times sec^2$ and 10 cm $H_2O \times sec$ respectively, equivalent values of $\mathbf{M_{TW}}$ and $\mathbf{R_{TW}}$ are 1.1 grams and 11000 grams/second.

With these studies in mind, the vocal system model's tracheal wall mass value should be on the order of 1 gram. The tracheal wall resistance, if needed because a subject's tracheal wall resonance appears close to his or her fundamental frequency, should be between 10^3 and 10^4 grams/second.

3.2.2.2 The subglottal transfer impedance, $\mathbf{Z_T(f)}$

The subglottal transfer impedance is based on the measurements of Ishizaka, Matsudaira & Kaneko (1976). However, instead of basing it on the anatomy of the subglottal system and adjusting the model to match their measurements, the transfer impedance $\mathbf{Z_T(f)}$ is based on a "perturbation black-box" model. It is treated as a system of poles and zeros, which do not necessarily correspond to anatomical features of the subglottal system, but that can be manipulated to produce a desired input impedance and transfer impedance. Starting with the driving-point impedance measured by Ishizaka, Matsudaira & Kaneko (1976), a set of poles and zeros was chosen to produce a reasonable estimate of their measured impedance. Figures 3.7a and b show the pole-zero plot and the resulting driving-point impedance respectively. This impedance is used as the starting point for creating the transfer impedance $\mathbf{Z_T(f)}$ of each subject, after which the poles and zeros are manipulated in a matching procedure described in Section 5.1. To convert the driving-point impedance shown in Figure 3.7 into the desired transfer impedance, we assume the acoustic model of the trachea and subglottal system shown in Figure 3.8.

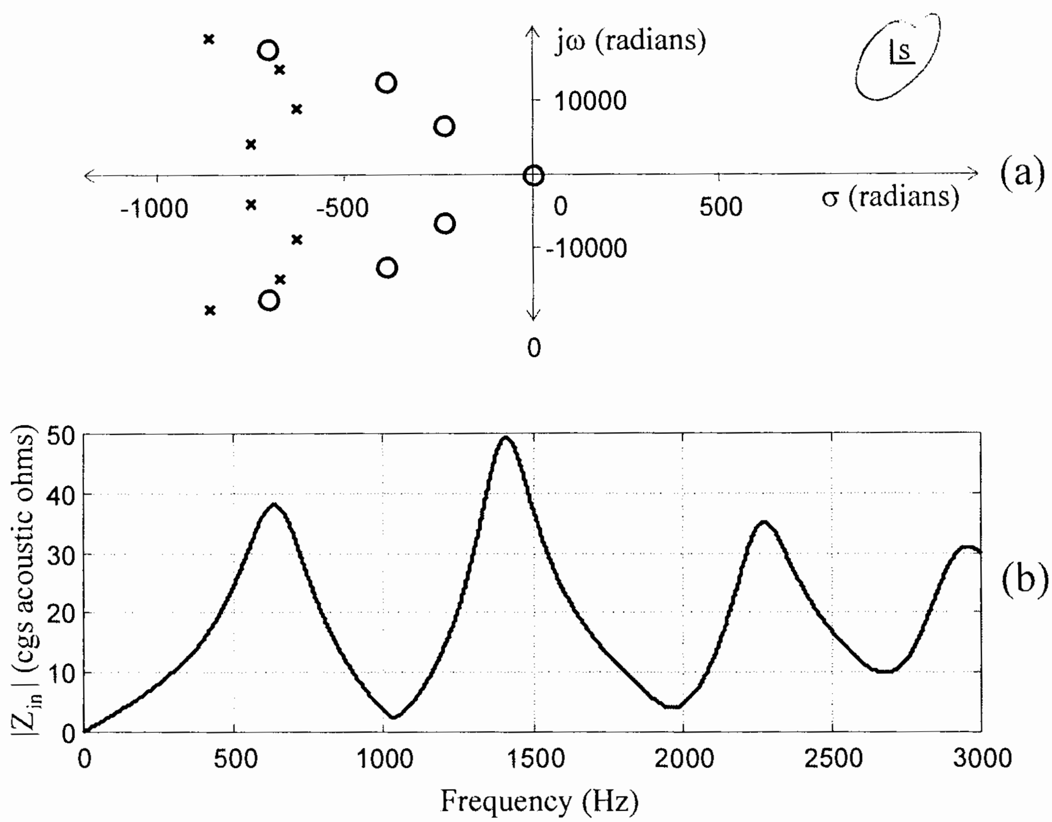
Assume the input impedance to the subglottal system, $\mathbf{Z_{IN}}$, equals the driving-point impedance shown in Figure 3.7b. Then model the subglottal system as a lossless acoustic transmission line of length 5cm and cross-sectional area A=2.5 cm^2, with a load impedance $\mathbf{Z_L}$. With this model, an accelerometer placed somewhere inferior to the glottis at location $\mathbf{L_A}$ will experience an acoustic pressure $\mathbf{P_A}$ through the tracheal wall. Knowing $\mathbf{Z_{IN}}$ and $\mathbf{L_A}$, the transfer impedance can be solved as

$$Z_T(f) = \frac{P_A}{V_{VG}} = \frac{Z_0}{A(1-\Gamma)} \times \left(e^{\frac{-j2\pi f L_A}{c}} + \Gamma e^{\frac{+j2\pi f L_A}{c}}\right)$$

Equation 3.1

where $Z_0 = \rho_0 c$, the acoustic characteristic impedance of air, and $\Gamma = \dfrac{Z_{IN} - \frac{Z_0}{A}}{Z_{IN} + \frac{Z_0}{A}}$, the reflection coefficient.

The effects of varying the distance between the glottis and the accelerometer, $\mathbf{L_A}$, are discussed further in Section 4.1.



Figures 3.7a and b. A pole-zero model of the subglottal acoustic input impedance as measured by Ishizaka, Matsudaira & Kaneko (1976). This impedance is used as a starting point for creating a model for each subject's subglottal transfer impedance.

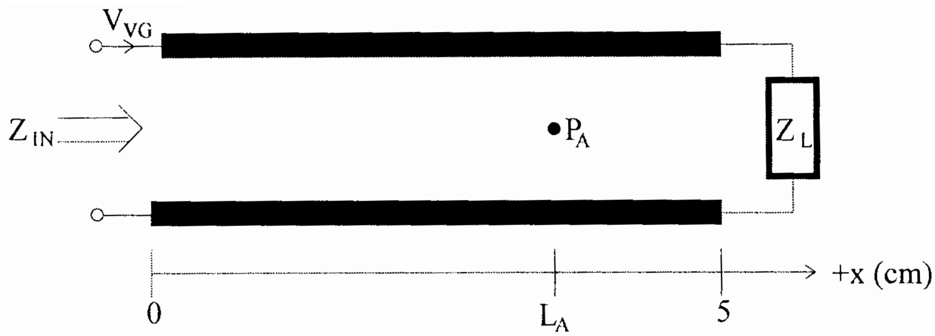


Figure 3.8. Subglottal transfer impedance model. The input impedance Z_{IN} initially equals the impedance shown in Figure 3.7, before poles & zeros are matched for a particular subject. A lossless acoustic transmission line connects the glottis at x = 0 cm to the load impedance Z_L at x = 5 cm. Knowing Z_{IN}, P_A can be calculated using Equation 3.1.

3.2.2.3 The glottal source, vocal tract and radiation characteristic

The glottal source uses the model described by Stevens (1998), mentioned in Section 2.2.2. The form of the model, two ideal volume velocity sources connected by a glottal impedance Z_G, allows complete glottal closure ($Z_G \rightarrow \infty$) or an arbitrary "glottal chink" size. If necessary for the theoretical prediction of the acceleration, the derivative of the waveform for the ideal sources $\frac{dv_{VG}(t)}{dt}$ can be produced from the LF-model by Fant, Liljencrants and Lin (1985).

The vocal tract transfer function **T(f)** is also treated as a "black box" like the subglottal transfer impedance. It is an all-pole system, with its poles being estimated from the measured acoustic pressure or airflow using Linear Predictive Coding (LPC). For a typical LPC scheme, see Rabiner & Juang (1993). The first three formant frequencies and bandwidths are adjustable while the upper formant frequencies are fixed, based on the vocal tract length as estimated from the average third formant frequency.

Lastly, the radiation characteristic *R(f)* is modeled as described by Stevens (1998) in Section 2.2.3. Since the radiation characteristic modeled by Fant and that by Stevens differ by a maximum of about 6 dB, the simpler approximation by Stevens (see Equations 2.2 and 2.3) is used for the model initially.

3.2.2.4 Estimating MFDR and SPL from the acceleration using the model

Using the vocal system model presented in Figure 3.6, an analysis framework was developed for the estimation of MFDR and SPL from the acceleration signal. To compare estimates of the MFDR from the acceleration and airflow signals, the model is simplified a bit since no radiation characteristic is needed. From the airflow signal, the vocal tract transfer function

T(f) is estimated with LPC. With **T(f)**, a digital filter is designed to inverse filter the effect of the vocal tract from the airflow time signal, giving the glottal flow time signal $\mathbf{v_{VG}(t)}$. Then a digital derivative filter is applied to $\mathbf{v_{VG}(t)}$, to produce the glottal flow derivative $\frac{dv_{VG}(t)}{dt}$, from which the MFDR is directly measured. From the acceleration signal, estimates of the subject's tracheal wall parameters $\mathbf{M_{TW}}$ and $\mathbf{R_{TW}}$, and subglottal transfer impedance formants and zeros are used to design digital filters to remove these effects from the acceleration. Since the tracheal wall model relates the velocity of the accelerometer to the tracheal pressure, giving the tracheal wall inverse filter the acceleration as input will produce the derivative of the tracheal pressure. Likewise, putting the derivative of the tracheal pressure through the digital inverse filter for the subglottal transfer impedance will give the glottal flow derivative, without the need for a derivative filter. This processing framework is shown in Figure 3.9.

For estimating SPL from the acceleration, the same steps as above for obtaining the glottal flow derivative from the acceleration are followed, and then the FFT of $\frac{dv_{VG}(t)}{dt}$ is calculated. Now in the frequency domain, the spectral magnitude of $\frac{dv_{VG}(t)}{dt}$ is multiplied by $\frac{G}{2\pi f}$, which is equivalent to integrating in the time domain, to obtain the glottal volume velocity Fourier Series spectral magnitude $|\mathbf{V_{VG}(f)}|$. Here the gain G compensates for the effects of windowing and taking the FFT of $\frac{dv_{VG}(t)}{dt}$, leaving the harmonic amplitudes of the spectrum as what they would be in the Fourier Series. The gain G equals $\frac{2}{nw_{DC}}$, where n is the FFT length in samples, and w_{DC} is the mean value in time of the windowing function used with the FFT (e.g., for a rectangular window, $w_{DC} = 1$; for a Hanning window, $w_{DC} = 0.5$). Then $|\mathbf{V_{VG}(f)}|$ is multiplied by the magnitude of the vocal tract transfer function $|\mathbf{T(f)}|$ and the magnitude of the radiation characteristic $|\mathbf{R(f)}|$ to obtain the Fourier Series spectral magnitude of the pressure at the microphone. The power of the harmonics of this spectrum are summed to give the SPL as calculated from the acceleration signal. This transformation of glottal flow derivative to microphone pressure is schematized in Figure 3.10.

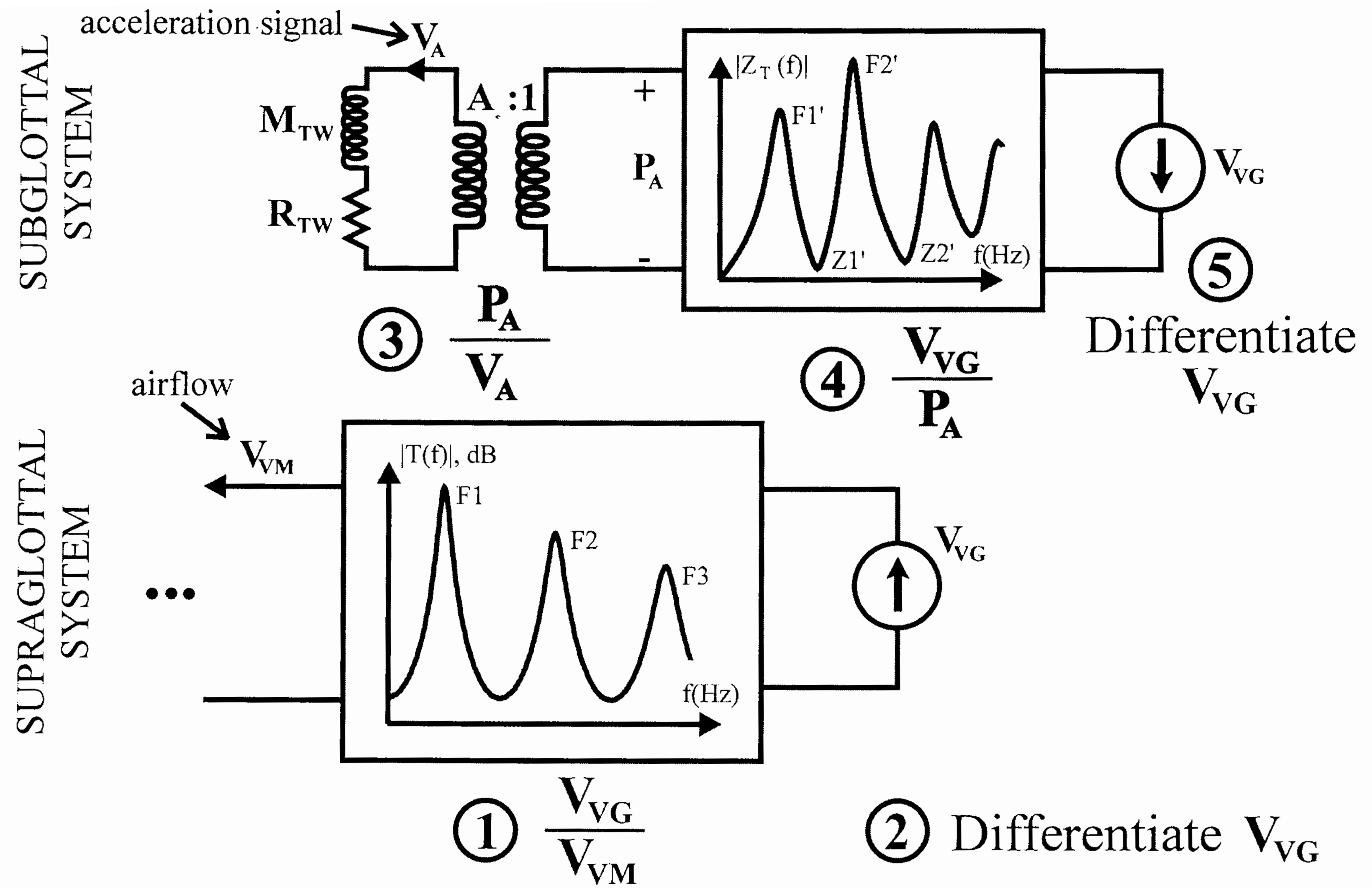


Figure 3.9. A step-by-step processing framework for estimating MFDR from the airflow and acceleration signals: 1) inverse filter airflow to estimate V_{VG}; 2) differentiate airflow-derived V_{VG}; 3) inverse filter acceleration signal to get P_A; 4) apply P_A to inverse of subglottal transfer impedance to get V_{VG}; 5) differentiate acceleration-derived V_{VG}.

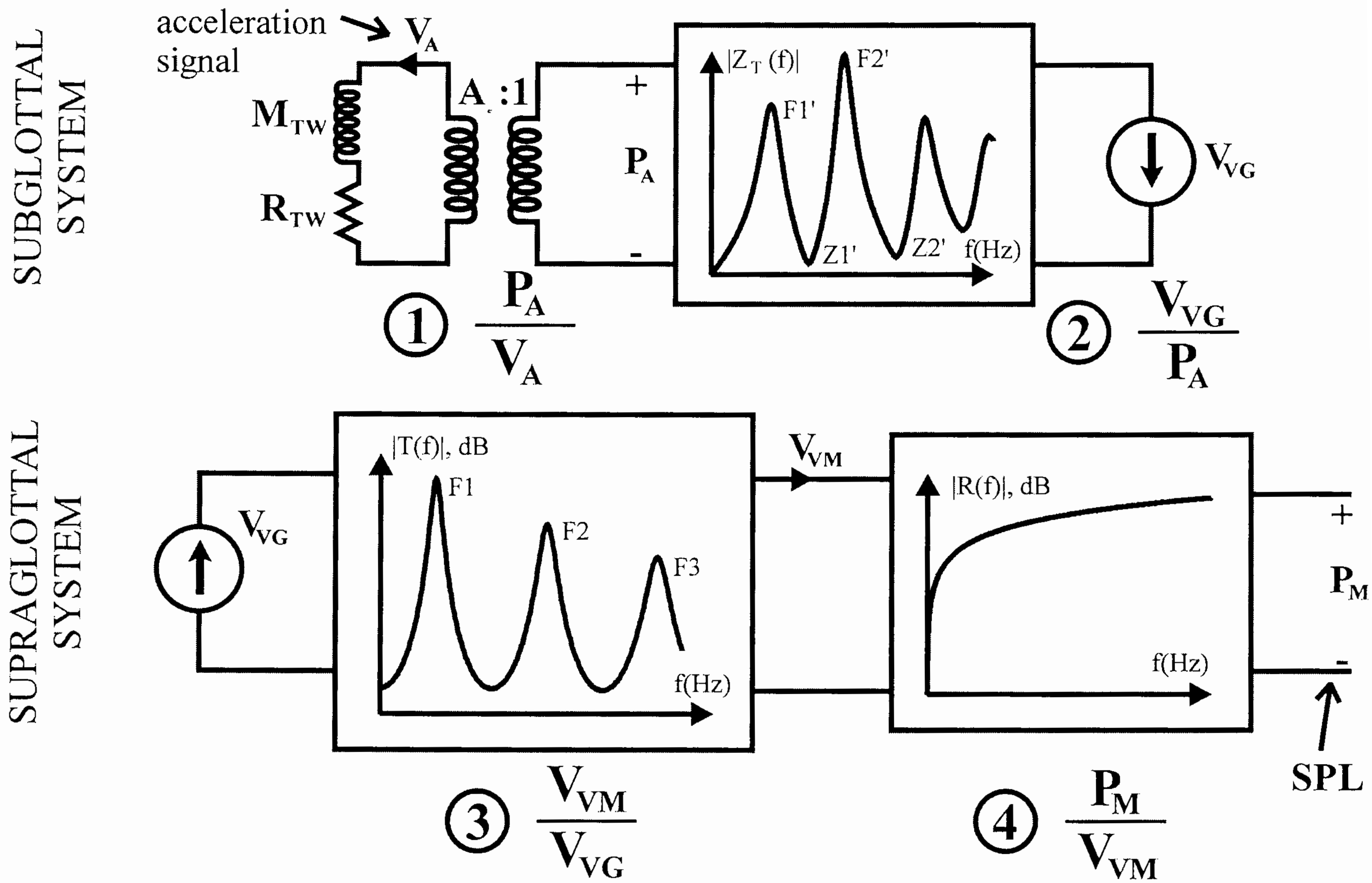


Figure 3.10. A step-by-step processing framework for estimating SPL from the acceleration signal: 1) inverse filter acceleration signal to get P_A ; 2) inverse filter effect of subglottal transfer impedance on P_A to get V_{VG} ; 3) apply vocal tract transfer function to V_{VG} to get V_{VM}; 4) apply radiation characteristic to V_{VM} to get P_M.

4 Experiment I: Verifying assumptions used in the vocal system model

The vocal system model developed for this work is presented and described in the previous chapter. Using the model to obtain estimates of MFDR and SPL requires two assumptions that have not been tested: 1) variation in the vertical distance between the glottis and accelerometer does not significantly affect the acceleration signal, and 2) variation in lung volume does not significantly affect the acceleration signal. Three of the vocal tasks described in Section 3.1 – Recordings are tailored to test these assumptions. This chapter describes the experimental methods, analyses, and results of testing these assumptions.

During speech, both the vertical larynx position (*VLP*) and the lung volume (*LV*) vary in time. The vocal system model presented in Section 3.2.2 contains the variable $\mathbf{L_A}$ to account for different VLP's across subjects, but we assume that the VLP will not vary significantly within a subject; or, equivalently, that its variation will not significantly affect the measured acceleration. If the VLP does vary significantly within a subject, then using the model to estimate vocal function parameters from the acceleration may become a very difficult problem, requiring knowledge of the VLP for a particular subject across the vowel and loudness continuum. Unlike VLP, there is no vocal system model parameter to account for changes in LV. Again we assume that changes LV do not significantly affect the measured acceleration. To investigate the effect of *VLP* on the measured acceleration, first a simple acoustic model of the subglottal system is used to predict what changes would occur in Section 4.1.1. Then simultaneous acceleration and acoustic data from this study are used to validate the model's predictions in Sections 4.1.2 and 4.1.3. The effect of *LV* on the subglottal formants and zeros was previously modeled by Ishizaka, Matsudaira and Kaneko (1976). Their findings are discussed briefly in Section 4.2 and compared to results from simultaneous acceleration and acoustic recordings from this study in Section 4.2.1.

4.1 The effect of Vertical Laryngeal Position (VLP) on the acceleration

4.1.1 Predictions using the subglottal transfer impedance model

A simple acoustic model of the subglottal system can be constructed with a lossless acoustic transmission line terminated with load impedance Z_L:

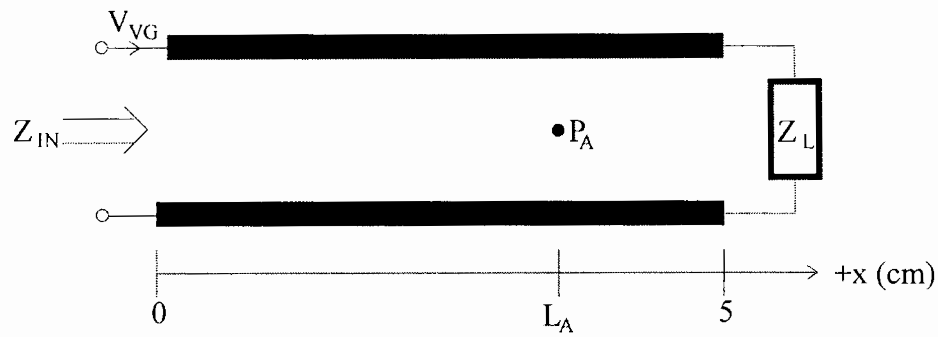


Figure 4.1. Subglottal system model originally presented in Figure 3.8, for estimating the influence of accelerometer-to-glottis distance L_A on the measured subglottal formants and zeros.

In Figure 4.1, the first 5 cm of the trachea below the glottis are represented by the acoustic transmission line, while the rest of the subglottal system is treated as the load impedance Z_L. The position of the accelerometer $\mathbf{L_A}$ on the neck is assumed to be between 0 and 5 cm below the glottis. Since the vocal system model described in Section 3.2.2 relates the measured acceleration to the pressure in the trachea deep to the accelerometer mounting ($\mathbf{P_A}$), the transfer impedance between the glottal volume velocity $\mathbf{V_{VG}}$ and the pressure $\mathbf{P_A}$ will describe how the subglottal formants and zeros measured from the acceleration signal may change with accelerometer position (see Equation 3.1).

The input impedance $\mathbf{Z_{IN}}$ is taken from the data of Ishizaka, Matsudaira, and Kaneko (1976), which is also shown in Figure 3.7 of Section 3.2.2.2. Varying the accelerometer position $\mathbf{L_A}$ from 2 to 4 cm in 0.5 cm steps results in the predicted transfer impedance $\mathbf{Z_T}$ changing as shown in Figure 4.2 below. The range 2 to 4 cm is chosen as an exaggeration of the measurements by Ewan and Krones (1974), who reported the range of mean larynx vertical displacements to be about 5mm across VCV utterances in English. Increasing their 5mm range by a factor of four makes spectral changes due to vertical laryngeal position more obvious over the frequency range 0 to 3000 Hz in the modeled $\mathbf{Z_T}$. The most significant effects of varying $\mathbf{L_A}$ on $\mathbf{Z_T}$ occur around **Z1′** and **Z2′**. Approximate changes in $\mathbf{f_{Z1'}}$ and $\mathbf{f_{Z2'}}$ of 100 Hz and 50 Hz (or 7.7% and 2.2%), respectively, and in $\mathbf{A_{Z1'}}$ and $\mathbf{A_{Z2'}}$ of 5 dB occur over the 2 cm to 4 cm range. The movement of **Z2′** upward in frequency, towards **F3′**, varies $\mathbf{A_{F3'}}$ by approximately 5 dB and $\mathbf{f_{F3'}}$ by approximately 75 Hz or 2.9%.

4.1.2 Varying VLP through changes in F0

As the first validation of the above model predictions, estimates of the first subglottal zero frequency are made from simultaneous acceleration and acoustic recordings during vocal tasks 5. This task is chosen to elicit laryngeal movement based on a study by Shipp (1975). Vocal task 5 – having the subject sustain the vowel /a/ while varying F0 from normal to lowest to highest (glissando) – may result in relative changes in VLP since the laryngeal mechanisms used to raise and lower pitch also may affect VLP. Shipp (1975) found that in six of six male subjects, changes in the VLP during a glissando corresponded to changes in fundamental frequency. However, the changes were not absolute, only relative to the starting VLP at the beginning of the glissando. In the two subjects whose entire VLP range could be visualized during the glissando, one had a range of 6mm and the other 14mm. From these ranges, the model predicts a change in $\mathbf{f_{Z1'}}$ of approximately 20 to 60 Hz. Because extremely high F0 values could result in too few harmonics in a spectrum to accurately estimate the subglottal zero frequencies, this analysis was restricted to the first octave of upward change in F0. Any changes in the first subglottal zero, chosen because its amplitude usually exceeds the noise floor and it potentially changes by about 100 Hz, are visualized by creating spectrograms of the first octave of vocal task 5. These spectrograms appear in Figures 4.3 through 4.12.

The spectrograms of the acceleration signal during vocal task 5 are produced using SpeechStation2 software by Sensimetrics. Each spectrogram is edited in time to display the start of the task through the first octave of increase in **F0**. Horizontal bands of white or less intense gray are the most obvious evidence of the first subglottal zero. Given the steeper spectral slope of the acceleration as compared to the acoustic signal and the reasons mentioned in the previous paragraph, the first subglottal zero **Z1′** offers the best visual cue as to whether the subglottal transfer impedance $\mathbf{Z_T(f)}$ changes with VLP as a result of changing **F0**. Harmonics that move towards $\mathbf{f_{Z1'}}$ due to the changing **F0** tend to fade in intensity or disappear altogether, suggesting again that $\mathbf{f_{Z1'}}$ remains stable across varying **F0**.

Figure 4.3 shows subject 1 performing vocal task 5, and note in this signal only there are 4 transients of unknown origin at roughly 275, 1500, 2225, and 2700 msec. None of the other recorded signals experienced these transients, the gain applied to the acceleration did not cause saturation, and the battery power to the accelerometer was sufficient. Also, these transients could not be reproduced in subsequent subject recordings. Aside from these transients, subject 1 shows a stable $\mathbf{f_{Z1'}}$ around 1100 Hz, appearing as a less intense band around **H5** from 0-900 msec, and beyond that as a white band. Subject 2 shows a stable $\mathbf{f_{Z1'}}$ in Figure 4.4 around 1150 Hz as a narrow white band around from 200-1300 msec and from 1600-3200 msec. Also, **H4** disappears as it passes through **Z1′** from 3350 to 3500 msec. Likewise, subject 3 shows a stable $\mathbf{f_{Z1'}}$ around 1100 Hz in Figure 4.5, with **H8** disappearing between 1100 and 1600 msec, and **H6** disappearing between 5700 and 5900 msec. Figure 4.6 displays subject 4 performing vocal task 5, with **Z1′** shown by a white band around 1150 Hz from 0 to 5.5 seconds, and then by **H5** disappearing from 5800 to 6500 msec and **H4** disappearing around 7 seconds. **Z1′** is difficult to find in the first second of the spectrogram of subject 5, shown in Figure 4.7, because the acceleration signal is more intense there and

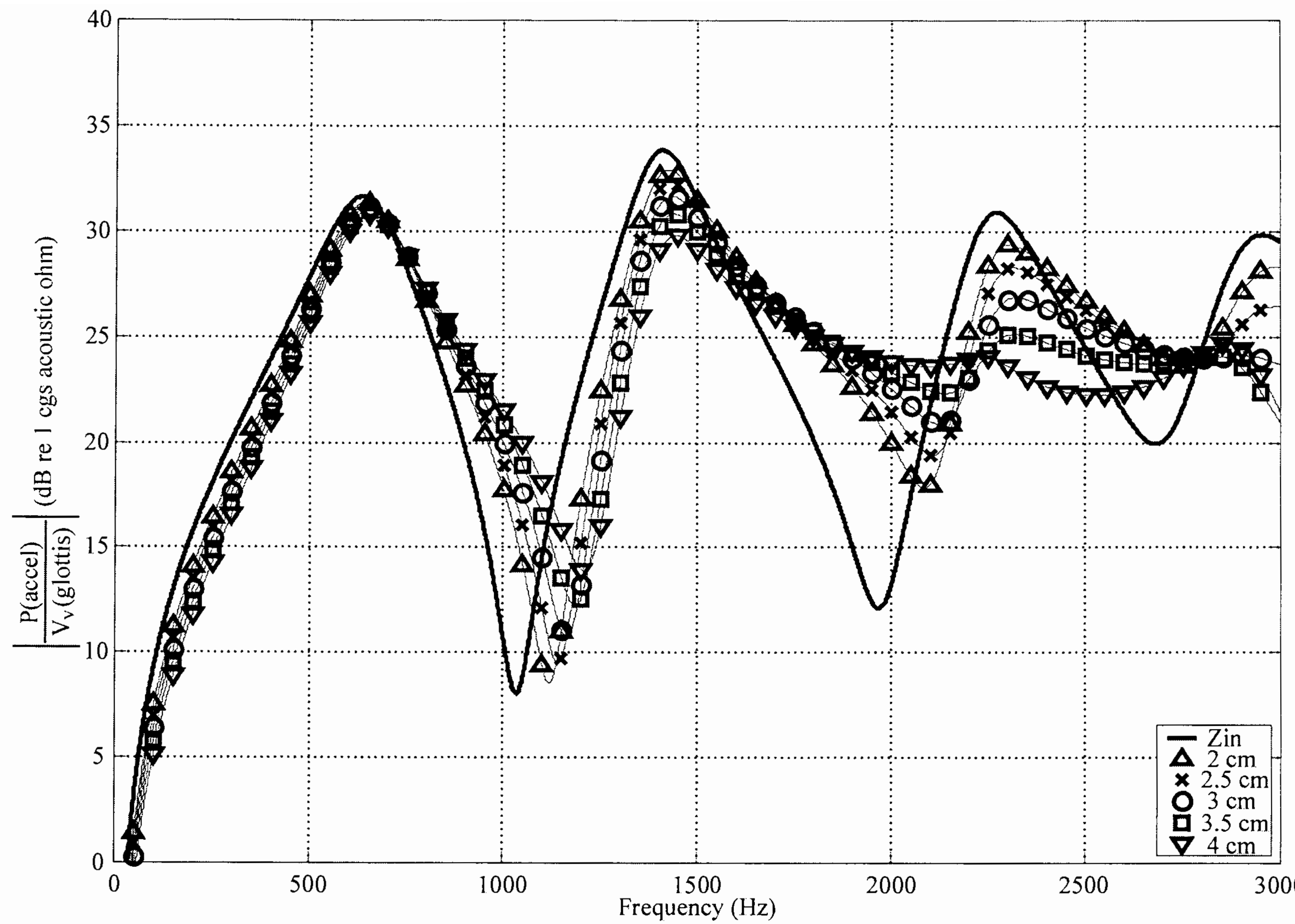


Figure 4.2. Changes in the subglottal transfer impedance as a function of accelerometer-to-glottis distance L_A. The solid line shows the condition L_A= 0 cm, which is equal to Z_{IN}. L_A varies in 5 mm steps from 2 to 4 cm below the glottis. Note the largest changes, of about 5 dB, in the subglottal transfer impedance occur around Z1' (~1100 Hz) and Z2' (~2100 Hz).

saturates the spectrogram. Around 1400 msec a $\mathbf{f_{Z1'}}$ of about 1200 Hz appears as a decrease in the intensity of **H7**, which fades completely by 2.5 seconds. Then between 5300 and 5700 msec **H5** disappears as well. Again in Figure 4.8, the **Z1′** of subject 6 is difficult to find because most of the energy above 1500 Hz is in the noise, and because the acceleration intensity is so decreased between 4 and 6 seconds. However, a wide white band around an $\mathbf{f_{Z1'}}$ of 950 Hz is evident between 0 and 4 seconds, and **H6** disappears between 8000 and 8500 msec, followed by **H4** disappearing between 9800 and 10100 msec. Subject 7 shows the most obvious **Z1′** in Figure 4.9, around 900 Hz with a narrow white band that extends from 0 to 6.5 seconds. In contrast, subject 8 shows a white band of $\mathbf{f_{Z1'}} \cong 975$ Hz only from 0 to 1.5 seconds in Figure 4.10, and then **H5** disappears between 5100 and 5700 msec, followed by **H4** disappearing between 6200 and 6600 msec. Figure 4.11 shows subject 9 has an $\mathbf{f_{Z1'}}$ of about 1300 Hz, with **H5** and **H6** being less intense than surrounding harmonics from 0 to 2 seconds, followed by **H4** disappearing at 4250 msec and **H3** disappearing at 5450 msec. Lastly, Figure 4.12 shows a stable $\mathbf{f_{Z1'}} \cong 1050$ Hz for subject 10 as a narrow white band that extends almost throughout the task, with **H6** disappearing between 4100 and 4300 msec.

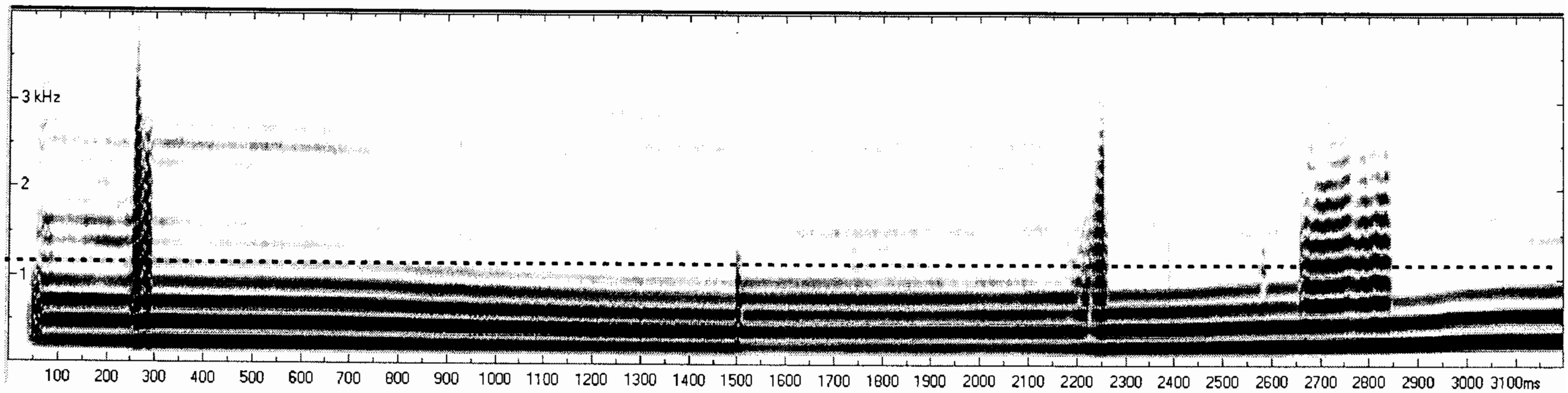


Figure 4.3. Spectrogram of the acceleration signal for the sustained vowel /a/ with varying pitch, subject 1. The dashed line shows the location of Z1'. This subject showed unusual transient noise in the acceleration signal near 275, 1500, 2225, and 2700 msec.

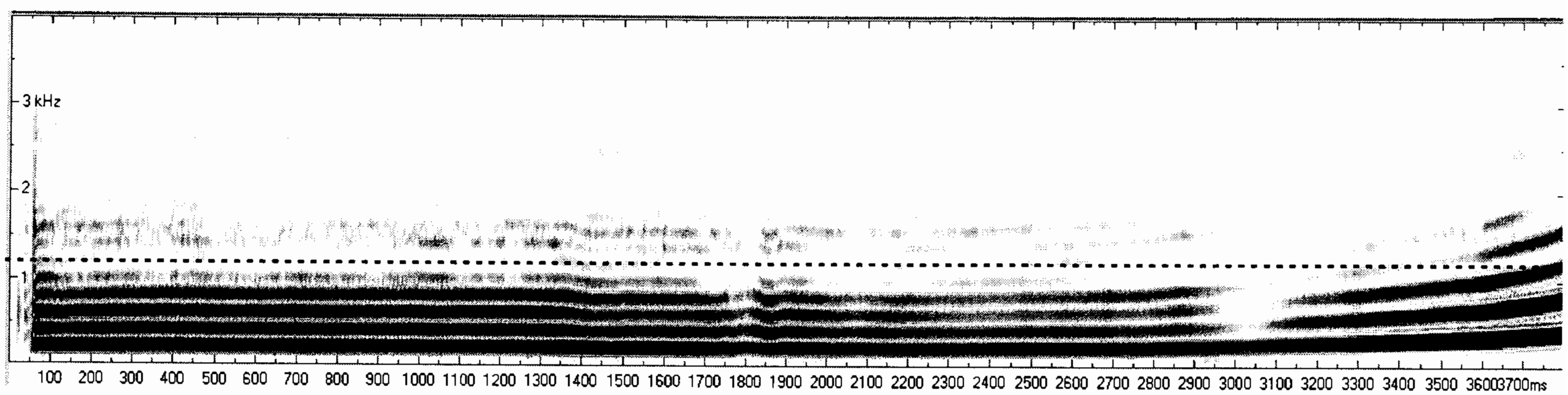


Figure 4.4. Spectrogram of the acceleration signal for the sustained vowel /a/ with varying pitch, subject 2. The dashed line shows the location of the white band corresponding to Z1' around 1150 Hz. Nnote its stability throughout the task and how the fourth harmonic disappears as it passes through Z1' between 3350-3500 msec.

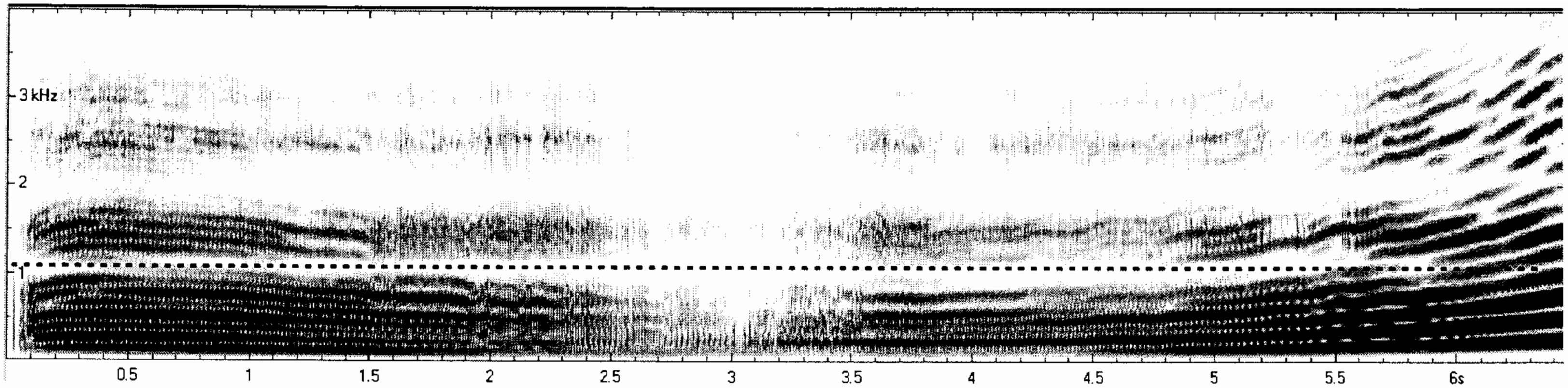


Figure 4.5. Spectrogram of the acceleration signal for the sustained vowel /a/ with varying pitch, subject 3. The dashed line shows the location of the white band corresponding to Z1' around 1100 Hz. Note how the eighth harmonic disappears from 1100-1600 msec as it goes through Z1', and also the sixth harmonic disappears from 5700-5900 msec.

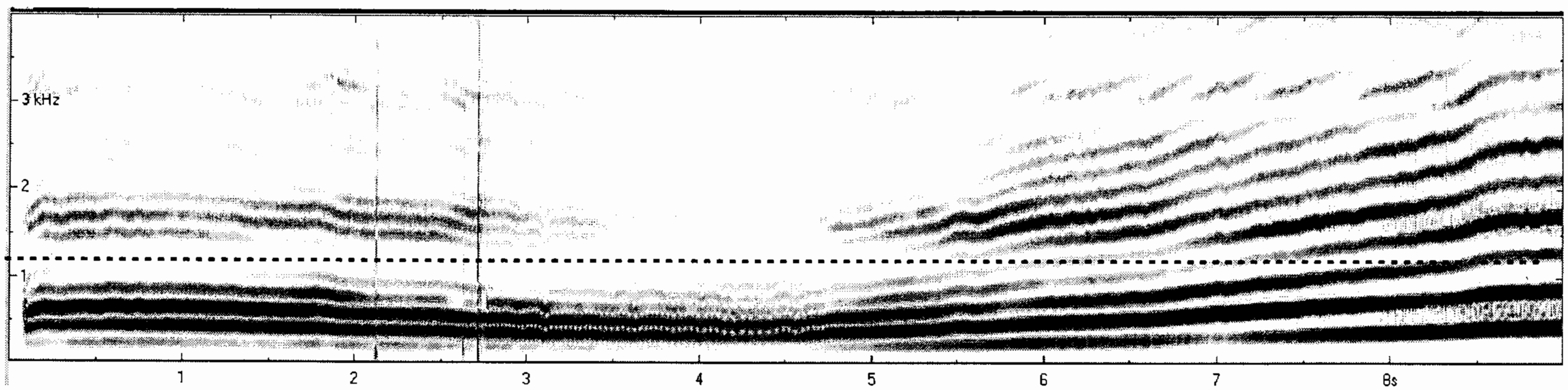


Figure 4.6. Spectrogram of the acceleration signal for the sustained vowel /a/ with varying pitch, subject 4. The dashed line shows the location of the white band corresponding to Z1' around 1150 Hz up to 5.5 seconds. Note how the fifth harmonic disappears as it passes through Z1' between 5800 and 6500 msec, then how the fourth harmonic disappears around 7 seconds.

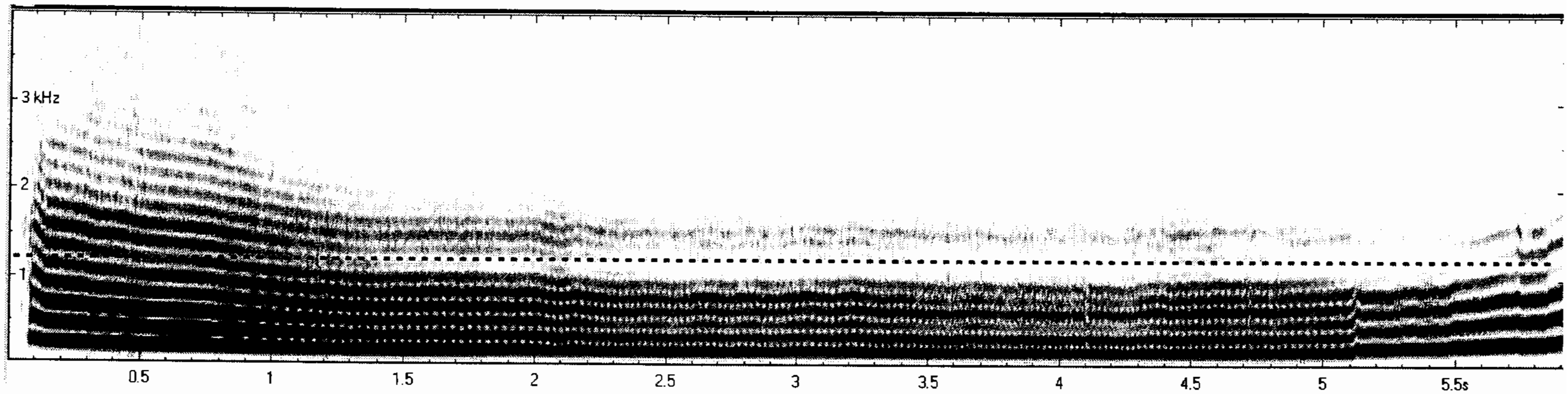


Figure 4.7. Spectrogram of the acceleration signal for the sustained vowel /a/ with varying pitch, subject 5. The dashed line shows the location of the white band corresponding to Z1' around 1200 Hz. The intensity of the acceleration before 1400 msec makes identifying Z1' difficult. Note how the fifth harmonic disappears between 5300 and 5700 msec.

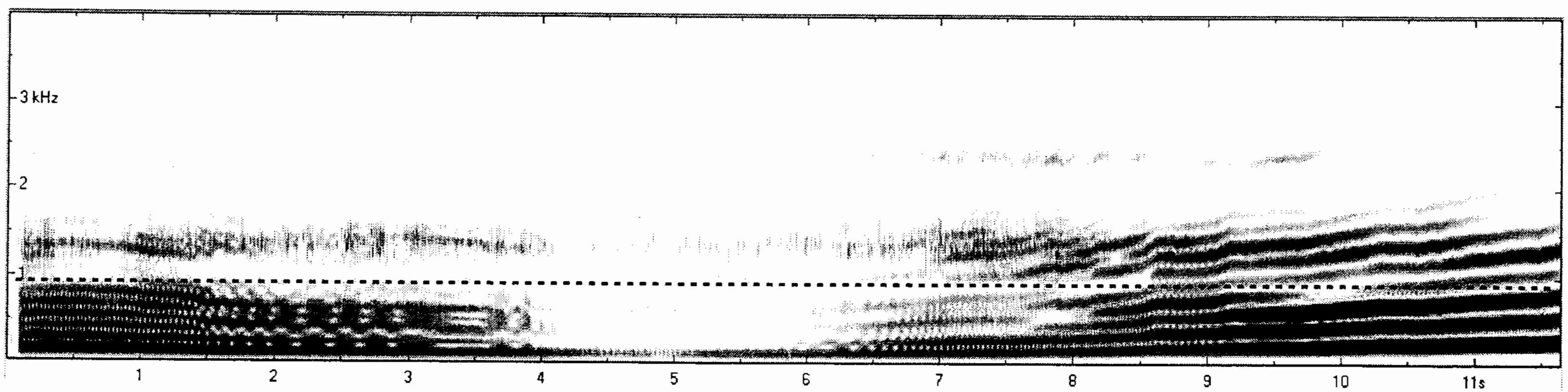


Figure 4.8. Spectrogram of the acceleration signal for the sustained vowel /a/ with varying pitch, subject 6. The dashed line shows the location of Z1' around 950 Hz. For this subject, Z1' is difficult to find because most of the acceleration energy above 1500 Hz is in the noise. However, note how the sixth harmonic disappears from 8000-8500 Hz, then the fourth harmonic disappears from 9800-10100 msec.

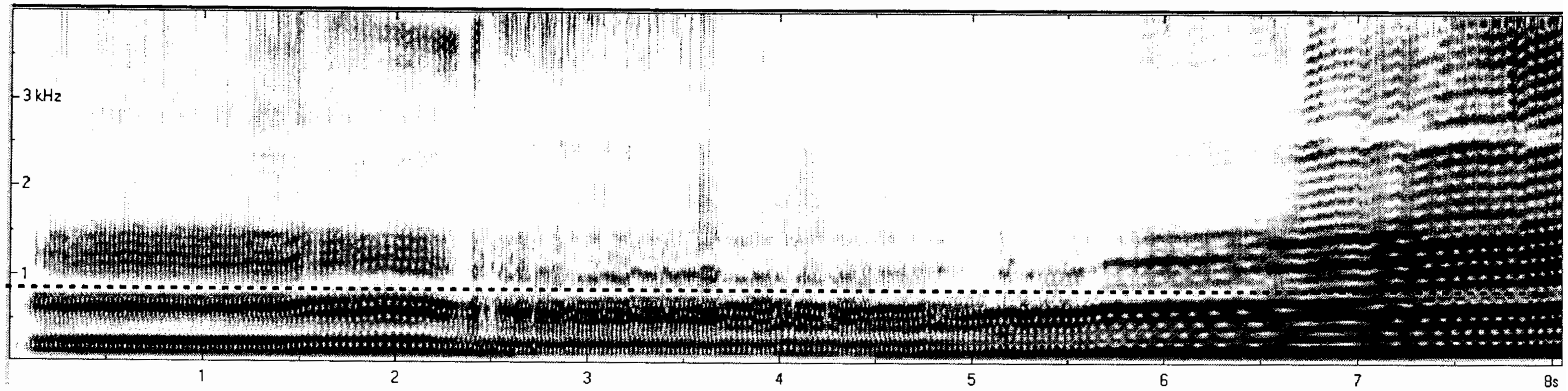


Figure 4.9. Spectrogram of the acceleration signal for the sustained vowel /a/ with varying pitch, subject 7. The dashed line shows the location of the white band corresponding to Z1' around 900 Hz.

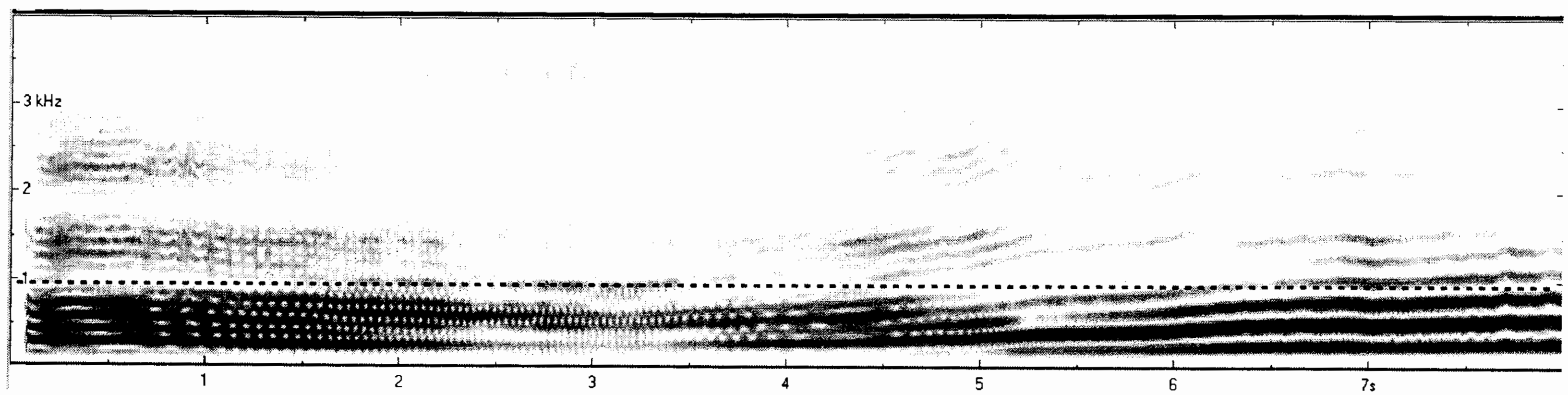


Figure 4.10. Spectrogram of the acceleration signal for the sustained vowel /a/ with varying pitch, subject 8. The dashed line shows the location of Z1'. Note that for this subject, the white band corresponding to Z1' only shows between 0 and 1.5 seconds, but the fifth harmonic fades as it passes through Z1' from 5100 to 5700 msec, followed by the fourth harmonic disappearing between 6200 and 6600 Msec.

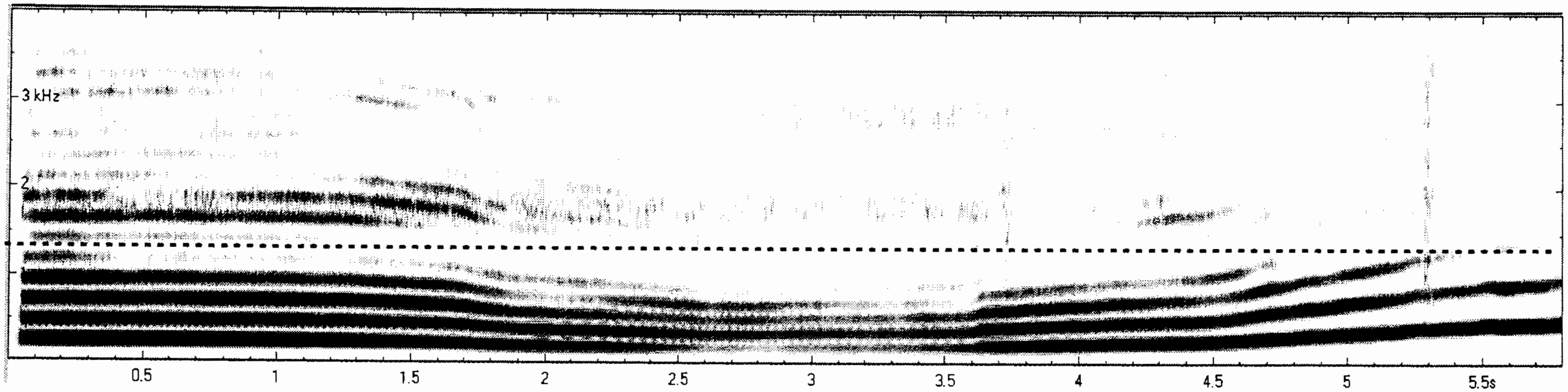


Figure 4.11. Spectrogram of the acceleration signal for the sustained vowel /a/ with varying pitch, subject 9. The dashed line shows the location of Z1' around 1300 Hz. For this subject, Z1' first appears as the fifth and sixth harmonics being less intense than the surrounding harmonics from 0-2 seconds. Later in the task, the fourth harmonic disappears at 4250 msec and the third harmonic disappears at 5450 msec.

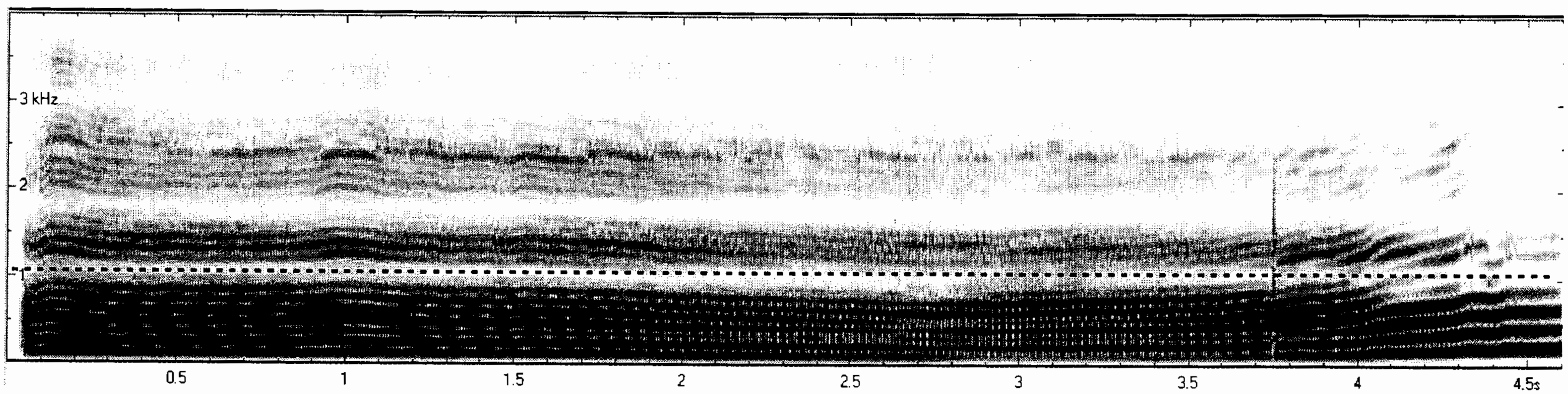


Figure 4.12. Spectrogram of the acceleration signal for the sustained vowel /a/ with varying pitch, subject 10. The dashed line shows the location of the white band corresponding to Z1' around 1050 Hz. Note that this subject shows a white band for Z1' that extends throughout the task, and the sixth harmonic fades between 4100 and 4300 msec.

4.1.3 Varying VLP by vocal task 6 – "Pat gave the dog a bag"

This task is chosen to elicit laryngeal movement based on a previous study by Ewan and Krones (1974). Vocal task 6 – "Pat gave the dog a bag" – should produce a decrease in VLP from the /ae/ in "Pat" to the /ae/ in "bag", based on their results. They reported that the VLP for voiced consonants is 4 to 6 mm lower than for voiceless consonants, and that difference in VLP is maintained, although it may be lessened, through the vowels adjacent to the voiced or voiceless consonants. Thus the VLP in the /ae/ in "bag" should be lower than the VLP for the /ae/ in "Pat" by about 5mm. Consequently a change in the frequencies of the first and second subglottal zeros of about 25 Hz should occur, according to the model prediction shown in Figure 4.2.

For the analysis of vocal task 6, four wideband spectra of the acceleration signal are taken from consecutive periods in the middle of the vowel /ae/ from the word "Pat" and the word "bag". The MATLAB® program lplv (see Appendix B) is used for this analysis, and an example of its analysis display is shown in Figure 4.13. To create the wideband spectra, the FFT is performed on either a 64-point (3.2 msec, for female voice) or 128-point (6.4 msec, for male voice) Hanning window that is applied to the acceleration signal. The window was placed such that its start aligned with the positive-going excitation pulse of the acceleration signal, and its center encompassed the "ringing" or response of the subglottal system to that excitation. The magnitude of the spectrum, multiplied by $2\pi f$ to decrease the spectral tilt and emphasize the peaks and zeros, is plotted below the acceleration time signal. The user then selects the frequencies of the first two subglottal formants ($\mathbf{f_{F1'}}$, $\mathbf{f_{F2'}}$) and the first two subglottal zeros ($\mathbf{f_{Z1'}}$, $\mathbf{f_{Z2'}}$) with the cursor. The program lplv then locates the actual maximum or minimum in the spectrum within a 7-point (68 Hz) window centered on the cursor-selected point, and displays the frequency and magnitude of that maximum or minimum. Along with the acceleration signal results, the average value of the differential EGG signal for the 64-point window is displayed for relative VLP comparison. Results from this analysis are shown in Figures 4.14 and 4.15.

Figures 4.14 and 4.15 plot the frequencies of the first subglottal zero ($\mathbf{f_{Z1'}}$) versus the relative vertical laryngeal position (VLP) for the vowel /ae/ from the word "Pat" and the word "bag" in vocal task 6. Again, **Z1′** is chosen as the subglottal transfer impedance parameter of interest because it exhibits relatively large changes as VLP varies. Figure 4.14 shows the female data and Figure 4.15 shows the male data. The open symbols denote the mean $\mathbf{f_{Z1'}}$ and EGG voltage ($\mathbf{V_{EGG}}$) for the word "Pat", and the closed symbols denote the same for the word "bag". The bars extending horizontally and vertically represent the range of values found for $\mathbf{f_{Z1'}}$ and $\mathbf{V_{EGG}}$ respectively.

For the female data shown in Figure 4.14, all five of the subjects (S1, S2, S4, S5, S9) exhibit the expected decrease in the mean relative VLP from "Pat" to "bag". And four out of the five (S1, S2, S4, S5) also show the corresponding increase in $\mathbf{f_{Z1'}}$ predicted by the vocal system model in Section 4.1.1. In contrast, Figure 4.15 shows that only two of the five males (S7, S8) exhibit the expected decrease in mean relative VLP and corresponding increase in $\mathbf{f_{Z1'}}$. Table 4.1 lists the mean change in $\mathbf{f_{Z1'}}$ (percent), the direction of mean VLP change (up ↑ or

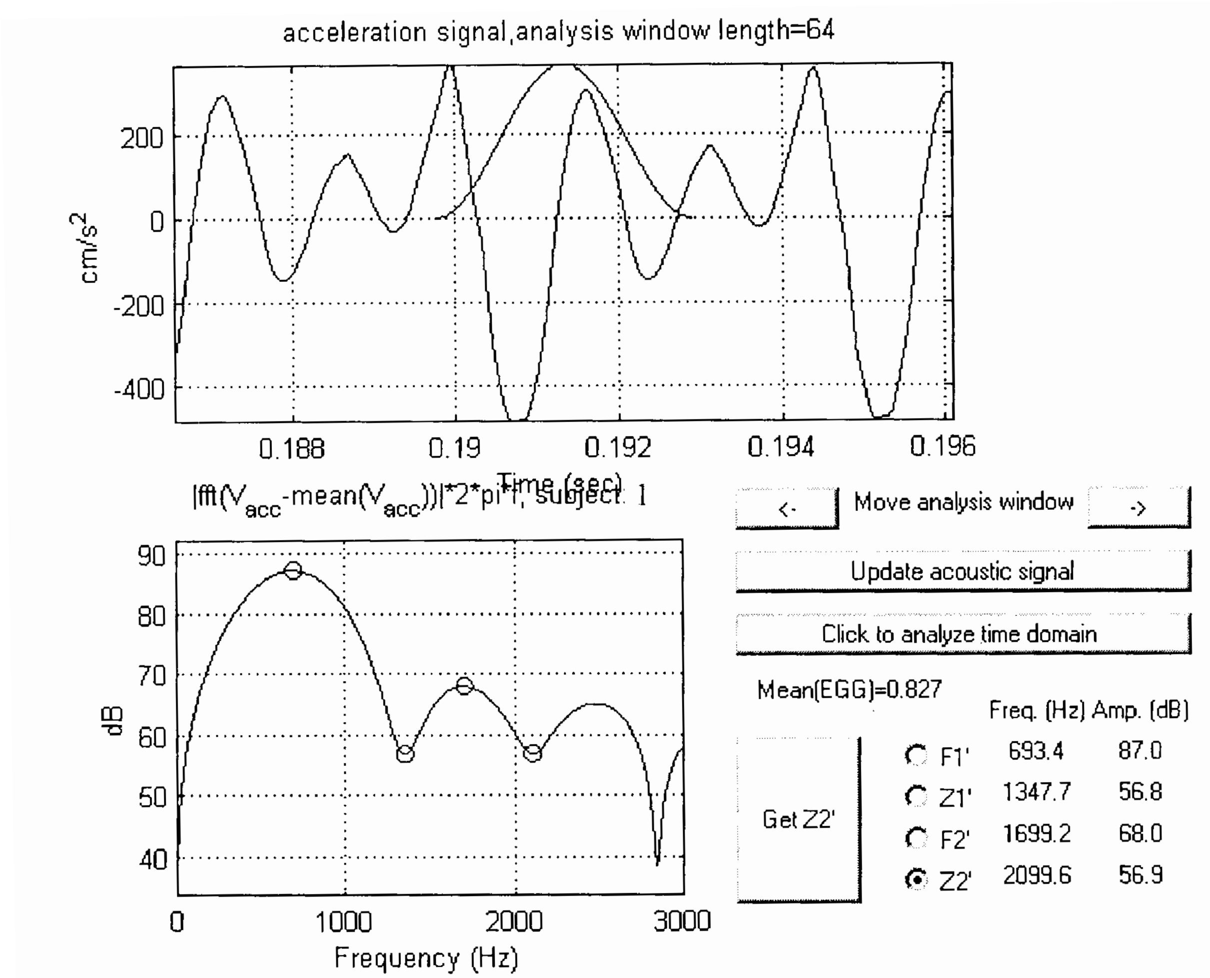


Figure 4.13. An example of the spectral analysis performed on the vowel /ae/ from the words "Pat" or "bag" in vocal task 6 – "Pat gave the dog a bag". This example is from the word "Pat". The top plot shows about two periods of the raw acceleration signal, along with the Hanning window used to create the spectrum shown in the lower left plot. Note that the Hanning window starts at the center of the positive-going excitation, and is centered on the "ringing" of the subglottal system following the excitation. The first two subglottal poles and zeros are identified in the spectrum with circles, and their frequencies are recorded in the lower right.

down ↓), and the predicted change in the magnitude of the subglottal transfer impedance around **Z1′** (dB) from the vocal system model.

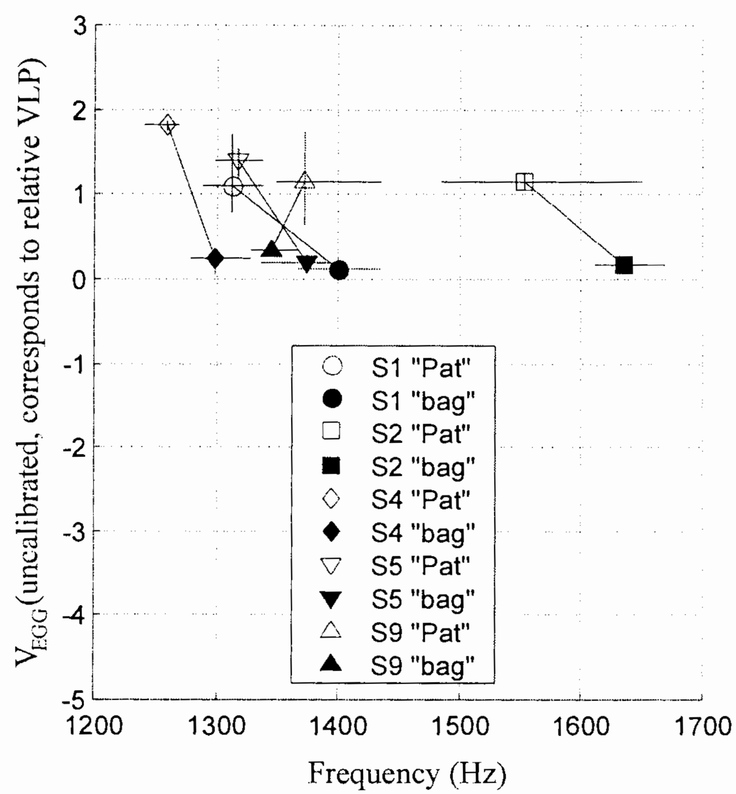
Overall, these results suggest that VLP affects the subglottal transfer impedance similarly to the predictions of the vocal system model in Section 4.1.1. Changes in $|\mathbf{Z_T(f_{Z1'})}|$ on the order of 5 dB will not significantly affect estimates of MFDR or SPL, which rely mostly on the harmonics below 1000 Hz.

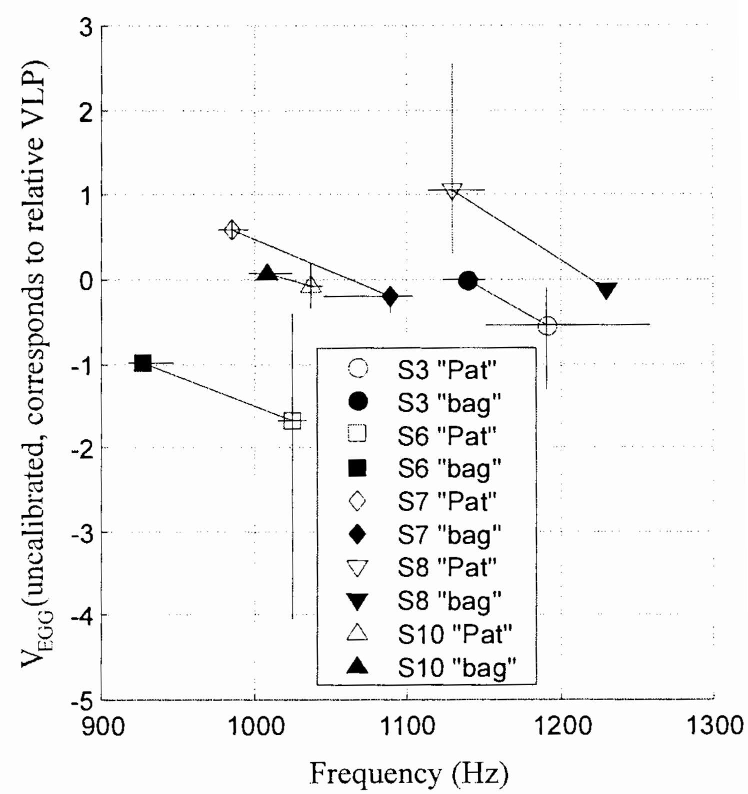
Subject	$\frac{f_{Z1'}(Pat) - f_{Z1'}(bag)}{f_{Z1'}(Pat)} \times 100\%$	Direction of mean VLP change	Predicted change in $\lvert Z_T(f_{Z1'}) \rvert$ (dB)
1	-6.5 %	↓	+5.4
2	-5.3 %	↓	+5.3
3	4.3 %	↑	-2.6
4	-3.1 %	↓	+1.8
5	-4.2 %	↓	+2.7
6	9.5 %	↑	-5.8
7	-10.4 %	↓	+6.6
8	-8.9 %	↓	+6.3
9	-2.0 %	↓	+1.1
10	2.8 %	↑	-1.3

Table 4.1. Percent change in mean frequencies of the first subglottal zero, the direction of VLP motion, and the predicted change in $|Z_T(f_{Z1'})|$ from the word "Pat" to the word "bag".

4.1.4 Discussion of the effect of VLP on the acceleration

Concerns about varying VLP relative to the accelerometer position were initially evaluated through the model of the subglottal transfer impedance $\mathbf{Z_T(f)}$. Results of that analysis showed that even with extreme changes in VLP, the peaks in the spectrum of $\mathbf{Z_T(f)}$ remained essentially unchanged up to 2000 Hz, and the largest changes in the zeros of the spectrum would be on the order of 5 dB. These predictions are further supported by the subject data shown in Figures 4.3 through 4.12, 4.14 and 4.15, both qualitatively and quantitatively. Qualitatively, the spectrograms of the acceleration signal from vocal task 5 showed stable **Z1′** frequencies as horizontal white bands that diminished any harmonics that moved through them. Quantitatively, the changes in $\mathbf{f_{Z1'}}$ between the vowel /ae/ in "Pat" and in "bag" from vocal task 6 were on the order of the changes predicted by the model. The maximum difference in $\mathbf{f_{Z1'}}$ between these two words was about 100 Hz, corresponding to a change in the spectral magnitude around **Z1′** of 6.6 dB, which is just 1.6 dB more than the model's prediction. Furthermore, since the estimates of MFDR, SPL, and $\mathbf{BW_{F1'}}$ rely heavily on the peaks in the glottal spectrum as estimated from the acceleration, as is seen in Sections 5, 6 and 7, a change in $\mathbf{f_{Z1'}}$ of 100 Hz will not dramatically affect these estimates. However, the estimates of $\mathbf{f_{Z1'}}$ obtained from these two methods are not always consistent within a subject. For example, subject 2 shows $\mathbf{f_{Z1'}} \approx$ 1150 Hz from the spectrogram in Figure 4.4 but $\mathbf{f_{Z1'}} \approx$ 1600 Hz from vocal task 6. These inconsistencies suggest that the short window used to analyze vocal task 6 may lead to errors in estimating $\mathbf{f_{Z1'}}$, and that an approach like the one described in the next Section may be more appropriate for estimating $\mathbf{f_{Z1'}}$ from a spectrum.





Figures 4.14 (females, left) and 4.15 (males, right). Frequencies of the first subglottal zero versus the relative vertical larynx position (VLP) for the vowel /ae/ in "Pat"(open symbols) and "bag"(closed symbols) from vocal task 6, for all subjects S1 through S10. The symbols show the mean values of the first subglottal zero frequency and the relative VLP, while the horizontal and vertical bars show the range of values for those measures. Note that all of the females show the expected decrease in relative VLP from "Pat" to "bag", but only 2 of the 5 males show that decrease.

4.2 The effect of Lung Volume (LV) on the acceleration

Lung volume (LV) can significantly influence the subglottal formants and zeros, as was shown with a model by Ishizaka, Matsudaira and Kaneko (1976). They initially measured the acoustic input impedance of the subglottal system in Japanese subjects with tracheostomas. Then they created a model of the subglottal system based on Western anatomical data by Weibel (1963), and their model initially predicted subglottal formant frequencies that were lower than they had measured (e.g., $\mathbf{f_{F1'}}$ was modeled as 615 Hz but measured as 640 Hz). They noted that other literature suggests that the average ratio of tracheal lengths between Japanese and Western anatomies is 0.941. By multiplying the lengths of the airways given by Weibel by 0.941 their modeled subglottal formant frequencies better matched their data. From their modeling effort it can be inferred that if LV significantly varies within an individual across an utterance and LV affects tracheal and bronchial lengths, then the subglottal formants and zeros may vary in frequency and influence the acceleration signal in a time-varying manner.

4.2.1 Varying LV by vocal task 4 – Maximum Phonation Duration

To test whether LV significantly affects the subglottal formants and zeros, vocal task 4 – the maximum phonation duration (MPD) - is analyzed. The MATLAB® program lplvlong was used with this analysis, a variation of the program lplv used above. A typical analysis screen from this task is shown in Figure 4.16. First the start and end points of the MPD task are located. The starting point (t_I) is defined as the seventh period of phonation from the onset of phonation. The end point (t_F) is defined as the last regular period of phonation. In this case, "regular" means that there could be no breaks in phonation before t_F, and visually the acceleration signal had to be above the noise so that a reasonable spectrum could be obtained. After locating the start (t_I) and end (t_F) points of the MPD task, six additional points spaced at $(t_F - t_I)/7$ seconds are selected. For each time point, an analysis of the spectral magnitude similar to that for vocal task 6 (see Section 4.1.3) is carried out. The main differences between these two analyses are 1) the window used to produce the spectra encompassed 4-5 periods by being 512 points (25.6 msec) for females and 1024 (51.2 msec) for males, and 2) the acceleration spectral magnitudes are smoothed with a 41-point (400 Hz) window to minimize the movement of the subglottal formant/zero frequencies with changing fundamental frequency. For the results shown below in Figures 4.17 and 4.18, the first subglottal zero frequency is plotted versus time, where time was normalized to the final time value for each subject, such that every subject's time axis ends at t=1. The first subglottal zero is again chosen here because it is usually above the acceleration noise floor, and because it is the salient feature chosen for the VLP investigation.

Figures 4.17 and 4.18 show the results from tracking $f_{Z1'}$ over the duration of vocal task 4. As with vocal task 6, female data ~~is~~ are shown in Figure 4.17 on the left, and male data ~~is~~ are shown in Figure 4.18 on the right. Each time point was obtained as described above. The first and last points (T_1 and T_8) are excluded to reduce the influence of "end effects", that is, unusually large physiological and acoustic changes that may occur at either lung volume extreme

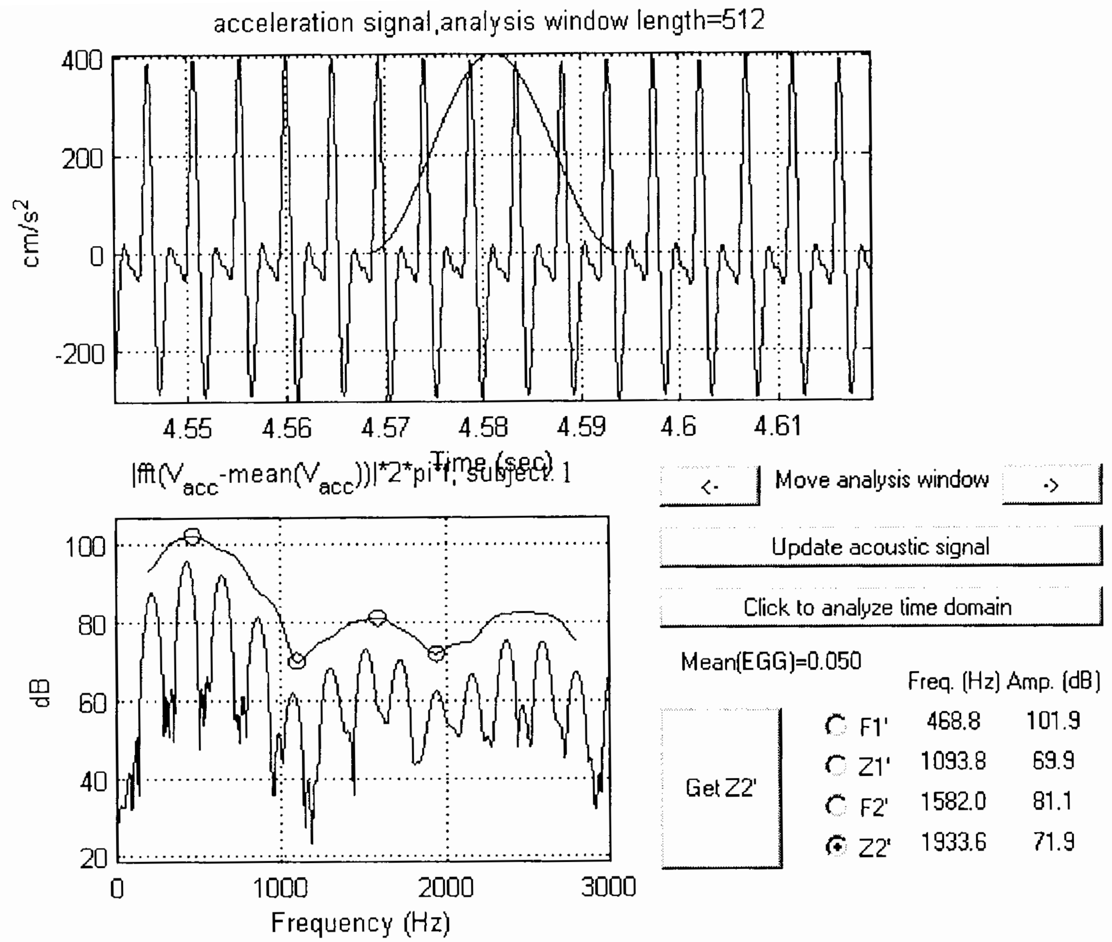
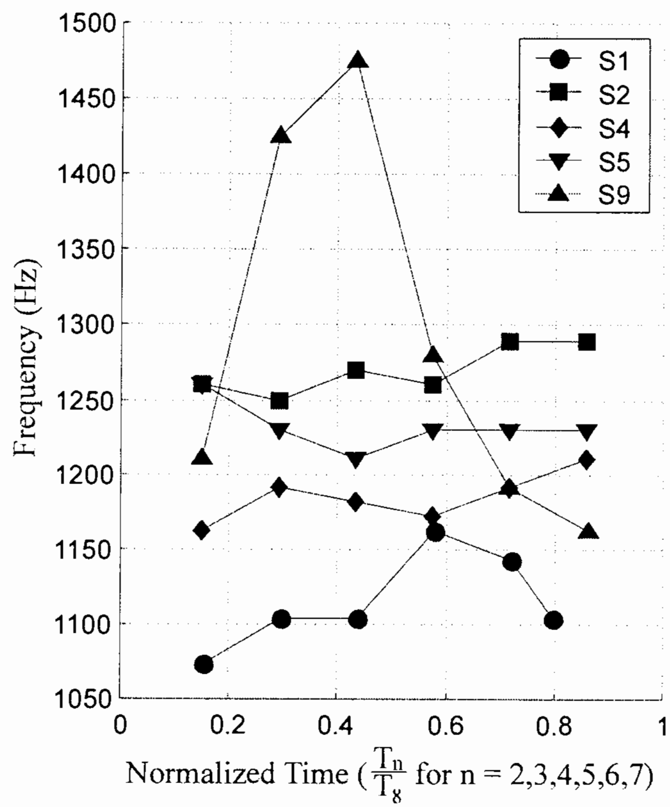
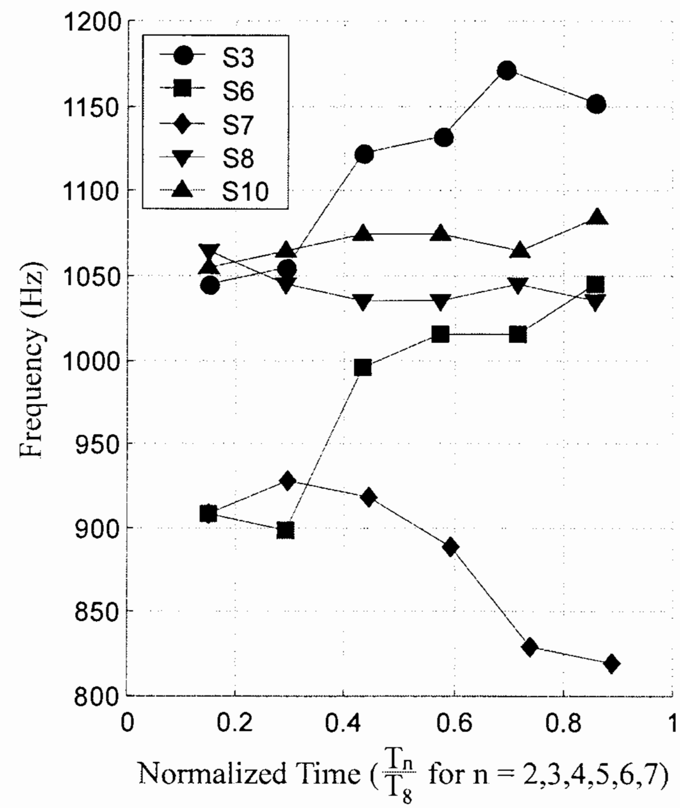


Figure 4.16. Spectral analysis screen example for vocal task 4 – maximum phonation duration. The upper plot shows the raw acceleration signal for the sustained vowel /a/ and the Hanning window used to create the uncalibrated, pre-emphasized acceleration spectrum shown in the lower plot. The curve above the acceleration spectrum is a 41-point (400 Hz) smoothed version of the spectrum, and the circles denote the frequencies of the first two subglottal poles and zeros, which are listed in the lower right.

(Zenker, 1964). Given the results of the model by Ishizaka, Matsudaira & Kaneko (1976) discussed above, $\mathbf{f_{Z1'}}$ is expected to increase steadily as lung volume decreases throughout the maximum phonation duration.

For the females, Figure 4.17 shows that only two subjects (S2 and S4) agree with this prediction. For those two subjects as well as S5, $\mathbf{f_{Z1'}}$ appears to remain relatively constant across this task, changing less than 50 Hz. From the vocal system model, this corresponds to a change in $|\mathbf{Z_T(f_{Z1'})}|$ of less than 2.5 dB, which likely has little effect on the acceleration signal. Subjects S1 and S9 show larger yet inconsistent changes with LV. In particular, subject S9 displays changes greater than 250 Hz, which appear to be due to her acceleration





Figures 4.17 (females, left) and 4.18 (males, right). Frequency of the first subglottal zero versus normalized time over the course of vocal task 4 - maximum phonation duration - for all subjects S1 through S10. Note the first and last time points have been omitted to avoid end effects (see text). Note that in general, none of the subjects exhibit a consistent, gradual increase in the frequency of the first subglottal zero with time (equivalent to decreasing lung volume).

signal being so small around T_3 and T_4 that the valley normally present due to **Z1′** was filled in by the noise floor.

Three of the males (S3, S6 and S10) show an increase in $\mathbf{f_{Z1'}}$ over time in Figure 4.18, although only one of them (S10) shows the expected small (i.e., less than 50 Hz), gradual increase. The sharp increase of $\mathbf{f_{Z1'}}$ in subjects S3 and S6 around time T_4 may be a result of increased glottal coupling to the vocal tract leading to a boost in the acceleration spectrum near **F2**, which was 1180 Hz for S3 and 1100 Hz for S6. Subject S8 shows relatively little change in $\mathbf{f_{Z1'}}$ over this task – less than 30 Hz. And subject S7's $\mathbf{f_{Z1'}}$ drops more than 100 Hz from T_3 to T_7, again possibly due to increased glottal coupling with a second vocal tract formant frequency of 1050 Hz.

Overall, these results suggest that when glottal coupling does not appear to obscure the location of **Z1′**, changes in LV over the range of a subject's entire lung capacity minimally affect the subglottal transfer impedance. Specifically, subjects S2, S4, S5, S8 and S10 show changes in $\mathbf{f_{Z1'}}$ on the order of 50 Hz or less, which as noted above corresponds to a change in $|\mathbf{Z_T}(\mathbf{f_{Z1'}})|$ of less than 2.5 dB. Such a small change in **Z1′** has a negligible effect on surrounding harmonics in the acceleration spectrum.

4.2.2 Discussion of the effect of LV on the acceleration

Assessing the effect of LV on $\mathbf{Z_T(f)}$ is not as straightforward as the study of VLP. As mentioned above, errors in $\mathbf{f_{Z1'}}$ could be made due to coupling between the supra- and subglottal systems introducing peaks near **F2** in the acceleration spectrum. Even with these errors, no consistent change in $\mathbf{f_{Z1'}}$ with LV across subjects is apparent. Within subjects, when $\mathbf{f_{Z1'}}$ seems to change consistently with LV (e.g., see subject 10 in Figure 4.18), it changes by about 50 Hz, which again does not significantly affect the acceleration-derived measures to be explored in Sections 5, 6 and 7. In addition, the maximum phonation duration task likely exaggerates any effects that LV may have on $\mathbf{Z_T(f)}$, since for conversational speech people generally start at 50-60% of their lung's vital capacity, and only use 10-20% of the vital capacity overall (Hixon, 1973).

5 Experiment II: Estimating MFDR from the acceleration using the vocal system model

Chapters 1 through 4 provide the introduction, motivation, experimental methods, vocal system model, and tests of the model assumptions for the goal of estimating the vocal function parameters MFDR and SPL from the acceleration signal. In this chapter, the vocal system model is used to estimate MFDR from the acceleration, and that estimate is compared to the conventional measure of MFDR from the airflow signal.

5.1 Using the model to estimate MFDR from the acceleration and airflow signals

A series of representative images from the MATLAB MFDR analysis program accflo3 appear in Figures 5.1a through 5.1e, and are intended to demonstrate the sequence of signal processing performed on the acceleration and airflow signals. The signal processing is tailored after work by Perkell, Holmberg & Hillman (1991) and Rothenberg (1973), and adapts some work from Qi (1996a, 1996b), and it operates on the sustained vowel data from vocal task 2.

First, time-synchronous 2048-point segments of the acceleration and aerodynamic signals for a sustained vowel are plotted on separate axes with their respective spectra plotted above them. The spectra are scaled by the gain factor $G = \frac{2}{nw_{DC}}$, where n is the FFT length and w_{DC} is the mean value over time of the windowing function used. For this analysis, the airflow signal has a Hamming window applied to it (w_{DC}=0.5398), and the acceleration signal has a Hanning window applied to it (w_{DC}=0.5). Figure 5.1a shows this first step for subject 5 sustaining the vowel /o/ in a normal voice. Segments centered on the middle portions of the sustained vowel are used, unless either signal is unstable around the midpoint. In that case, a segment near the midpoint is selected such that both signals were stable within the segment.

The second step is to estimate the glottal flow waveform from the airflow signal – see Figure 5.1b. To do this, the vocal tract transfer function **T(f)** must first be estimated using Linear Prediction (Rabiner & Juang, 1993) with 20 coefficients on a Hamming-windowed portion of the airflow signal. The Hamming window length was chosen to encompass 5-6 periods, typically 512 points for females and 1024 points for males. Note that this window length is less than the window shown in the middle right plot of Figure 5.1b, which is the window used to calculate the airflow spectrum. From these Linear Prediction Coefficients (LPC's), the first three formant frequencies and bandwidths of the vocal tract are estimated and displayed immediately below the airflow time signal on the analysis screen. The third formant frequency $\mathbf{f_{F3}}$ is used to estimate the length of the vocal tract $\mathbf{l_{VT}}$, assuming that the vocal tract is a tube with a uniform cross-sectional area that is closed at the glottis, with the formula

$$l_{VT} = \frac{5 \times 35400 \frac{cm}{sec}}{4 \times f_{F3}}$$

Equation 5.1

The frequencies and bandwidths of **F1**, **F2**, and **F3** changed with each vowel and loudness condition, so the vocal tract length estimate also changed. Note that the LPC's sometimes produce unreasonably large bandwidth estimates or formant frequencies that are inappropriate for the vowel being analyzed. Figure 5.1b shows BW_{F3} too large by at least a factor of ten at 3188 Hz. The first three formant frequencies and bandwidths are compared to the peaks in the airflow spectrum and to the data from Peterson and Barney (1952), and altered if necessary in the next step.

The LPC-determined formant frequencies f_{F4} and above, up to 10kHz (half the sampling rate) are replaced with frequencies that again assume a uniform cross-sectional area tube model of the vocal tract using the formula

$$f_{Fn} = \frac{(2n-1) \times 35400 \frac{cm}{sec}}{4 \times l_{VT}}$$

Equation 5.2

and the bandwidths of those upper formants were fixed at 200 Hz.

A model of $\frac{1}{T(f)}$ is implemented as a digital filter and applied to the segment of the airflow signal to obtain an estimate of the glottal flow $\mathbf{v_{VG}(t)}$. One constraint placed on the model of the inverse of the vocal tract transfer function is that its gain at f = 0 Hz must be 1. The glottal flow estimate $\mathbf{v_{VG}(t)}$ is then passed through a fourth-order low-pass Butterworth filter (F_C=1250 Hz) to minimize the effect of the Rothenberg mask's primary anti-resonance (Rothenberg, 1973). Figure 5.1b shows the LPC-derived formants, and the center time plot shows a close-up of a few periods of the airflow-based estimate of $\mathbf{v_{VG}(t)}$, which will be referred to as $\mathbf{v_{VGF}(t)}$.

For the third step, shown in Figure 5.1c, the frequencies and bandwidths of **F1** through **F3** are adjusted as needed to minimize the ripple in the closed portion of the $\mathbf{v_{VGF}(t)}$ waveform. The ripple is minimized visually with the constraints of keeping the frequencies and bandwidths of **F1** through **F3** within reasonable bounds. Frequency bounds are decided for each subject and vowel based on the data of Peterson and Barney (1952). For example, a male subject producing the vowel */a/* would not have $\mathbf{f_{F1}}$ below 500 Hz or above 900 Hz. The bandwidths {$\mathbf{BW_{F1}}$, $\mathbf{BW_{F2}}$, $\mathbf{BW_{F3}}$} are generally reduced to {150, 200, 250} Hz for females and {100, 150, 200} Hz for males, which are about twice as large as the data of Fant (1962) for the closed glottis condition. Table 5.1 shows all of the formant frequencies and bandwidths used for the analysis of vowels produced with a normal voice, arranged by subject and vowel.

Once the ripple is minimized, the derivative of $\mathbf{v_{VGF}(t)}$ was approximated by passing $\mathbf{v_{VGF}(t)}$

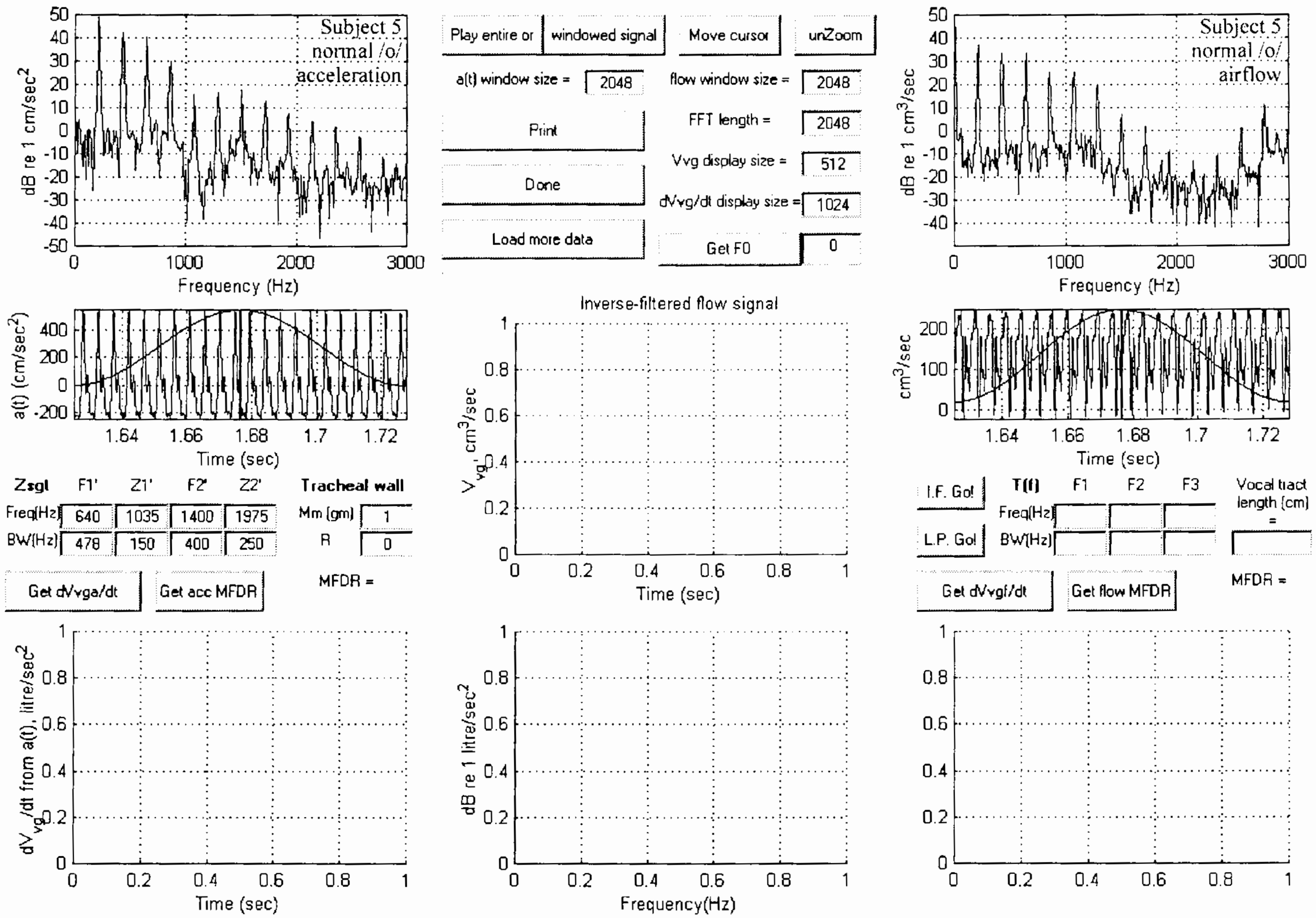


Figure 5.1a. An example of the analysis screen for the MFDR analysis. Step 1: 2048-point (102.4 msec) segments of the acceleration and airflow signals are selected and displayed with their spectra in the upper left and upper right of the screen. The spectra are scaled such that the harmonic magnitudes approximate the Fourier Series for the signals.

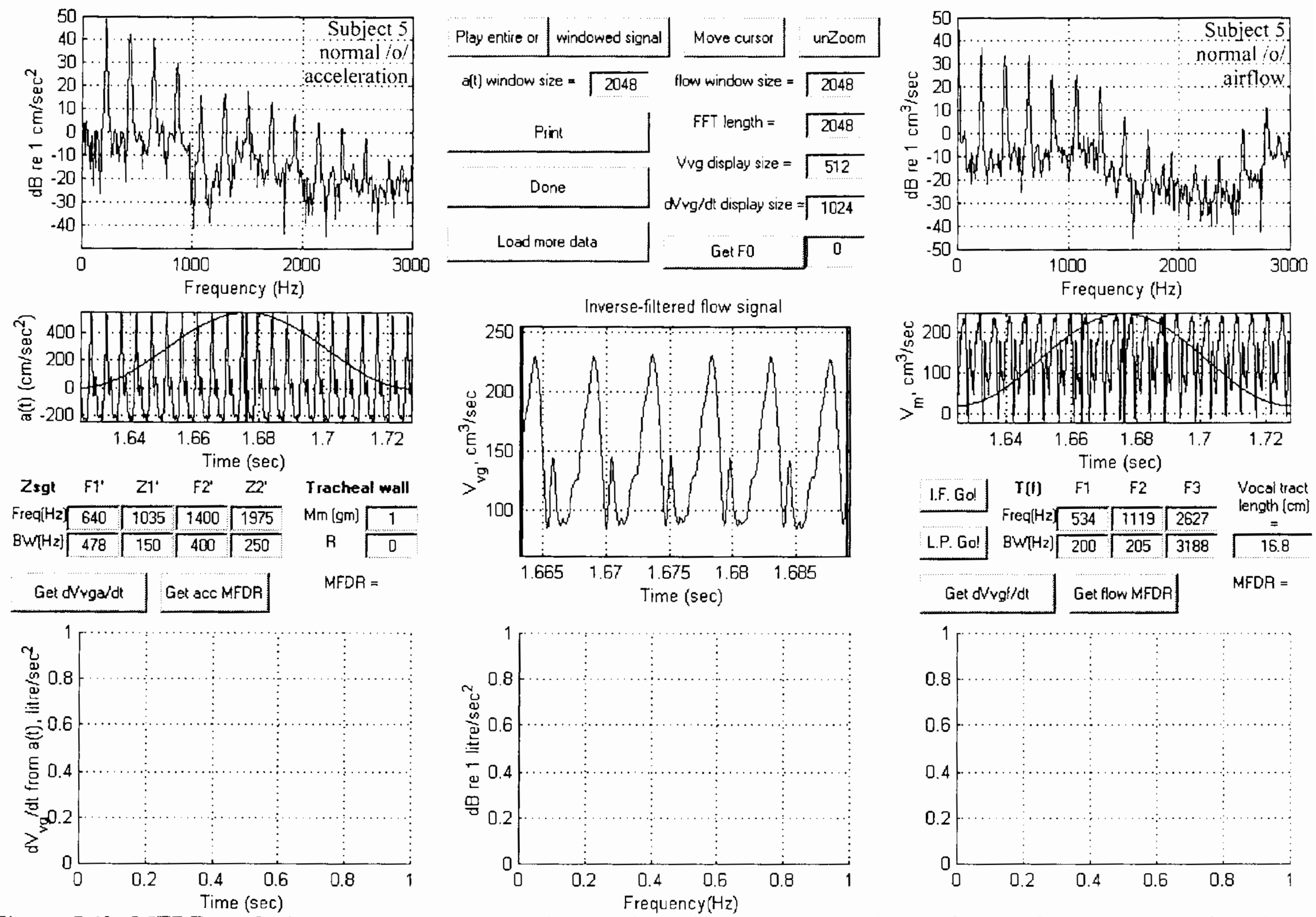


Figure 5.1b. MFDR analysis screen example, step 2: use Linear Prediction to estimate the vocal tract transfer function; show the frequencies and bandwidths of F1, F2, and F3 along with the estimated vocal tract length below the airflow time signal; filter the airflow signal with the inverse of T(f) and show the result in the center plot.

through the 512-point derivative digital filter derivf (see Appendix B) adapted from Qi (1996a),and based on Rabiner and Gold (1975). The resulting glottal flow derivative estimate, $\frac{dv_{VGF}(t)}{dt}$, is displayed in the lower right corner of the analysis screen. The cursor is used to select the approximate locations of six consecutive minima of $\frac{dv_{VGF}(t)}{dt}$, then a subroutine in accflo3 identifies the actual minima within a 50-sample (2.5 millisecond) window centered on each cursor-selected point. The six actual minima are averaged to give the airflow-based Maximum Flow Declination Rate, $\mathbf{MFDR_F}$, which is displayed above the upper right corner of the $\frac{dv_{VGF}(t)}{dt}$ plot as "MFDR = *value*". This method of averaging consecutive points for the MFDR is after Perkell, Holmberg & Hillman (1991), and adapted from Qi (1996b). Also, a line spectrum of $\frac{dv_{VGF}(t)}{dt}$ (i.e., an approximation of the Fourier Series) is plotted in the lower center of the analysis screen shown in Figure 5.1c. This line spectrum is obtained by selecting the fundamental frequency (F0) from the airflow spectrum, then using that F0 value to find the harmonic peaks in the spectrum of $\frac{dv_{VGF}(t)}{dt}$ up to 1500 Hz.

The fourth step in the MFDR analysis operates on the acceleration signal and is shown in Figure 5.1d. Section 3.2.2 showed the modeled relationship between the acceleration **a(t)** and the derivative of the glottal volume velocity $\frac{dv_{VGA}(t)}{dt}$. Using the frequencies and bandwidths of the first two subglottal poles and zeros – **F1′**, **Z1′**, **F2′**, and **Z2′** derived from Ishizaka, Kaneko & Matsudaira (1976), along with a $\mathbf{M_{TW}}$ of 1 gram and an $\mathbf{R_{TW}}$ of 0, a digital filter representation of the inverse of $\mathbf{Z_{SG}(f)}$ and $\mathbf{Z_{TW}(f)}$, minus the zero at f = 0 Hz, is created by swapping the poles and zeros of the two transfer impedances. The filter is scaled by solving the magnitude of the inverse of Equation 3.1 for f = 0 Hz, which gives

$$\left|Z_T^{-1}(0)\right| = \left|\frac{A(1-\Gamma)}{Z_0(1+\Gamma)}\right| = \left|\frac{A}{Z_0} - \frac{2}{Z_{IN}}\right| = 0.0023 \text{ cgs acoustic mhos} \qquad \text{Equation 5.3}$$

where $\mathbf{Z_{IN}}$ is assumed to be the impedance shown in Figure 3.7 without the zero at f = 0 Hz. Subglottal formants and zeros above **Z2′** are not used in creating the digital filter, because the same fourth-order low-pass Butterworth filter used on the airflow signal is applied to the acceleration signal. Furthermore, unlike the vocal tract transfer function **T(f)**, no higher-pole correction is needed. Applying the digital inverse filter and Butterworth filter to **a(t)** produces an estimate of $\frac{dv_{VGA}(t)}{dt}$. The lower left plot in the analysis screen shows $\frac{dv_{VGA}(t)}{dt}$, and the

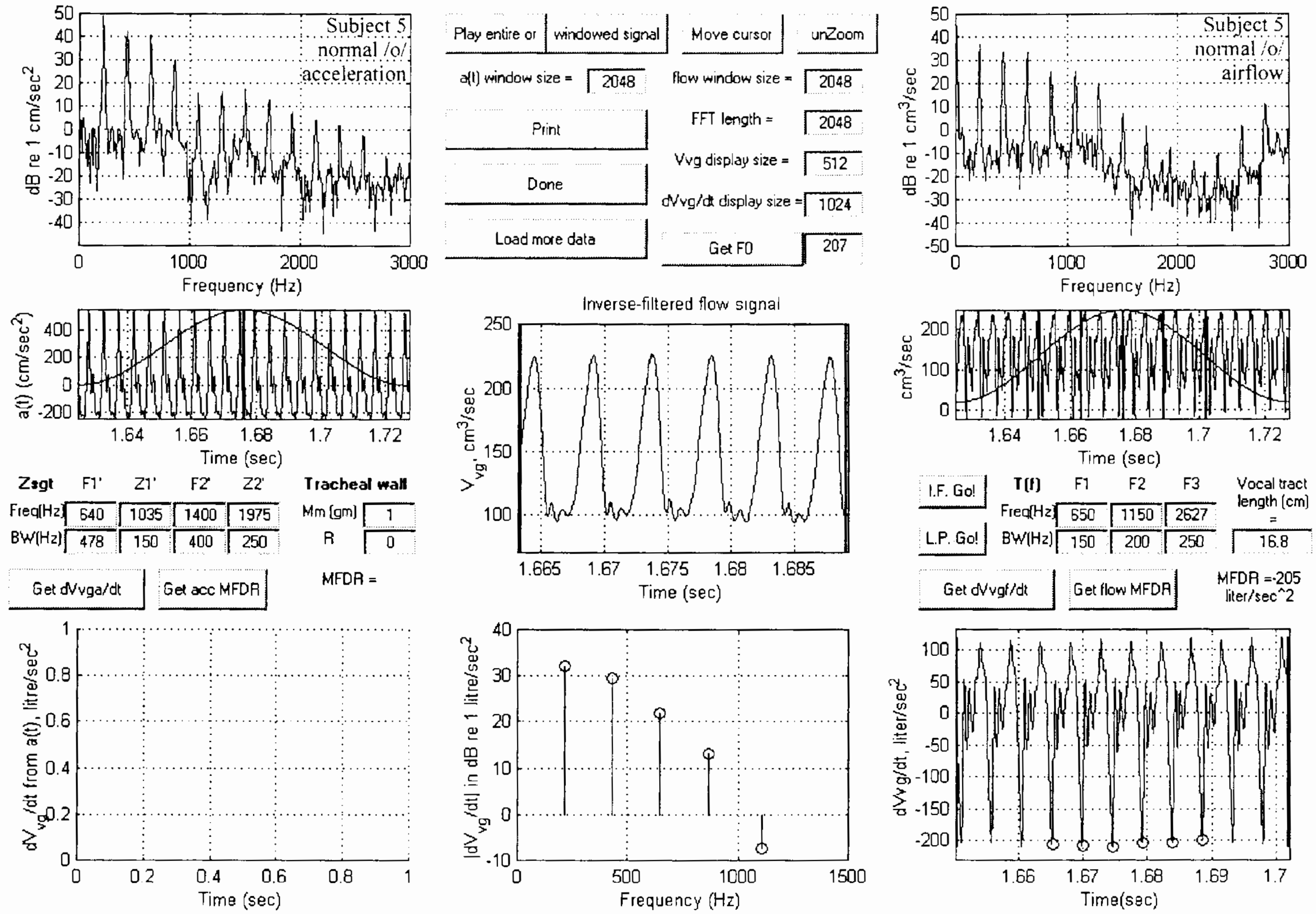


Figure 5.1c. MFDR analysis screen example, step 3: adjust F1, F2, and F3 to minimize the ripple during the closed (i.e., minimal flow) portion of the flow signal (center plot); once the ripple is minimized, get flow derivative (lower right plot) and get the average MFDR from six circled minima; show the flow derivative line spectrum in the lower center plot.

	Subject									
	1	2	3	4	5	6	7	8	9	10
/ae/ f_{F1}	760	912	660	860	860	660	650	700	900	720
/ae/ BW_{F1}	150	150	50	80	140	100	100	100	150	120
/ae/ f_{F2}	1677	1565	1453	1700	1700	1450	1550	1500	1900	1600
/ae/ BW_{F2}	200	200	150	200	200	150	150	150	200	150
/ae/ f_{F3}	2927	2744	2425	2800	2847	2408	2500	2648	2900	2679
/ae/ BW_{F3}	250	250	200	250	250	200	200	200	250	200
/a/ f_{F1}	830	950	670	870	850	700	750	680	900	650
/a/ BW_{F1}	150	150	80	100	200	100	100	100	150	100
/a/ f_{F2}	1512	1114	1130	1200	1460	1107	1500	1200	1700	1260
/a/ BW_{F2}	200	200	150	150	250	85	100	150	200	150
/a/ f_{F3}	2911	2752	2357	3058	2961	2393	2400	2648	3108	2693
/a/ BW_{F3}	250	250	200	250	300	202	200	200	109	200
/i/ f_{F1}	330	370	340	340	420	400	370	220	380	280
/i/ BW_{F1}	150	149	100	150	200	104	100	120	150	100
/i/ f_{F2}	2870	2497	2150	2640	2554	2322	1932	2232	2784	2300
/i/ BW_{F2}	200	116	150	98	200	318	150	200	200	150
/i/ f_{F3}	3400	2949	2918	3123	3216	2718	3001	3247	3386	3010
/i/ BW_{F3}	250	198	200	104	250	315	200	200	250	200
/o/ f_{F1}	550	745	460	660	650	550	450	390	600	390
/o/ BW_{F1}	150	150	80	100	200	100	100	100	150	100
/o/ f_{F2}	1215	1028	900	1073	1150	1016	1000	950	1065	850
/o/ BW_{F2}	200	200	150	176	250	68	150	150	200	150
/o/ f_{F3}	3079	2550	2528	3050	2836	2657	2500	2573	2906	2941
/o/ BW_{F3}	250	250	200	131	300	520	200	200	250	200
/u/ f_{F1}	331	390	350	450	390	340	360	270	400	290
/u/ BW_{F1}	103	100	80	150	100	100	100	100	150	100
/u/ f_{F2}	1117	1410	1050	1001	900	900	800	843	1000	900
/u/ BW_{F2}	200	200	150	149	150	150	150	150	200	150
/u/ f_{F3}	2924	2776	2450	2649	2802	2425	2979	2532	2927	2791
/u/ BW_{F3}	250	250	200	197	250	200	200	200	250	200

Table 5.1. Formant frequencies and bandwidths used for inverse filtering the airflow signal in estimating $v_{VGF}(t)$, from the vowels sustained with a normal voice. Female subjects are denoted by white subject numbers on black, while male subjects are denoted by black subject numbers on white.

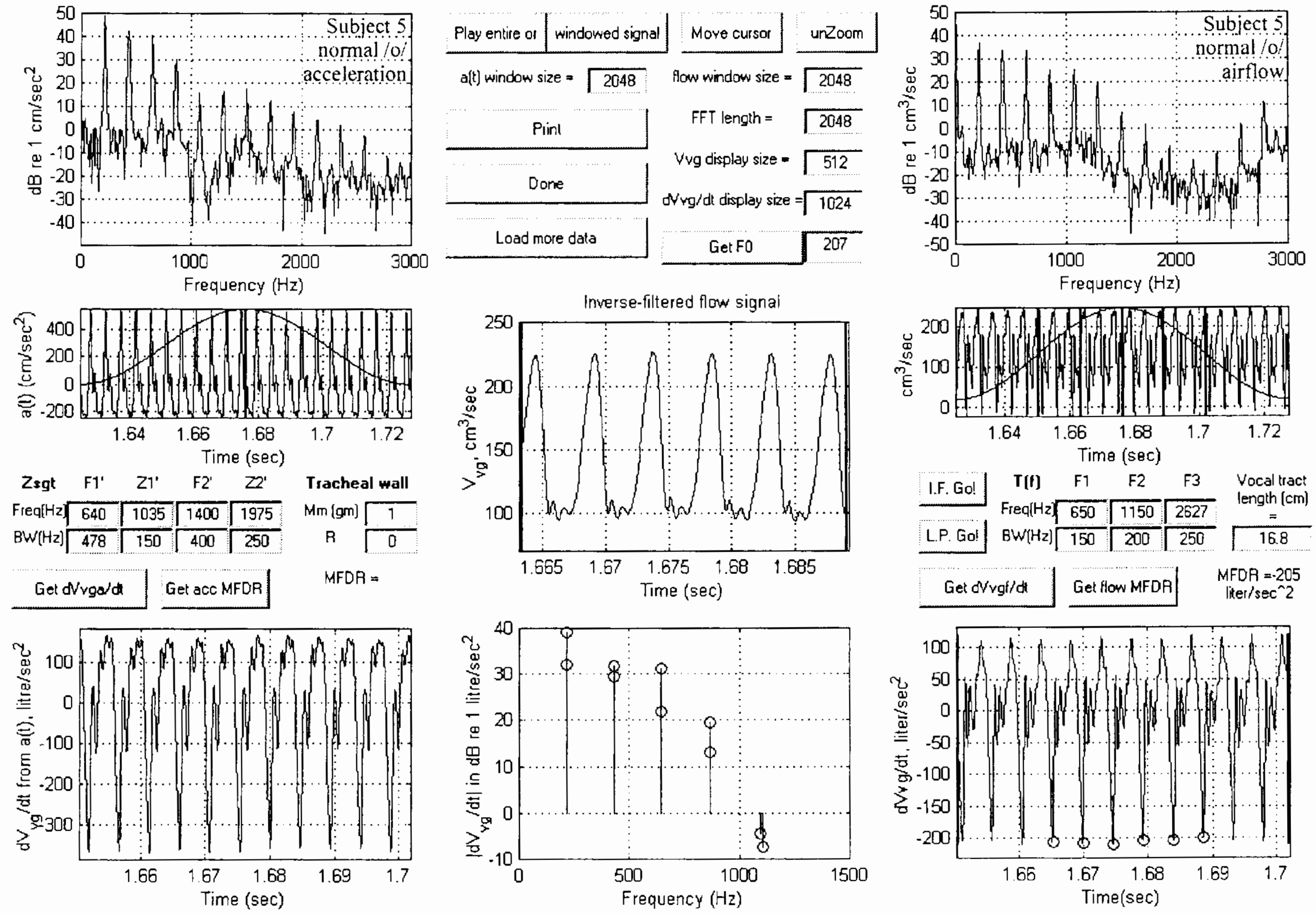


Figure 5.1d. MFDR analysis screen example, step 4: subglottal transfer impedance parameters from Figure 3.7 and a tracheal wall mass of 1 gram inverse filter the acceleration signal to obtain the acceleration-derived glottal flow derivative (lower left plot) and line spectrum (lower center plot) for comparison to the corresponding airflow-derived signals.

line spectrum of $\frac{dv_{VGA}(t)}{dt}$ is plotted in the lower center plot for comparison to the line spectrum of $\frac{dv_{VGF}(t)}{dt}$. Note that during the actual analysis these two line spectra are plotted in two different colors.

The fifth and final step in the MFDR analysis is shown in Figure 5.1e, and involves adjusting the values of F1′, Z1′, F2′, Z2′, M_{TW}, and R_{TW} with the goal of matching first the time signals $\frac{dv_{VGF}(t)}{dt}$ and $\frac{dv_{VGA}(t)}{dt}$, and second their respective line spectra. This matching procedure is done by hand for each subject, while attempting to keep the values of the parameters F1′, Z1′, F2′, Z2′, M_{TW}, and R_{TW} within ±10% of their corresponding maximum and minimum published values. A consistent exception to this ±10% bound was $BW_{F1'}$, which is discussed in Section 5.3. See Table 5.2 for the subglottal transfer impedance parameter values used for each subject in this experiment, and a comparison to two previously published sets of subglottal input impedance values from Ishizaka, Matsudaira & Kaneko (1976), and Cranen & Boves (1987). Subjects' parameter values that match those derived from Ishizaka, Matsudaira & Kaneko (1976) are shown in bold. Assuming that there is no acoustic coupling between the subglottal and supraglottal systems, the values of the subglottal parameters will not depend on the vowel being spoken. So the vowel /ae/ sustained in a normal voice was chosen to set the values of $F1'$, $Z1'$, $F2'$, $Z2'$, M_{TW}, and R_{TW}. Those values then remained fixed for the analysis of the remaining twenty-four other sustained vowels.

Once reasonable matches between the respective time signals and line spectra for the vowel /ae/ sustained in normal voice were obtained, the subglottal parameters were fixed to their values recorded in Table 5.2. Then six consecutive minima from the plot of $\frac{dv_{VGA}(t)}{dt}$ were selected in the same manner as they were selected for $\frac{dv_{VGF}(t)}{dt}$. The means of these six minima were averaged to give $MFDR_A$, the acceleration-derived Maximum Flow Declination Rate, which was displayed above the upper right corner of the $\frac{dv_{VGA}(t)}{dt}$ plot as "MFDR = *value*". Figure 5.1e shows the best match and *MFDR* values for subject 5 and the vowel */o/* in a normal voice, one of the best matches obtained among all subjects. Once the twenty-five sustained vowels were analyzed for each subject, a plot of $MFDR_F$ versus $MFDR_A$ was generated. These plots appear in Section 5.2.

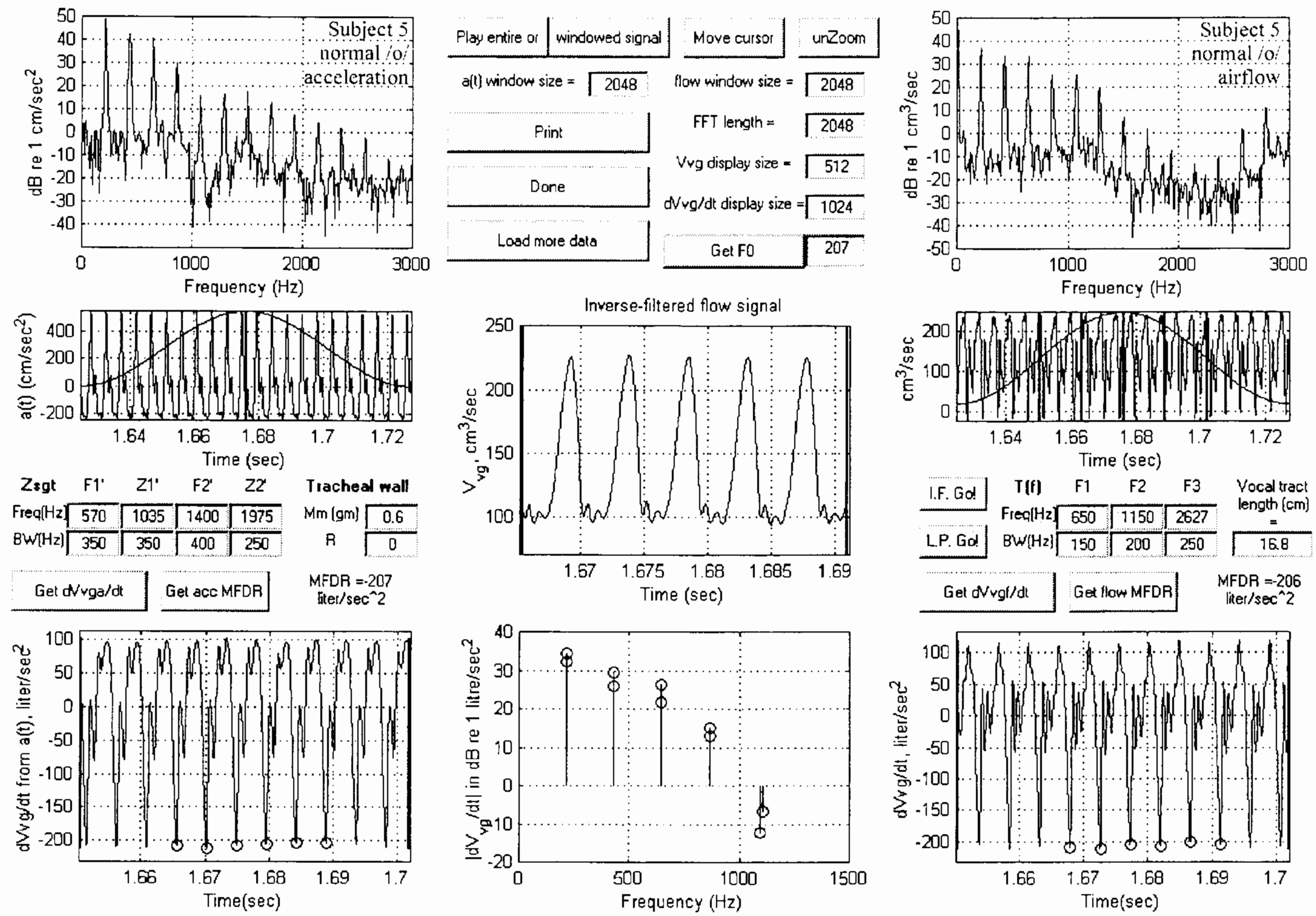


Figure 5.1e. MFDR analysis screen example, step 5: Adjust the subglottal transfer impedance and tracheal wall impedance parameters to the values matched to the vowel /ae/ at normal loudness (see text for matching process), and calculate the acceleration-derived MFDR from the six consecutive minima corresponding to the points selected in the lower right plot.

Subglottal Parameter / Authors or Subject	Ishizaka, Matsudaira & Kaneko (1976)	Cranen & Boves (1987)	1	2	3	4	5	6	7	8	9	10
$f_{F1'}(Hz)$	640	510	670	**640**	**640**	620	570	**640**	500	550	**640**	500
$BW_{F1'}(Hz)$	478	654	150	250	**478**	200	350	450	150	200	350	150
$f_{Z1'}(Hz)$	1035	-	1130	**1035**	**1035**	1140	**1035**	900	**1035**	900	**1035**	**1035**
$BW_{Z1'}(Hz)$	150	-	**150**	350	**150**	**150**	350	**150**	**150**	**150**	350	**150**
$f_{F2'}(Hz)$	1400	1355	1550	**1400**	**1400**	1500	**1400**	**1400**	**1400**	**1400**	**1400**	**1400**
$BW_{F2'}(Hz)$	400	967	**400**	**400**	**400**	**400**	**400**	**400**	200	**400**	**400**	**400**
$f_{Z2'}(Hz)$	1975	-	2100	**1975**	**1975**	1800	**1975**	**1975**	**1975**	**1975**	**1975**	**1975**
$BW_{Z2'}(Hz)$	250	-	450	**250**	**250**	**250**	**250**	**250**	**250**	**250**	**250**	**250**
$M_{TW}(gm)$	-	-	0.75	1.1	1.37	2.2	0.6	1.1	1.1	0.85	2.5	1
R_{TW}	-	-	1250	0	0	0	0	0	800	1200	0	0

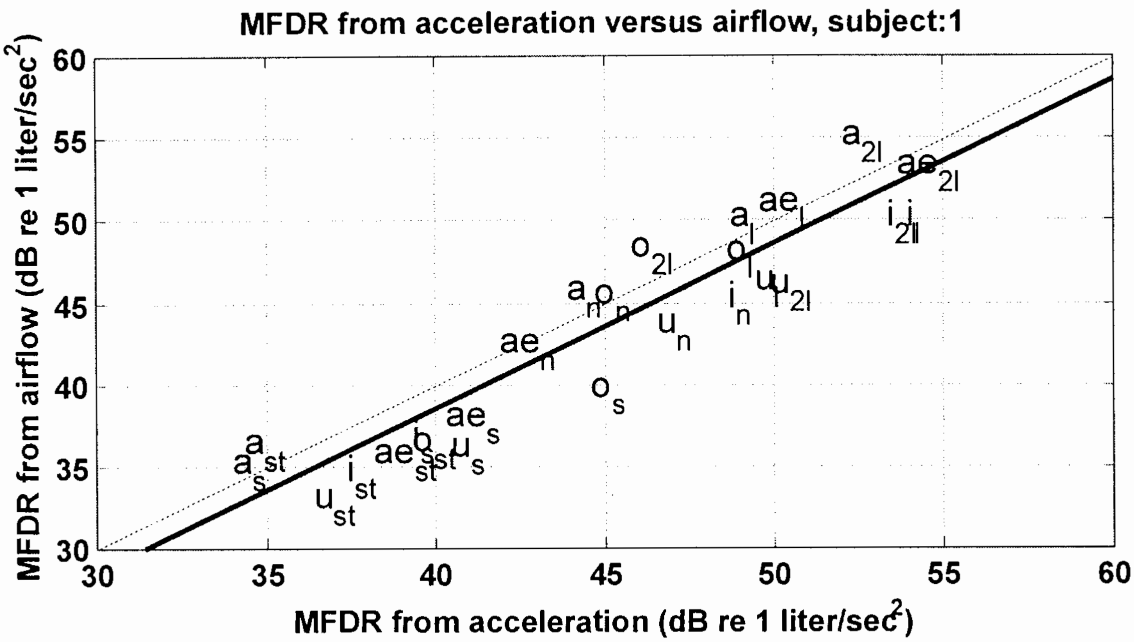
Table 5.2. Subglottal transfer impedance pole and zero frequencies and bandwidths used for each subject in Experiment II, with published values of subglottal input impedance pole and zero frequencies and bandwidths for reference. Also listed are the tracheal wall mass and resistance model parameters. A dash through a cell indicates that particular parameter value was not reported by the authors. Values in BOLD indicate subject values that equal those of Ishizaka, Matsudaira, and Kaneko (1976). White subject numbers on black indicate the female subjects, while black subject numbers on white indicate the male subjects.

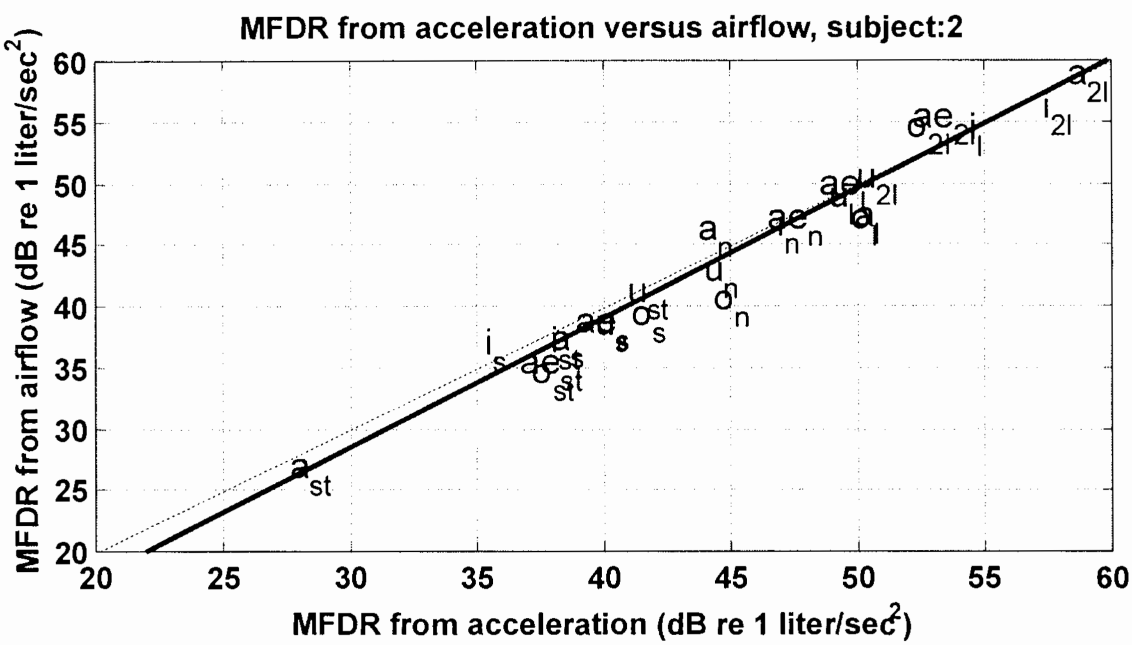
5.2 Results from Experiment II: Estimating MFDR from the acceleration using the vocal system model

Estimates of the MFDR using the acceleration signal (**$MFDR_A$**) are plotted against the corresponding airflow-derived MFDR (**$MFDR_F$**) in Figures 5.2 through 5.11 below. The MFDR values are plotted in dB (i.e., $20\times\log_{10}$MFDR) due to the large range of MFDR values obtained for the five loudness conditions. Each figure has twenty-five points, corresponding to the five vowels and five loudness conditions from vocal task 1. The points are plotted as the vowel with a subscript denoting the loudness; see the figure captions for more explanation of this. Also in the figures are a dashed line of slope m=1, and a solid line of best fit in the least-squares sense. Table 5.3 lists the mean error between **$MFDR_A$** and **$MFDR_F$**, the standard deviation of that error, the equation for the line of best fit, and the correlation coefficient for that line for each subject, as well as the mean error and standard deviation of the error for all subjects excluding subject 6.

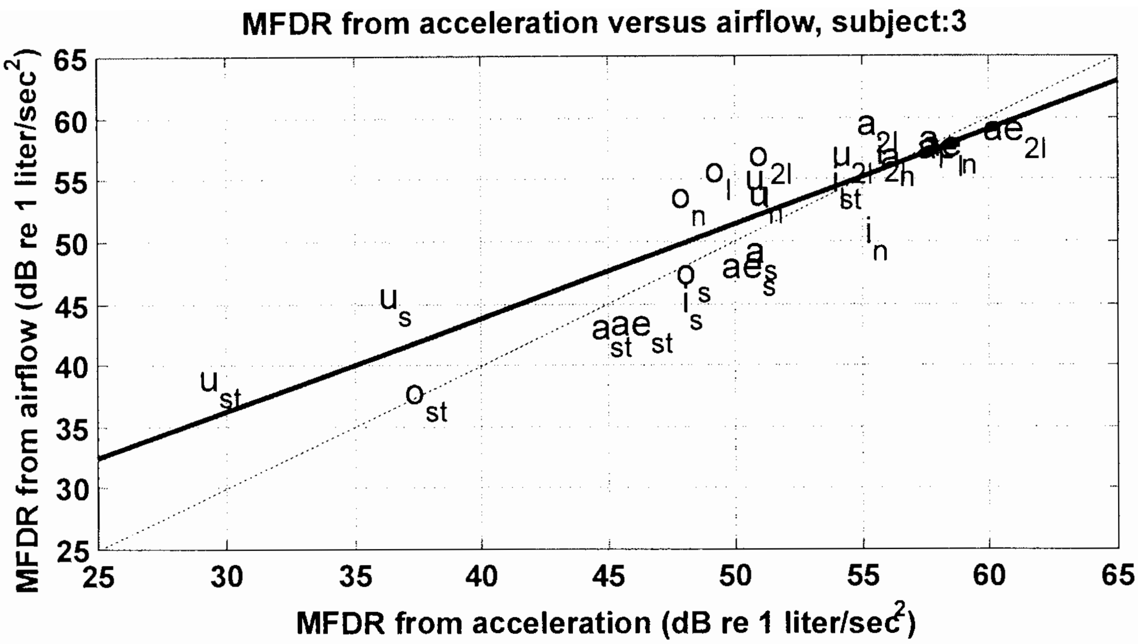
The data from subject 6 are excluded from the overall mean error computations because this subject was informally diagnosed with bilateral vocal fold sulci after the recording and data analysis for this experiment took place. Reviewing Figure 5.7 may provide some insight into how this acceleration-derived estimate of MFDR behaves when applied to a subject with pathology. Note the **$MFDR_A$** values for the twice as loud task for all vowels equal those of the softest task for vowels /o/ and /u/. Also the /o/ and /i/ vowels in loud voice produced an **$MFDR_A$** less than that of their corresponding normal and soft voice.

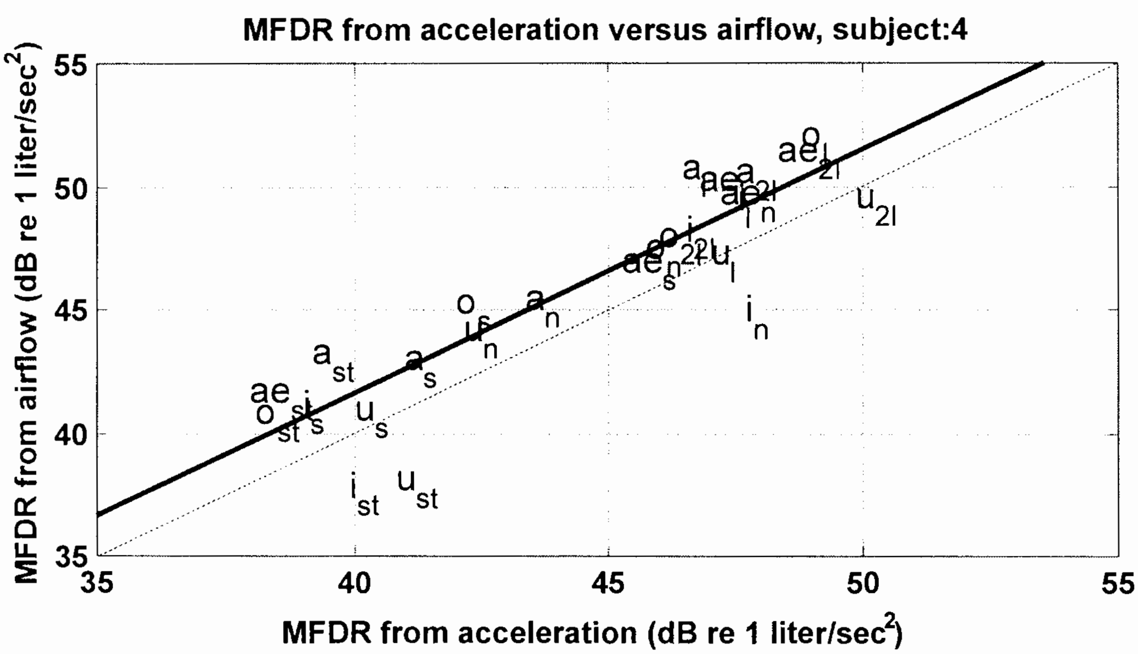
However, the results shown in Figures 5.2 through 5.11 suggest in general that the vocal system model can predict the airflow-derived MFDR with reasonable accuracy. For all subjects, the mean error is less than 2 dB, and for 8 of 10 subjects (subjects 1, 2, 4-9) the slope of the line of best fit is not significantly different from m = 1 (p = 0.05). For all subjects excluding subject 6, the mean error is –0.5 dB and the standard deviation of the error is 2.9 dB. If the error is assumed to follow a Gaussian distribution with zero mean, then approximately 95% of the MFDR predictions will fall within ±5.8 dB. Particularly high correlation values ($r^2>0.75$) and error standard deviations that are less than the overall average of 3.1 dB are achieved with subjects 1, 2, 4, 5, 7, and 9. The results from subject 6, and the rest of the results, are discussed further in Section 5.3.



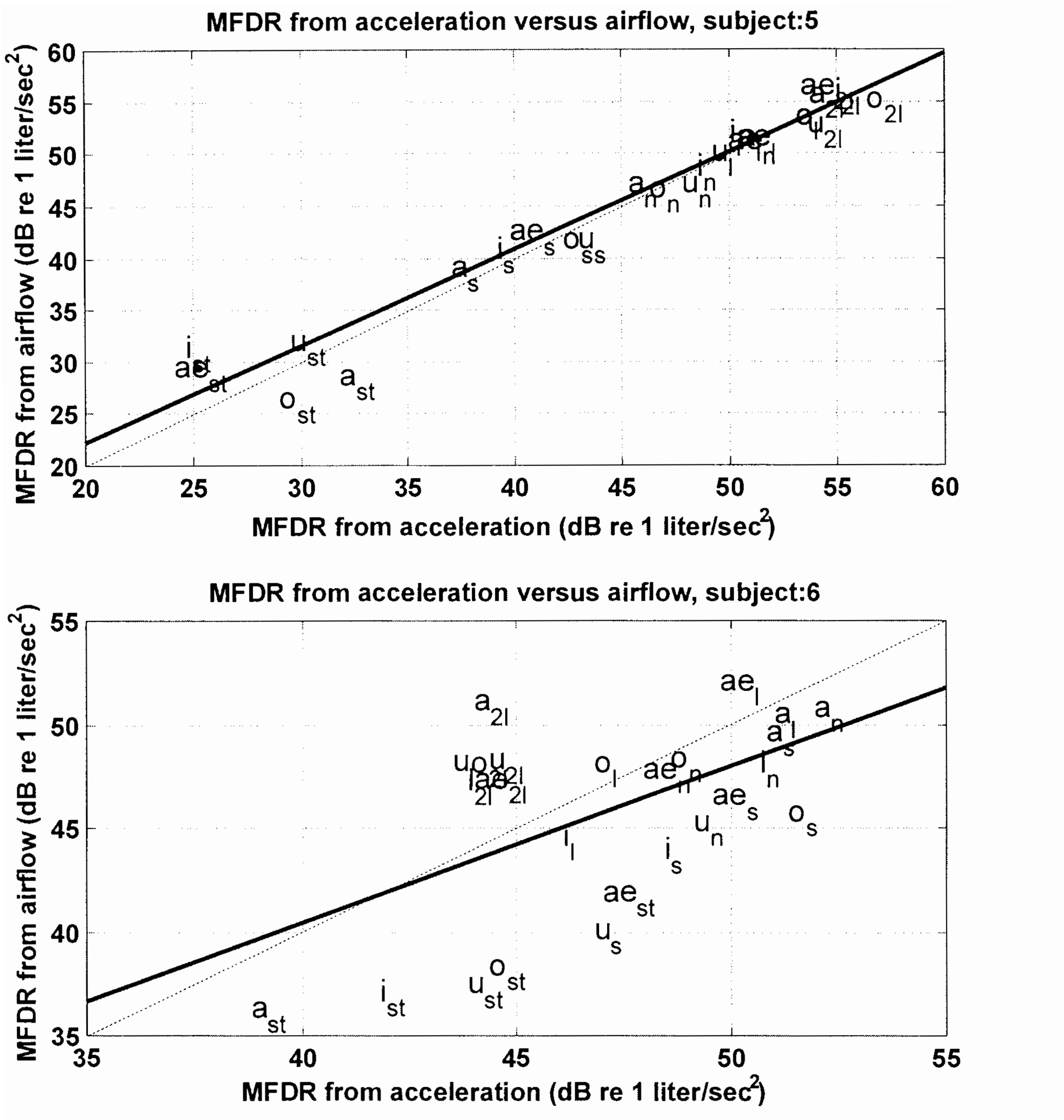


Figures 5.2 (upper) and 5.3 (lower). MFDR from airflow ($MFDR_F$) versus MFDR from acceleration ($MFDR_A$) for subjects 1 and 2. Each vowel and loudness condition is plotted as that vowel (a, ae, i, o, u) with a subscript for the loudness (st = softest, s = soft, n = normal, l = loud, 2l = twice as loud). The dashed line has slope m = 1, and the solid line is a least-squares best fit. See Table 5.3 for line of best fit equations, correlation coefficients, mean errors and error standard deviations.

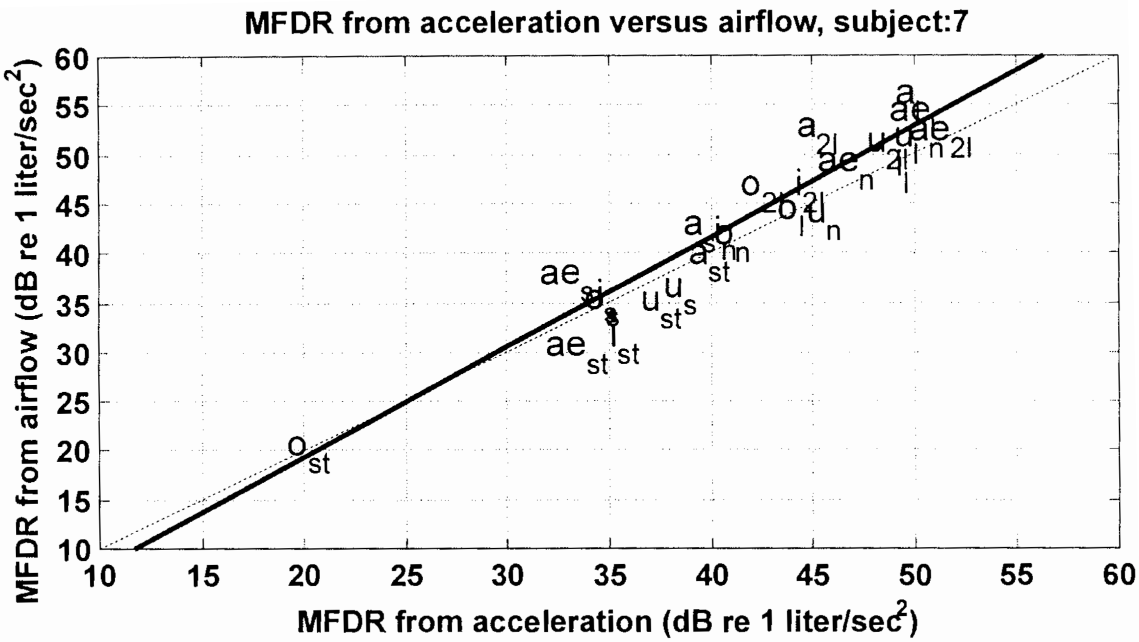


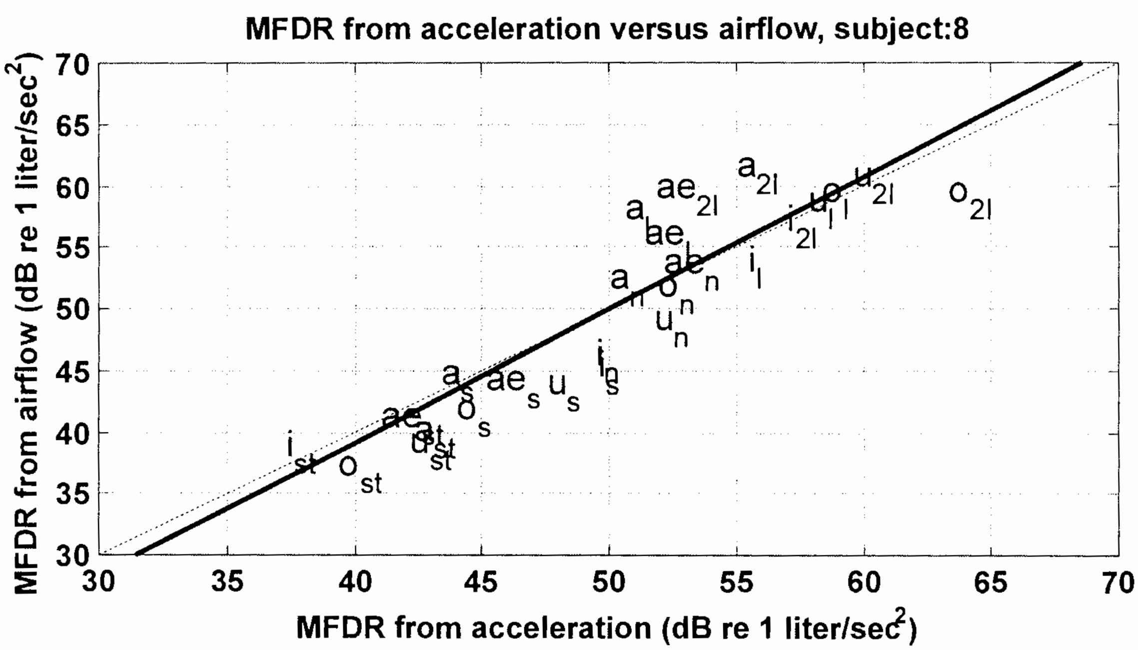


Figures 5.4 (upper) and 5.5 (lower). MFDR from airflow ($MFDR_F$) versus MFDR from acceleration ($MFDR_A$) for subjects 3 and 4. Each vowel and loudness condition is plotted as that vowel (a, ae, i, o, u) with a subscript for the loudness (st = softest, s = soft, n = normal, l = loud, 2l = twice as loud). The dashed line has slope m = 1, and the solid line is a least-squares best fit. See Table 5.3 for line of best fit equations, correlation coefficients, mean errors and error standard deviations.

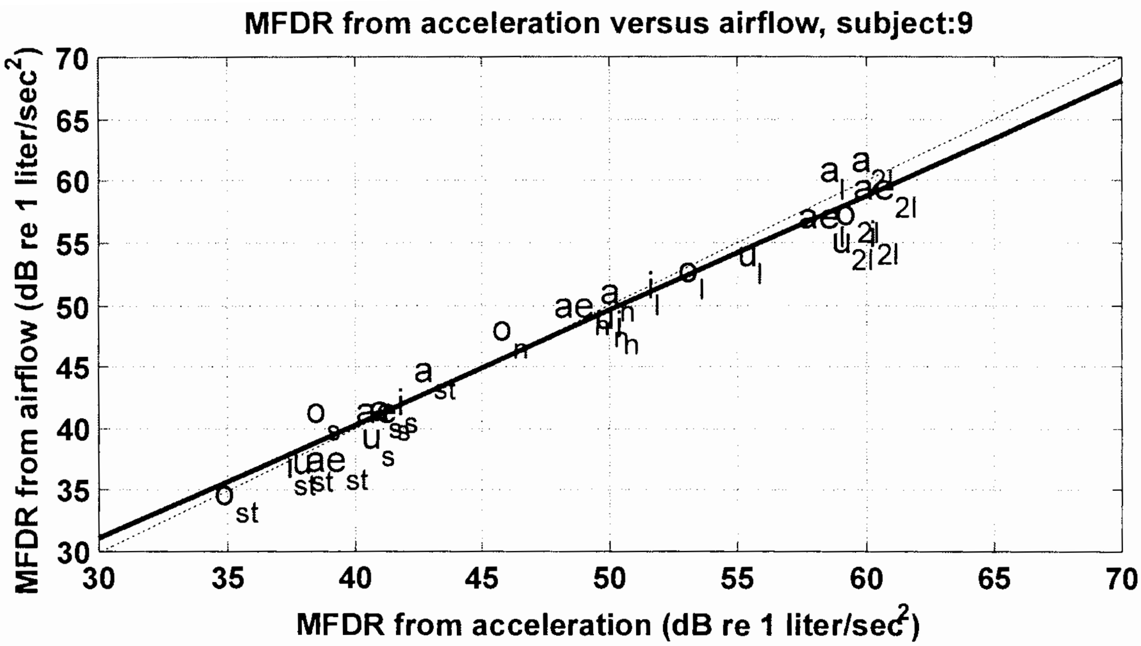


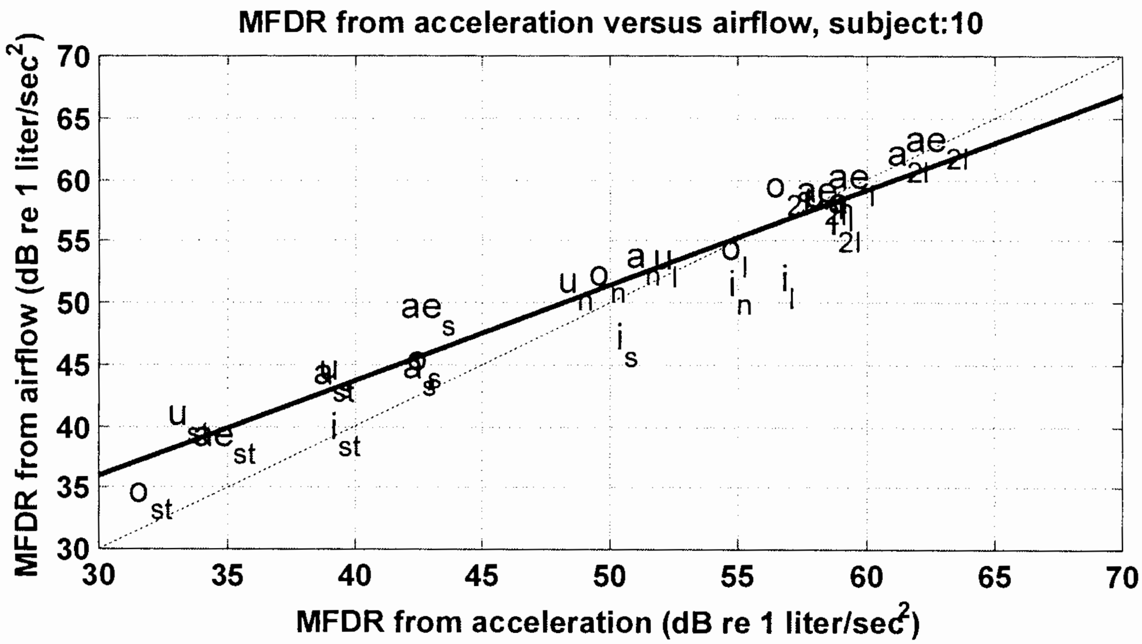
Figures 5.6 (upper) and 5.7 (lower). MFDR from airflow ($MFDR_F$) versus MFDR from acceleration ($MFDR_A$) for subjects 5 and 6. Each vowel and loudness condition is plotted as that vowel (a, ae, i, o, u) with a subscript for the loudness (st = softest, s = soft, n = normal, l = loud, 2l = twice as loud). The dashed line has slope m = 1, and the solid line is a least-squares best fit. See Table 5.3 for line of best fit equations, correlation coefficients, mean errors and error standard deviations.





Figures 5.8 (upper) and 5.9 (lower). MFDR from airflow ($MFDR_F$) versus MFDR from acceleration ($MFDR_A$) for subjects 7 and 8. Each vowel and loudness condition is plotted as that vowel (a, ae, i, o, u) with a subscript for the loudness (st = softest, s = soft, n = normal, l = loud, 2l = twice as loud). The dashed line has slope m = 1, and the solid line is a least-squares best fit. See Table 5.3 for line of best fit equations, correlation coefficients, mean errors and error standard deviations.





Figures 5.10 (upper) and 5.11 (lower). MFDR from airflow ($MFDR_F$) versus MFDR from acceleration ($MFDR_A$) for subjects 9 and 10. Each vowel and loudness condition is plotted as that vowel (a, ae, i, o, u) with a subscript for the loudness (st = softest, s = soft, n = normal, l = loud, 2l = twice as loud). The dashed line has slope m = 1, and the solid line is a least-squares best fit. See Table 5.3 for line of best fit equations, correlation coefficients, mean errors and error standard deviations.

Subject	$20\log_{10}\left(\frac{MFDR_A}{MFDR_F}\right)$	$\sigma_{20\log_{10}\left(\frac{MFDR_A}{MFDR_F}\right)}$	Line of best fit, $\hat{Y}$ = m[$20\log_{10}MFDR_A$]+b		r^2
1	1.4	2.5	**m = 1.00**	b = -1.4	0.86
2	0.7	1.7	**m = 1.05**	b = -3.0	0.96
3	-1.5	3.7	m = 0.76	b = 13.3	0.75
4	-1.6	1.9	**m = 0.99**	b = 2.0	0.79
5	-0.6	2.2	**m = 0.94**	b = 3.4	0.95
6	1.2	3.9	**m = 0.76**	b = 10.1	0.31
7	-1.8	2.9	**m = 1.12**	b = -3.1	0.90
8	0.1	3.5	**m = 1.08**	b = -4.0	0.82
9	0.3	1.9	**m = 0.93**	b = 3.2	0.95
10	-1.7	3.3	m = 0.77	b = 12.8	0.90
1-5, 7-10	-0.5	2.9	-		-

Table 5.3. Mean error and error standard deviation (where error = $MFDR_A$-$MFDR_F$ in dB), linear best-fit equations and correlation coefficients (r^2) for all subjects, to describe the relationship between the acceleration-derived MFDR ($MFDR_A$) and the airflow-derived MFDR ($MFDR_F$). Linear best fit slopes that appear in BOLD indicate that the slope is not significantly different from m = 1 (p = 0.05). The last row gives the overall mean error and error standard deviation, for all subjects excluding subject 6 (see text).

5.3 Discussion on Experiment II: Estimating MFDR from the acceleration using the vocal system model

For the analysis procedure, several vocal system parameters are adjusted for the matching procedure on the vowel /ae/ produced at normal loudness. As mentioned in Section 5.1, attempts are made to keep these parameters within reasonable limits. The vocal tract formant bandwidths used for **F1**, **F2**, and **F3** are initially guided by data from Fant (1962), but many subjects needed bandwidths for **F1** that are twice to three times what Fant reported. However, Fant's measurements were made in the closed-glottis condition, and the analysis here is performed over several fundamental periods of phonated vowels. More recently, Hanson & Chuang (1999) and Hanson (1997) estimated the bandwidth of **F1** for males and females producing the vowel /ae/, and found a mean ± 1 standard deviation of 126 ± 55 Hz and 165 ± 34 Hz respectively. These bandwidths agree well with most subjects' $\mathbf{BW_{F1}}$, which are listed in Table 5.1 and are generally between 100 and 150 Hz.

The tracheal wall parameters $\mathbf{M_{TW}}$ and $\mathbf{R_{TW}}$ are also adjusted during the matching procedure for the vowel /ae/ at normal loudness (see Table 5.2). The resistance $\mathbf{R_{TW}}$ was non-zero for three subjects: 1250 grams/sec for subject 1 (female), 800 grams/sec for subject 7 (male), and 1200 grams/sec for subject 8 (male). These non-zero values of $\mathbf{R_{TW}}$ reduced the low-frequency gain around **F0** of the tracheal wall inverse filter, and are within the range of values given by Ishizaka, French & Flanagan (1975) and Stevens (1998, p. 26). Because $\mathbf{R_{TW}}$ was not consistently non-zero for females or males, these values suggest that $\mathbf{R_{TW}}$ varies considerably across subjects. Likewise, $\mathbf{M_{TW}}$ varied considerably between 0.6 grams (subject 5) and 2.5 grams (subject 9), although 8 out of the 10 subjects' values fell in the range 0.6-1.37 grams, which agrees again with the ranges of Ishizaka, French & Flanagan (1975) and Stevens (1998, p. 26). Two subjects (4 and 9, both female) needed particularly high values of M_{TW} – 2.2 and 2.5 respectively. Note that changes in $\mathbf{M_{TW}}$ simply apply differing amounts of gain in the inverse filtering process, which is analogous to shifting the spectrum of $|\mathbf{V_{VGA}}|$ vertically. These seemingly large values of $\mathbf{M_{TW}}$ could be true variation in the tracheal wall mass, or they could be correcting for other fixed parameters or inadequacies of the model. One fixed parameter of the model is the cross-sectional area **A** of the trachea; it is fixed across subjects at 2.5 cm^2 (see Equations 3.1 and 5.3). This area acts as another gain factor that may vary across subjects, and thus its effect on the acceleration may be counteracted by varying $\mathbf{M_{TW}}$. An example of a model inadequacy that may be offset by varying $\mathbf{M_{TW}}$ is the effect of acoustic coupling between the subglottal and supraglottal systems. With increased coupling and thus loss through the glottis, $\mathbf{BW_{F1'}}$ would increase, and that would effectively decrease the peak magnitude in the spectrum of $|\mathbf{V_{VGA}}|$ near $\mathbf{F1'}$ and thus affect $\mathbf{MFDR_A}$. Since the current vocal system model assumes no coupling, increasing $\mathbf{M_{TW}}$ could boost the spectrum near $\mathbf{F1'}$ to offset the loss.

The bandwidths of the subglottal transfer impedance also are manipulated for the MFDR matching. Although $\mathbf{BW_{Z1'}}$, $\mathbf{BW_{F2'}}$, and $\mathbf{BW_{Z2'}}$ are generally set to the values published by Ishizaka, Matsudaira & Kaneko (1976), $\mathbf{BW_{F1'}}$ is set to a value less than either previously published measurement for that bandwidth (see Table 5.2) in 9 of 10 subjects. Again, this may reflect variation across subjects' subglottal transfer impedances, but these values are likely exposing deficiencies in the vocal system model. Assume again that the loss through some glottal coupling is increasing $\mathbf{BW_{F1'}}$ and thereby reducing the magnitude of the $|\mathbf{V_{VGA}}|$ spectrum around $\mathbf{F1'}$. Inverse-filtering the acceleration signal using a $\mathbf{Z_T(f)}$ that has a decreased $\mathbf{BW_{F1'}}$ would offset that effect.

Note that the frequencies of the first subglottal zero $\mathbf{f_{Z1'}}$ shown in Table 5.2 are not consistent with the mean values of $\mathbf{f_{Z1'}}$ found in Section 4.1.2 and shown in Figures 4.13 through 4.12. A particularly extreme example of this is subject 9, whose spectrogram in Figure 4.11 shows an $\mathbf{f_{Z1'}}$ of about 1300 Hz, but whose inverse filter for the subglottal transfer impedance uses an $\mathbf{f_{Z1'}}$ of 1035 Hz. This difference is likely due to the different goals of the two experiments. The experiment in Section 4.1.2 is performed to estimate $\mathbf{f_{Z1'}}$ over time, while the experiment in this section aims to match the minima of two time signals by manipulating $\mathbf{f_{Z1'}}$.

Even with the simple vocal system model used to estimate MFDR from the acceleration signal, the results in Section 5.2 show encouraging correlation between $\mathbf{MFDR_A}$ and $\mathbf{MFDR_F}$. In addition, the MFDR values obtained from both the acceleration and the airflow agree well with two other published reports of MFDR in normal voices – see Table 5.4 below.

In general, the mean MFDR values found in this study are higher than those of Holmberg et al. (1995) and Holmberg et al. (1994), although across studies they are all roughly within one standard deviation of each other. These differences are probably due to a lack of control on SPL in both this study and Holmberg et al. (1994), where subjects were instructed for loudness with phrases like "louder than normal". Somewhat surprisingly, the Holmberg et al. (1995) study had subjects produce a loud voice that was 6±1 dB louder than normal, and the resulting MFDR values are almost identical to the Holmberg et al. (1994) results.

The matching procedure used to set the vocal system model parameters started with the vowel /ae/ in a normal voice because that vowel is close to being neutral for a female voice. As a result, worse matches of $\mathbf{MFDR_A}$ and $\mathbf{MFDR_F}$ often occur for the non-low vowels /i/, /o/ and /u/. For example, see the results of subject 1 in Figure 5.2. Eight of ten $\mathbf{MFDR_A}$ and $\mathbf{MFDR_F}$

Sex, Loudness	$\overline{MFDR_A} \pm \sigma_{MFDR_A}$	$\overline{MFDR_F} \pm \sigma_{MFDR_F}$	Holmberg et al. (1994)	Holmberg et al. (1995)
female, soft	120 ± 39	124 ± 57	102 ± 40	N.A.
female, normal	225 ± 69	258 ± 85	184 ± 63	172 ± 71
female, loud	309 ± 79	400 ± 154	374 ± 130	372 ± 140
male, soft	188 ± 114	192 ± 81	167 ± 57	N.A.
male, normal	462 ± 262	513 ± 272	337 ± 127	N.A.
male, loud	503 ± 260	635 ± 223	650 ± 251	N.A.

Table 5.4. Comparison of this study's MFDR mean and standard deviation values, from both the acceleration and airflow signals, to two previously published reports. All values are for the vowel /ae/ only, and are in liters/sec^2.

values for the vowels /a/ and /ae/ are within 3 dB of each other, while the majority of the values for the other vowels show $\mathbf{MFDR_A}$ greater than $\mathbf{MFDR_F}$ by more than 3 dB. Subject 2, in Figure 5.3, shows very good agreement for all the vowels except /o/, which has greater $\mathbf{MFDR_A}$ than $\mathbf{MFDR_F}$ by 2 dB for the conditions softest, softer than normal, normal, and louder than normal. Also see the results of subject 3 in Figure 5.4. All of this subject's $\mathbf{MFDR_A}$ and $\mathbf{MFDR_F}$ values for the vowels /a/ and /ae/ are within 2 dB of each other, except for the twice as loud as normal /ae/. In contrast, note how the $\mathbf{MFDR_A}$ for /u/ is consistently below $\mathbf{MFDR_F}$ by up to 10 dB, and similarly for /o/ by up to 8 dB. Subject 9, who shows remarkably high correlation and low error standard deviation between $\mathbf{MFDR_A}$ and $\mathbf{MFDR_F}$ in Figure 5.10, also displays the greatest differences between these values for the /i/ and /u/ vowels in the twice as loud as normal condition. Large differences at the twice as loud as normal condition like these may arise due to a dynamic increase in $\mathbf{R_{TW}}$. That is, the large subglottal pressure required to produce loud phonation may increase the mechanical resistance of the tracheal wall, thereby boosting the measured acceleration since the velocity of the accelerometer for a given subglottal pressure $\mathbf{P_A}$ depends on $\frac{R_{TW} + j\omega M_{TW}}{P_A A_S}$, according to the vocal system model.

Despite the choice to set the model parameters using the vowel /ae/, many subjects show good agreement between $\mathbf{MFDR_A}$ and $\mathbf{MFDR_F}$ for the vowels /i/, /o/, and /u/ in particular loudness conditions and even across all loudness conditions. Subject 2 displays $\mathbf{MFDR_A}$ and $\mathbf{MFDR_F}$

values within 2 dB of each other for all loudness conditions of the vowels /i/ and /u/, in Figure 5.3. In Figure 5.5, subject 4 shows the majority of her $\mathbf{MFDR_A}$ values are less than her $\mathbf{MFDR_F}$ values by about 2 dB, but the softer than normal, louder than normal, and twice as loud as normal conditions for the vowel /u/ match within 1 dB. And for subject 8 in Figure 5.9, almost all of the louder than normal and twice as loud as normal conditions for the vowels /i/, /o/, and /u/ have better matches between $\mathbf{MFDR_A}$ and $\mathbf{MFDR_F}$ than do the vowels /a/ and /ae/.

The results from estimating the MFDR from the acceleration signal and comparing it to the conventional airflow-based measure suggest a few ways to modify the vocal system model with the aim of improving the agreement between these two methods. First, more physiologically accurate values of the tracheal wall parameters $\mathbf{M_{TW}}$ and $\mathbf{R_{TW}}$ may reduce the error inherent in varying these values to correct for spectral differences. One possibility would be to introduce a known pressure into the vocal system and measure the resulting acceleration on the neck. By knowing the geometry of the vocal system, assuming it could be accurately measured using a technique as described by Fredberg et al. (1980), the pressure deep to the accelerometer could be calculated and used to find $\mathbf{M_{TW}}$ and $\mathbf{R_{TW}}$. Second, the current vocal system model assumes complete glottal closure. This idealized case has been shown to be more the exception than the rule in reality (Holmberg et al., 1995; Holmberg et al., 1994; Holmberg, Hillman & Perkell, 1988), so including a provision for incomplete glottal closure in the model may improve its ability to estimate MFDR from the acceleration. This addition could be as simple as a fixed acoustic mass, with area equal to an average "glottal chink" area, replacing the glottal impedance $\mathbf{Z_G}$ as shown in Figure 2.2. Third, the matching procedure used for setting the subglottal system parameters may be improved by averaging the parameters over more than one vowel and/or loudness condition. The matching procedure was designed with the long-term goal of having the acceleration signal calibrated to a given subject with the airflow signal as a comparison, and then obtaining MFDR estimates only from the acceleration signal. Changing this calibration to include /ae/, /i/, and /u/ would give a reasonable span of the **F1-F2** vowel space, and including softer than normal, normal, and louder than normal conditions would provide a reasonable range of typical loudness.

6 Experiment III: Estimating SPL from the acceleration using the vocal system model

Chapter 5 investigates the vocal system model's ability to estimate MFDR from the acceleration signal, following motivation, experimental methods and a description of the model in Chapters 1 through 4. This chapter extends the investigation of the vocal system model to SPL. Specific analyses of the acceleration signal using the model, the results of these analyses and a discussion of the results are presented here.

6.1 Using the model to estimate SPL from the acceleration

A block diagram of the model-based signal processing done on the acceleration signal to estimate the speech spectrum is shown in Figure 3.10. This processing was accomplished using the MATLAB® program accspl1 – see Appendix B. Figures 6.1a through 6.1d are images from that program's analysis screen intended to illustrate the signal processing procedure schematized in Figure 3.10.

As in the MFDR analysis, the first step is to select segments of both the acceleration and microphone pressure signals, 2048 points (102.4 milliseconds) in length, from the middle portions of each sustained vowel. If the vowel is unstable around the midpoint, then the nearest stable 2048-point window to the midpoint is selected. Figure 6.1a shows this segment selection step.

The second step, shown in Figure 6.1b, involves estimating the magnitude of the glottal volume velocity $|\mathbf{V_{VGA}}|$ from the measured acceleration. First the subglottal parameters $\mathbf{f_{F1'}}$, $\mathbf{BW_{F1'}}$, $\mathbf{f_{F2'}}$, $\mathbf{BW_{F2'}}$, $\mathbf{f_{F3'}}$, $\mathbf{BW_{F3'}}$, $\mathbf{M_{TW}}$ and $\mathbf{R_{TW}}$ were fixed within each subject and across vowels to their values from Experiment II (see Table 5.2). As described in Section 3.2.2.4, a digital filter representing the inverse of the subglottal system plus a pure differentiator is implemented to estimate the first derivative of the glottal volume velocity, $\frac{dv_{VGA}(t)}{dt}$. The Fast Fourier Transform (FFT) of $\frac{dv_{VGA}(t)}{dt}$ is then taken and its magnitude is divided by $2\pi f$ and scaled by the gain factor $G = \frac{2}{nw_{DC}}$, where n is the FFT length and w_{DC} is the mean value over time of the windowing function used. This gives the spectral magnitude $|\mathbf{V_{VGA}}|$ corresponding to the Fourier Series of the glottal volume velocity, shown in the lower left corner of the analysis screen.

Estimating the spectral magnitude of the microphone pressure signal is the third step, shown in Figure 6.1c. To estimate the spectral magnitude of the microphone pressure signal at 15cm from the lips, $|\mathbf{P_{mic}}|$, $|\mathbf{V_{VGA}}|$ is multiplied by the magnitude of a neutral vocal tract transfer function $|\mathbf{T(f)}|$, followed by multiplication by the magnitude of the radiation characteristic $|\mathbf{R(f)}|$. The transfer function $|\mathbf{T(f)}|$ assumes that the vocal tract has uniform cross-sectional

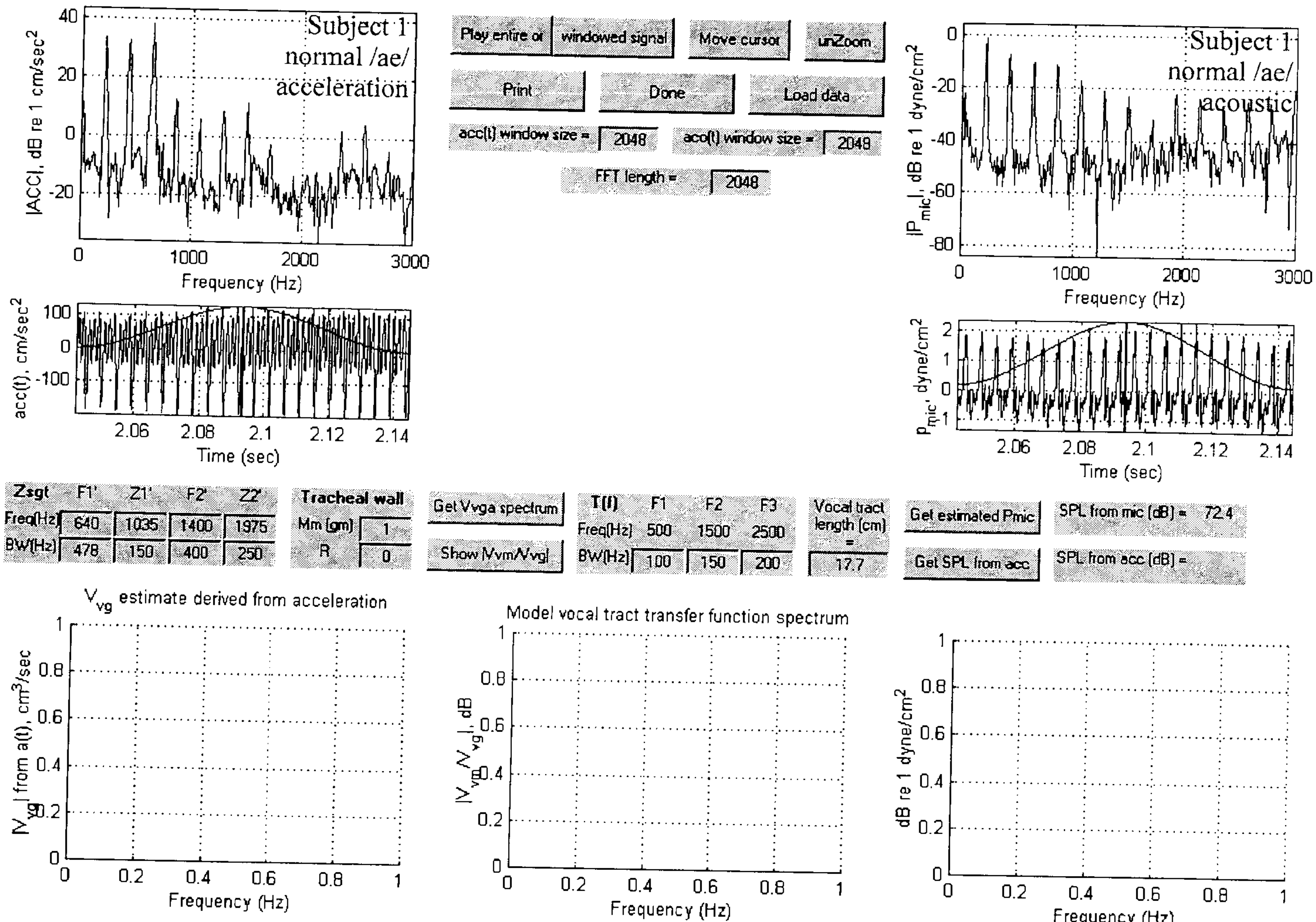


Figure 6.1a. SPL analysis screen example, step 1: select 2048-point stable segments from the acceleration and acoustic signals near the midpoint of the sustained vowel. Shown are the acceleration time signal and Hanning window used for its FFT (middle left plot), the acceleration spectral magnitude (upper left plot), the acoustic time signal and Hamming window used for its FFT (middle right plot), the acoustic spectral magnitude, and the SPL in dB (below acoustic signal).

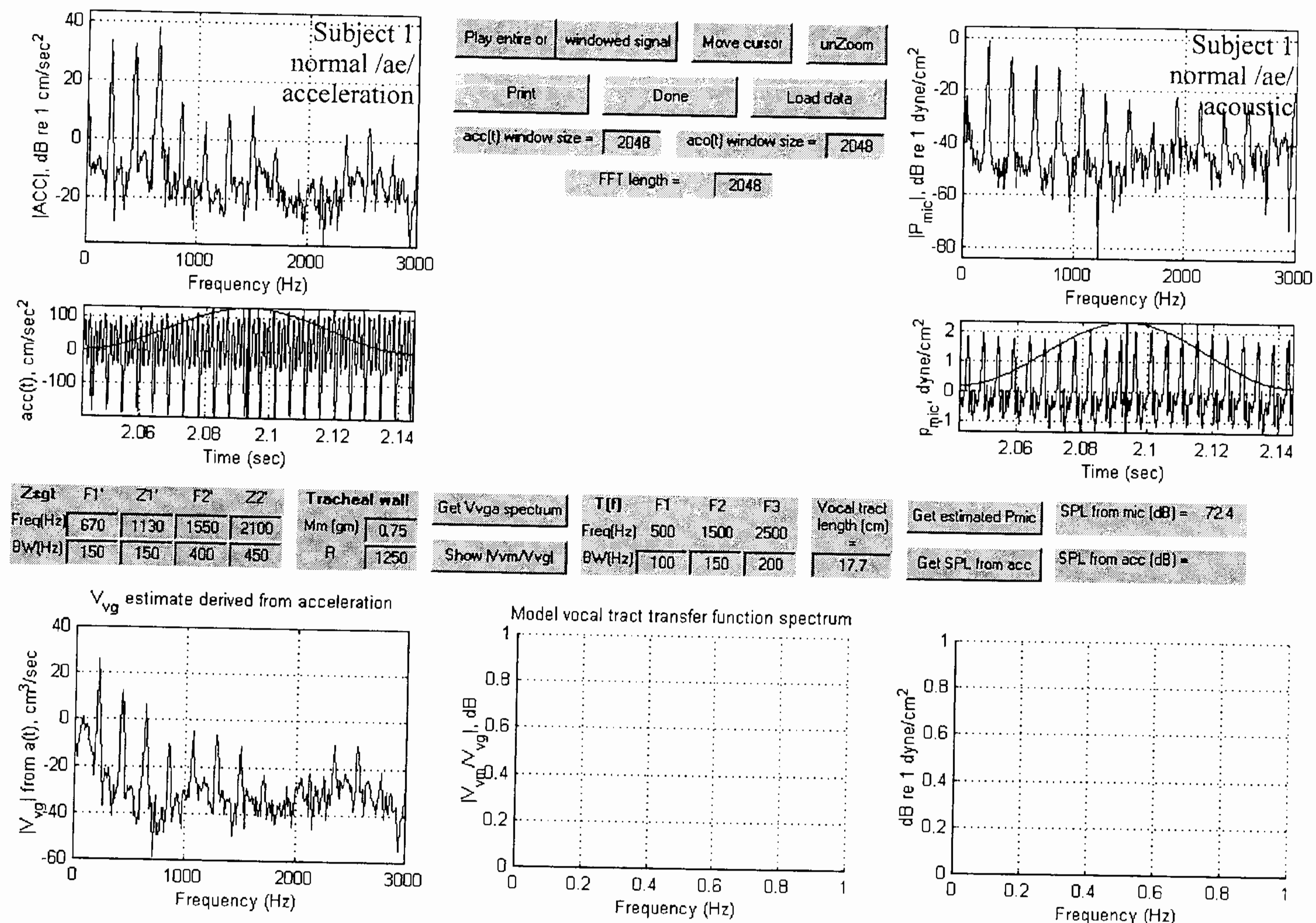


Figure 6.1b. SPL analysis screen example, step 2: the subglottal parameters are set to the values found for the vowel /ae/ at normal loudness in Experiment II - see Table 5.2. With these parameters an inverse filter is created to remove the effects of the tracheal wall and subglottal transfer impedance, and the result is the spectral magnitude (Fourier Series) of the acceleration-derived glottal volume velocity, shown in the lower left plot.

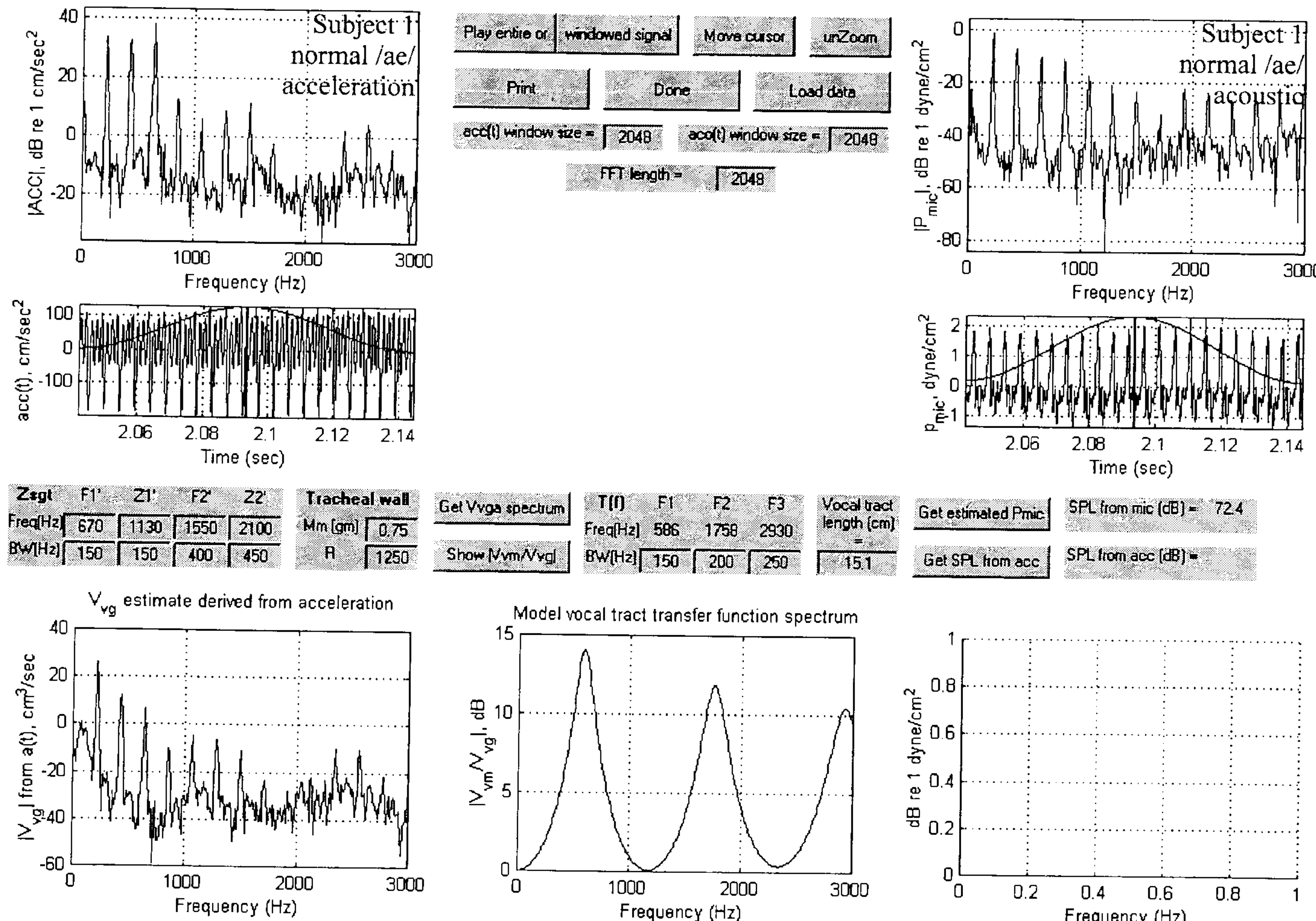


Figure 6.1c. SPL analysis screen example, step 3: set the vocal tract length to the value found for the vowel /ae/ at normal loudness in Experiment II. This sets the vocal tract formants to frequencies based on a neutral vocal tract shape according to Equation 5.2. Set the bandwidths of [F1, F2, F3] to [150, 200, 250] for females or [100, 150, 200] for males, and calculate the vocal tract transfer function magnitude, shown in the middle lower plot.

area, and that the formant frequencies $\mathbf{f_{Fn}}$ are determined solely by the vocal tract length $\mathbf{l_{VT}}$, according to Equation 5.2. The vocal tract length is fixed for each subject, across vowels, using the value determined for the /ae/ sustained vowel in normal voice from Experiment II. Further, the first three formant bandwidths are fixed at 150, 200 and 250 Hz for females and 100, 150 and 200 Hz for males respectively. The first three formant frequency and bandwidth values for each subject appear in Table 6.1 below. Higher formant bandwidths are fixed at 250 Hz. A simple source radiation characteristic magnitude $|R(f)|$ approximates the radiation from the head:

$$|R(f)| = \frac{\rho f}{2r}$$

Equation 6.1

where **r** is 15cm, the mouth-to-microphone distance, and ρ is the density of air.

Subject	1	2	3	4	5	6	7	8	9	10
f_{F1}	586	550	486	560	571	481	489	530	598	536
BW_{F1}	150	150	100	150	150	100	100	100	150	100
f_{F2}	1758	1649	1459	1680	1713	1443	1467	1590	1794	1609
BW_{F2}	200	200	150	200	200	150	150	150	200	150
f_{F3}	2930	2748	2431	2801	2855	2405	2445	2650	2990	2682
BW_{F3}	250	250	200	250	250	200	200	200	250	200

Table 6.1. Vocal tract formant frequencies and bandwidths used for Experiment III. These formant frequencies are based on the vocal tract lengths found for the vowel /ae/ at normal loudness in Experiment II, by using Equation 5.2. The bandwidths of [F1, F2, F3] are set to [150, 200, 250] for females and [100, 150, 200] for males.

Once $|P_{mic}|$ is estimated, the fourth and last step in estimating its corresponding SPL is summing the individual harmonic pressure amplitudes of the spectrum, shown in Figure 6.1d. Assuming that the $|P_{mic}|$ spectrum is a harmonic line spectrum, the overall SPL is calculated by

$$SPL(dB) = 10\log_{10}\left[\frac{P_{H1}^2 + P_{H2}^2 + P_{H3}^2 + \cdots + P_{Hn}^2}{P_{REF}^2}\right]$$

Equation 3.5

P_{H1} is the pressure amplitude of the first harmonic, given by $P_{H1} = P_{REF}10^{\frac{P_{H1}(dB)}{20}}$, where $P_{H1}(dB)$ is the magnitude of the first harmonic from the spectrum $|\mathbf{P_{mic}}|$ in dB, and the reference pressure P_{REF} equals 2×10^{-4} dyne/cm^2. The individual harmonic magnitudes and frequencies are found algorithmically. First the approximate frequency of **H1** is selected with

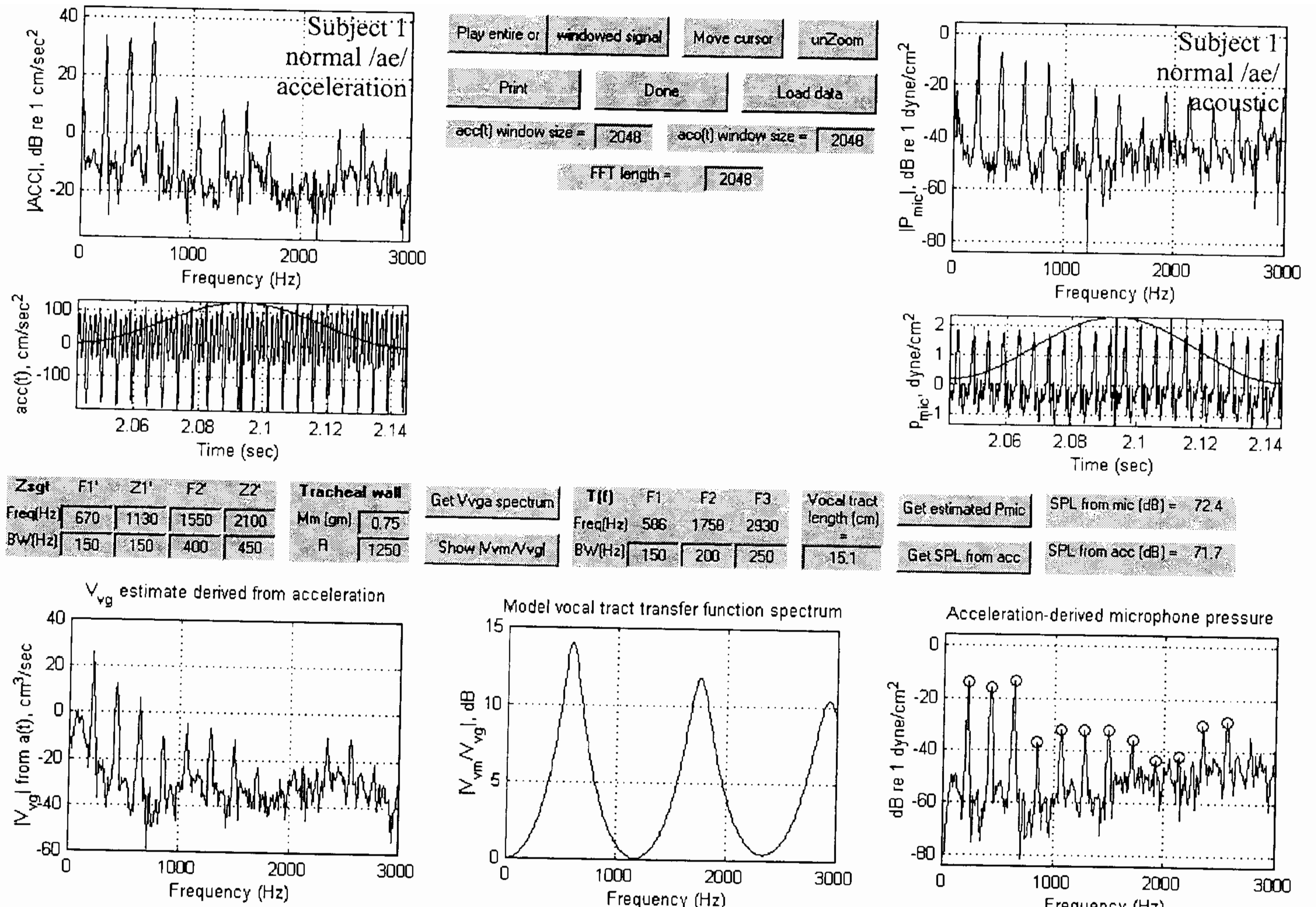


Figure 6.1d. SPL analysis screen example, step 4: multiply the glottal flow spectrum (lower left plot) by the vocal tract transfer function (lower middle plot) and the radiation characteristic (not shown) to get the acceleration-derived microphone pressure spectrum (lower right plot). From the acceleration-derived microphone pressure spectrum, sum the energy of the harmonics to get the estimate of SPL from the acceleration signal.

the cursor, and the subroutine getaccspl of program accspl1 selects the actual maximum magnitude from a 7-point (68 Hz) window centered on the cursor-selected point. The frequency of this maximum is **F0**, and the subroutine looks for the next harmonic **H2** in a 5-point (49 Hz) window centered around 2×**F0**. The subroutine now sets **F0** to half the frequency of **H2**, and the third harmonic is searched for in a 5-point window around 3×**F0**. This iterative process of modifying **F0** to $\frac{f_{H_n}}{n}$ and finding the next harmonic continues until the harmonic just below 3000 Hz is found. Results from this analysis are shown in the next section $\mathbf{SPL_A}$, the estimated SPL from the acceleration, versus $\mathbf{SPL_M}$, the measured SPL from the microphone.

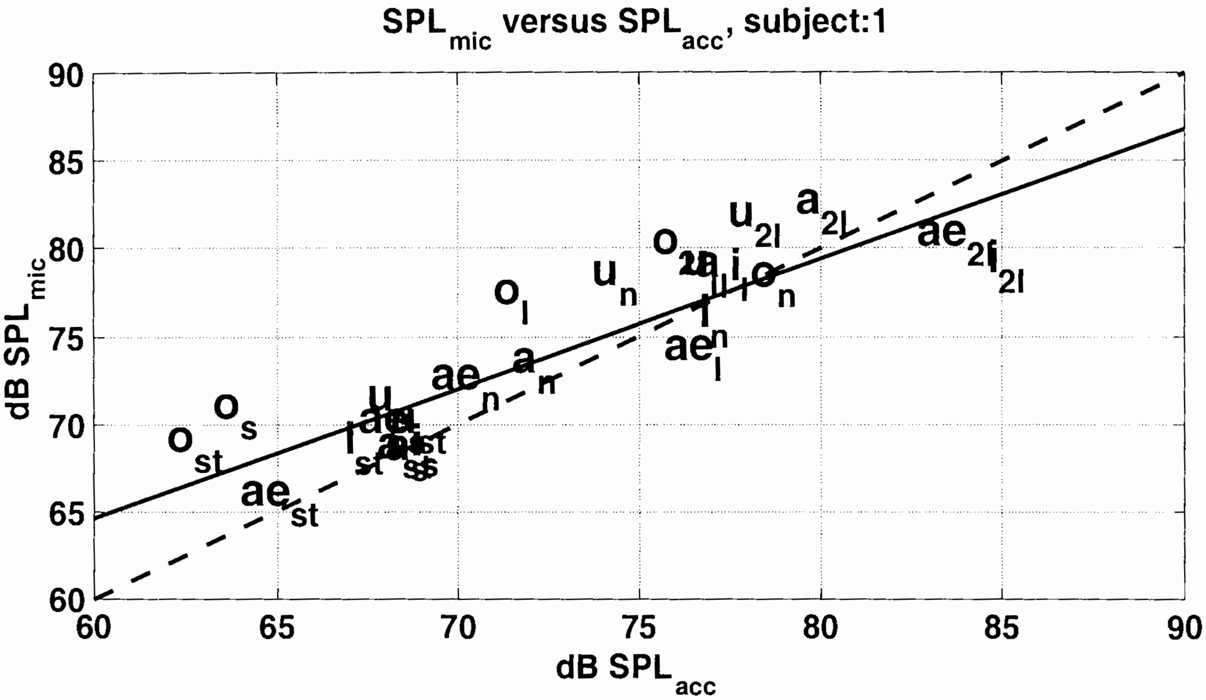
6.2 Results from Experiment III: Estimating SPL from the acceleration using the vocal system model

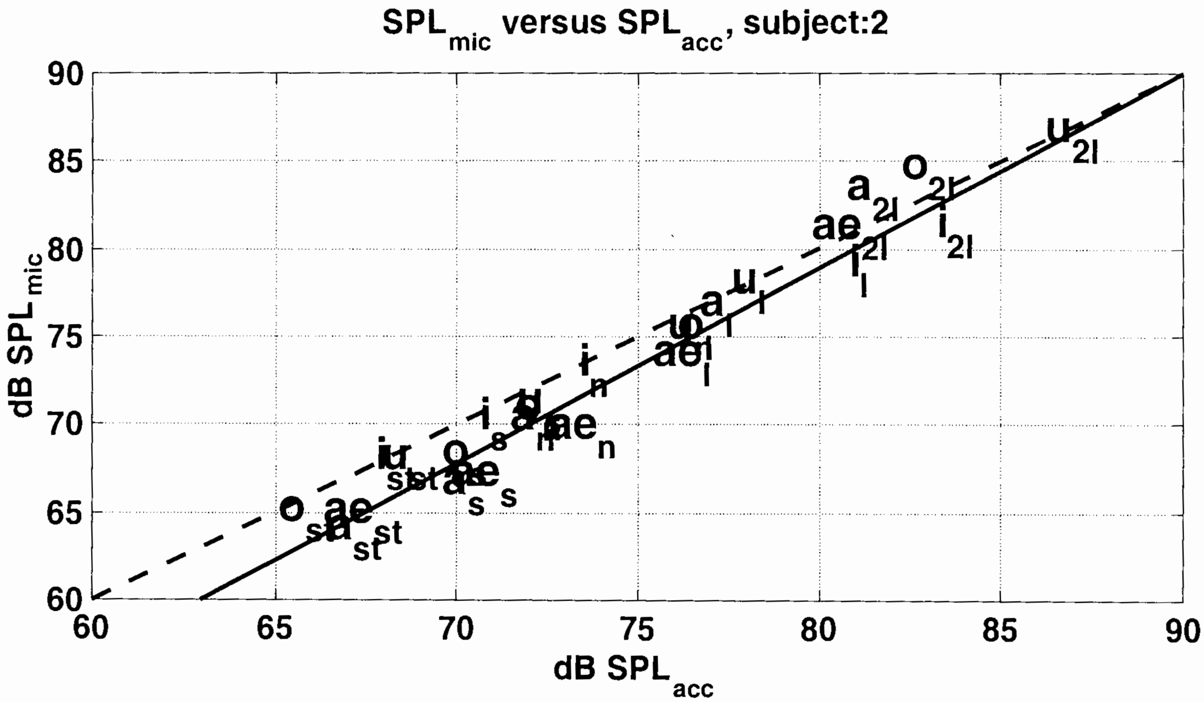
As described above in Section 6.1, the technique used to estimate the Sound Pressure Level (SPL) from the acceleration involves estimating the glottal flow, then applying that flow as input to a fixed neutral vocal tract transfer function and radiation characteristic. This technique is applied to the acceleration and acoustic signals from vocal task 7 - the five sustained vowels at five different intensities without the Rothenberg mask. The SPL measured at the microphone ($\mathbf{SPL_{mic}}$) and the estimated SPL from the acceleration ($\mathbf{SPL_{acc}}$) are plotted below for each subject in Figures 6.2 through 6.11, along with a dashed reference line of slope m=1, and a solid line-of-best-fit in the least squares sense. Data points are plotted as their corresponding vowel and intensity, as done in Section 5.2. The mean and standard deviation of the difference between the measured $\mathbf{SPL_{mic}}$ and the predicted $\mathbf{SPL_{acc}}$ (error) appear below in Table 6.2, along with the line-of-best-fit equation and its corresponding r^2 correlation coefficient.

It is interesting to note that for subjects 2 and 9, who have the most highly correlated $\mathbf{SPL_{mic}}$ and $\mathbf{SPL_{acc}}$ values, the linear fit to their data has a slope that is significantly different from one ($p = 0.05$). In addition, the mean errors here are larger than those for the MFDR predictions for subjects 2, 4, 5, 6, 7, 8, and 9, suggesting that the acceleration may provide a better measure of glottal activity (i.e., MFDR) than acoustic output. However, assuming the error follows a Gaussian distribution with zero mean, 95% of the SPL estimates (excluding subject 6) will be within ±7dB. Thus the acceleration signal has the potential of providing a reasonable SPL estimate, provided that the mean error is reduced.

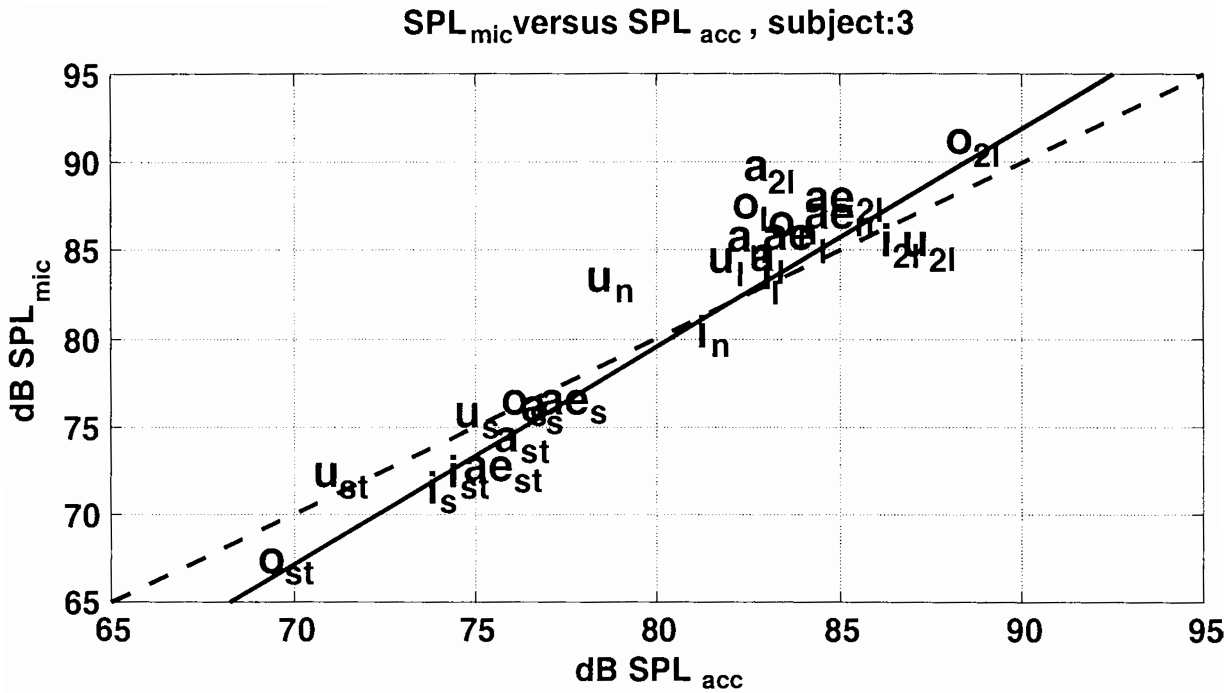
Subject	$\overline{(SPL_{mic} - SPL_{acc})}$,dB	$\sigma_{SPL_{mic}-SPL_{acc}}$,dB	Linear fit, $\hat{Y}$ =	r^2
1	1.1	2.9	0.74×**SPL$_{acc}$** + 20.3	0.77
2	-1.7	1.6	1.11×**SPL$_{acc}$** - 10.0	0.95
3	-0.2	2.6	1.24×**SPL$_{acc}$** - 19.5	0.89
4	-4.5	3.6	0.56×**SPL$_{acc}$** + 30.0	0.86
5	0.7	2.8	**0.90**×**SPL$_{acc}$** + 8.0	0.91
6	-1.5	3.3	**1.31**×**SPL$_{acc}$** - 25.4	0.67
7	-2.9	2.3	**0.91**×**SPL$_{acc}$** + 3.8	0.85
8	-2.8	2.3	**0.97**×**SPL$_{acc}$** - 0.2	0.93
9	-6.0	2.7	0.73×**SPL$_{acc}$** + 15.4	0.95
10	-1.1	3.7	**1.28**×**SPL$_{acc}$** - 23.6	0.71
1-5, 7-10	-1.9	3.5	-	-

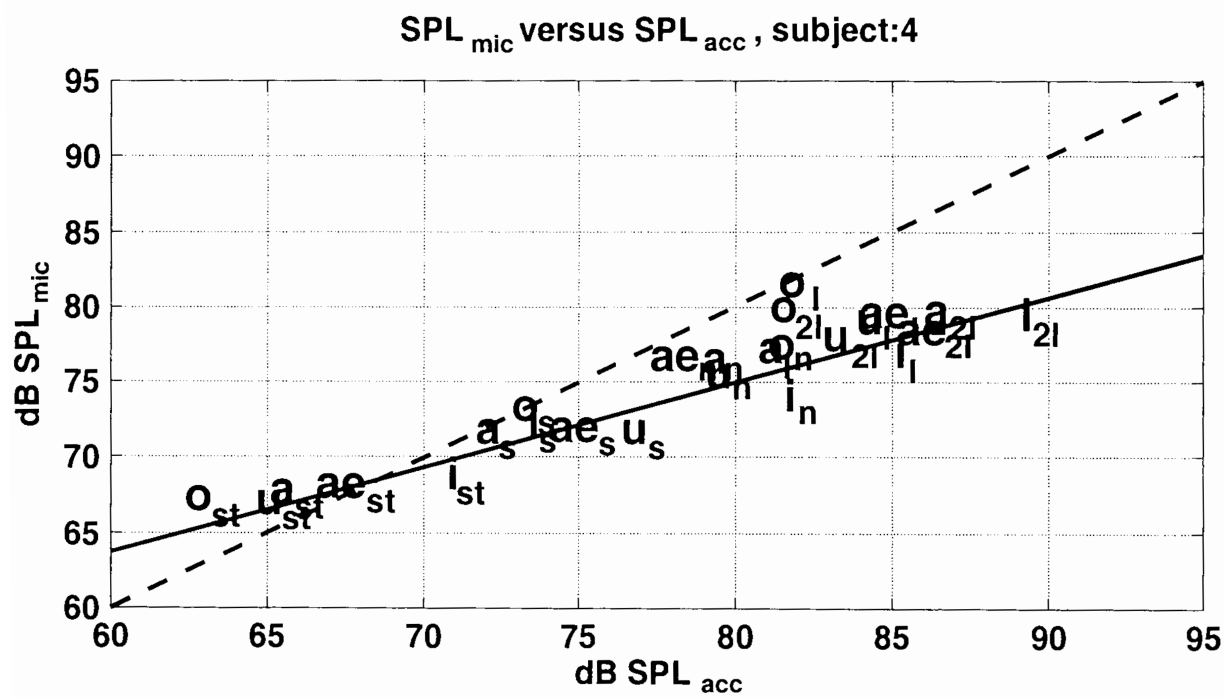
Table 6.2. Mean error, error standard deviation, linear fit equations, and correlation coefficients for the actual SPL (SPL_{mic}) versus the model-predicted SPL (SPL_{acc}). Linear fit slope values that appear BOLD indicate that the slope was not significantly different from m = 1 (p = 0.05). The last row gives the overall mean error and error standard deviation, for all subjects excluding subject 6.



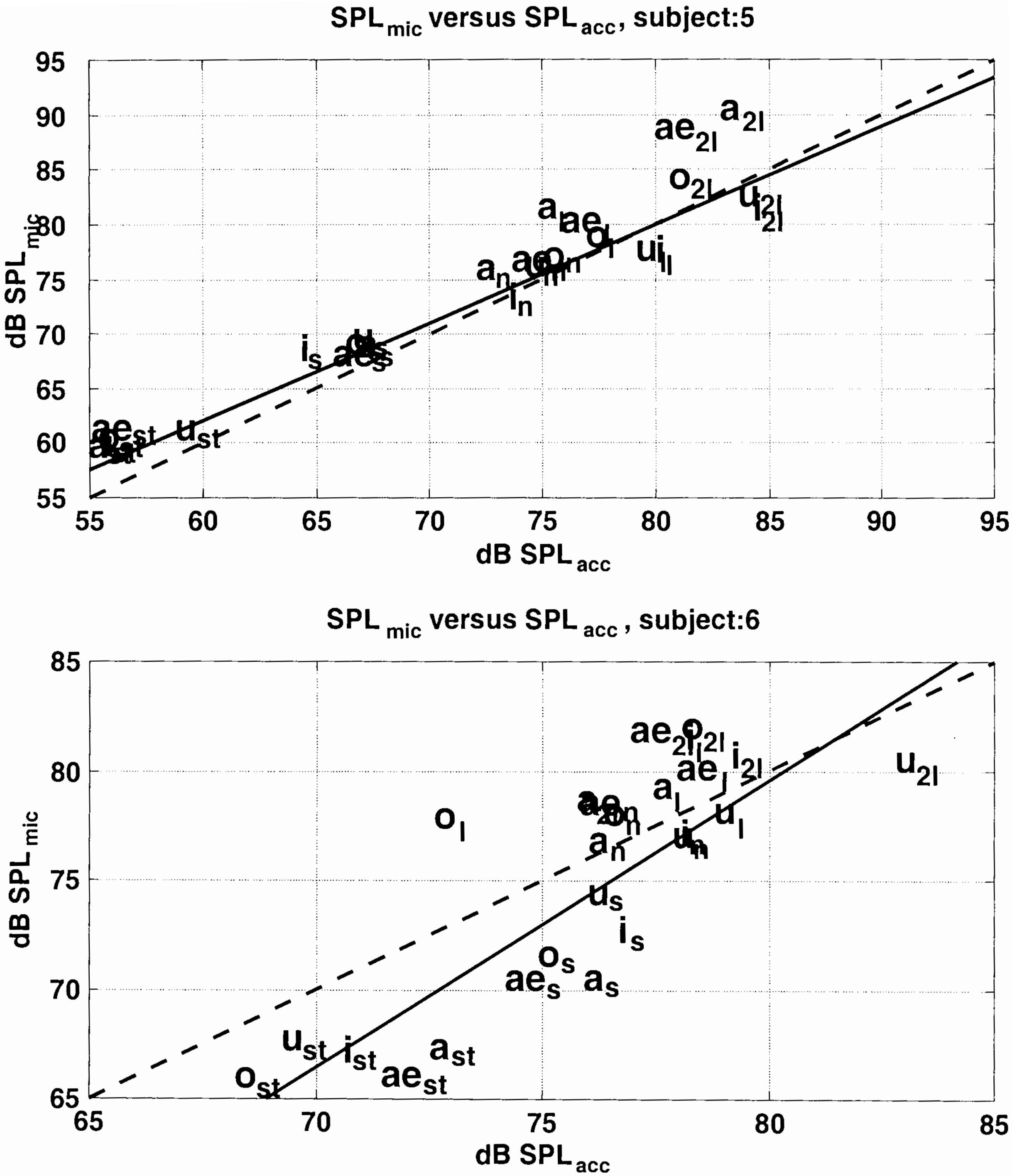


Figures 6.2 (upper) and 6.3 (lower). Actual SPL (SPL_{mic}) versus estimated SPL from the acceleration (SPL_{acc}) for subjects 1 and 2, over five vowels (/ae/, /a/, /i/, /o/, /u/) at five different intensities (softest, softer than normal, normal, louder than normal, twice as loud as normal). The solid line is a least-squares linear fit, and the dashed line is a reference with slope m = 1 and intercept b = 0.

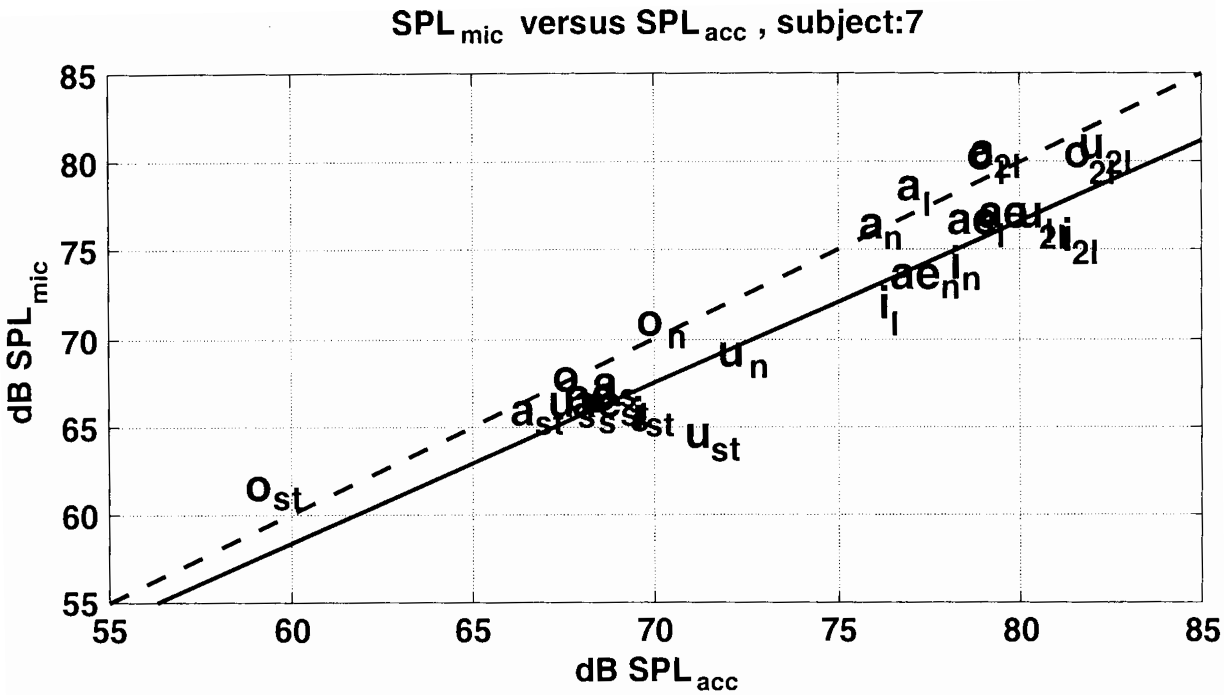


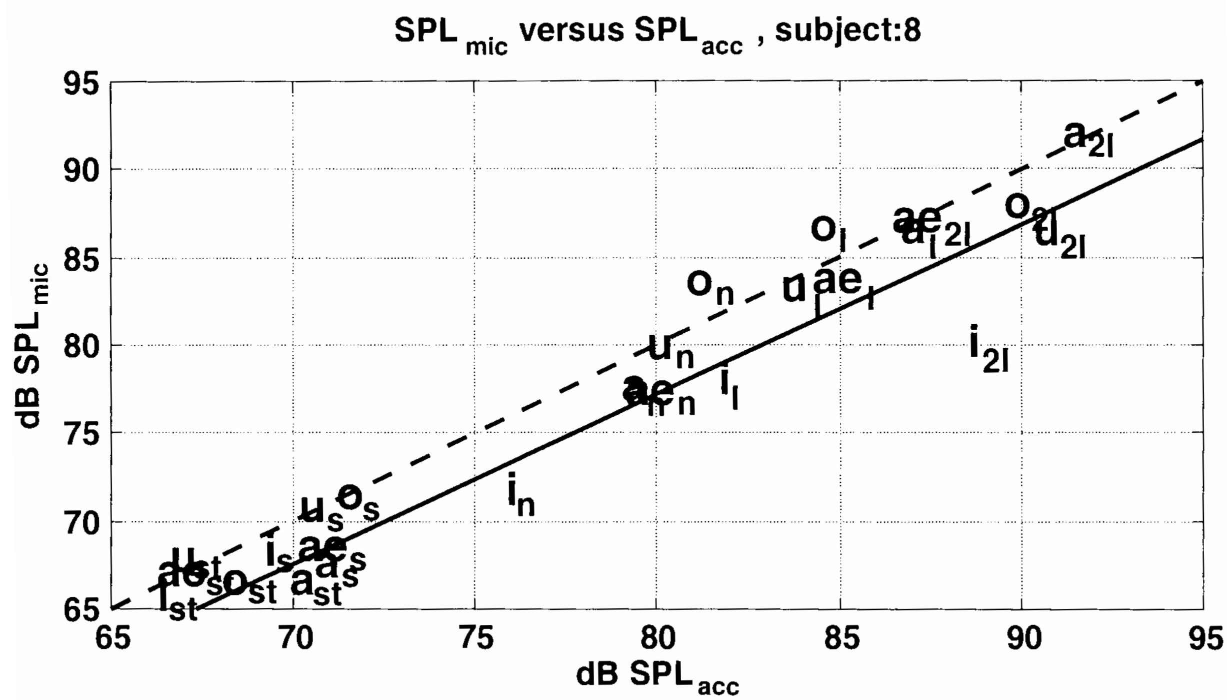


Figures 6.4 (upper) and 6.5 (lower). Actual SPL (SPL_{mic}) versus estimated SPL from the acceleration (SPL_{acc}) for subjects 3 and 4, over five vowels (/ae/, /a/, /i/, /o/, /u/) at five different intensities (softest, softer than normal, normal, louder than normal, twice as loud as normal). The solid line is a least-squares linear fit, and the dashed line is a reference with slope m = 1 and intercept b = 0.

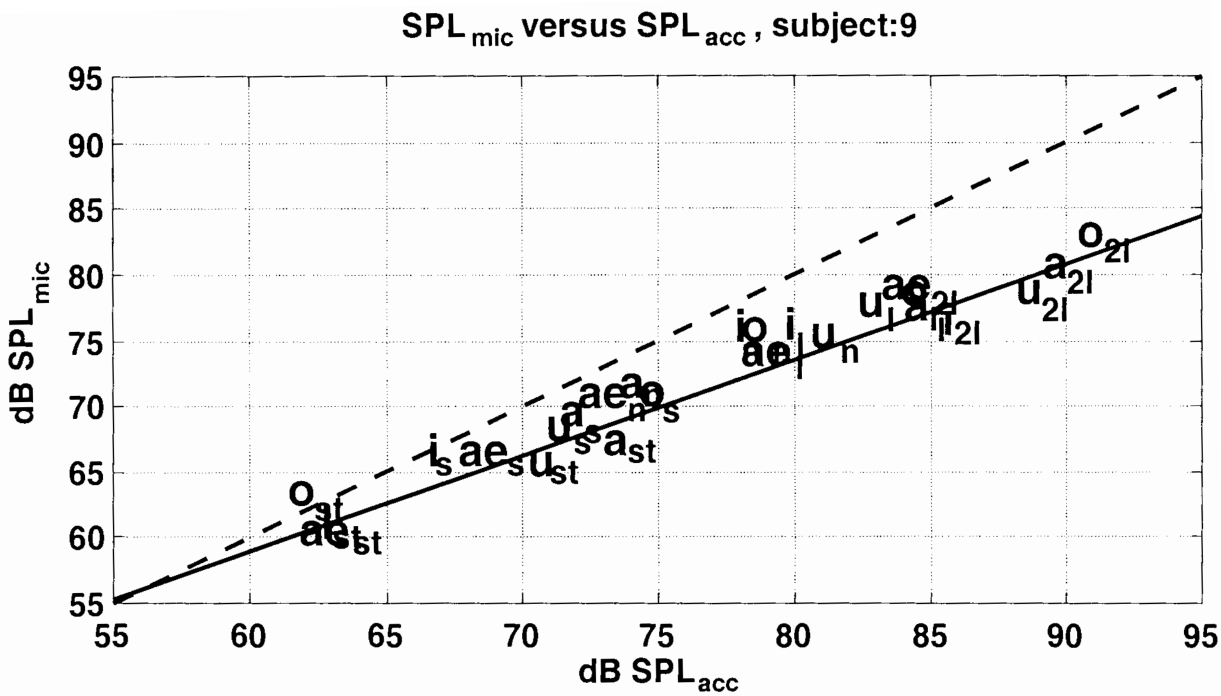


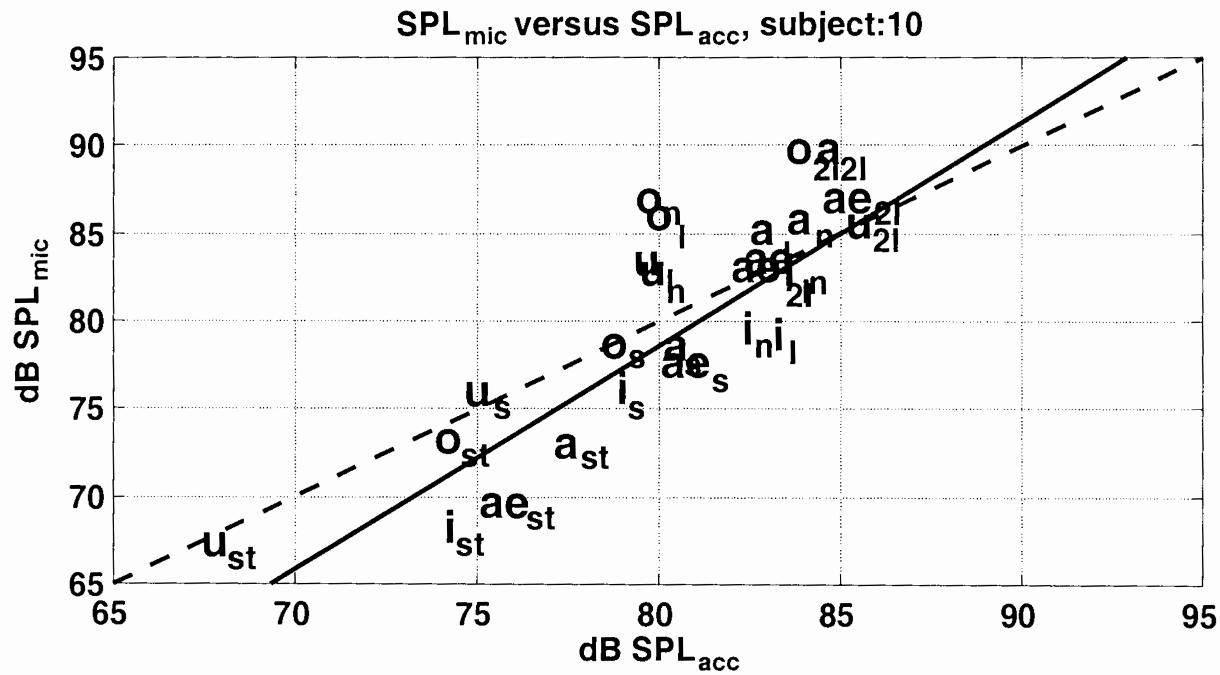
Figures 6.6 (upper) and 6.7 (lower). Actual SPL (SPL_{mic}) versus estimated SPL from the acceleration (SPL_{acc}) for subjects 5 and 6, over five vowels (/ae/, /a/, /i/, /o/, /u/) at five different intensities (softest, softer than normal, normal, louder than normal, twice as loud as normal). The solid line is a least-squares linear fit, and the dashed line is a reference with slope m = 1 and intercept b = 0.





Figures 6.8 (upper) and 6.9 (lower). Actual SPL (SPL_{mic}) versus estimated SPL from the acceleration (SPL_{acc}) for subjects 7 and 8, over five vowels (/ae/, /a/, /i/, /o/, /u/) at five different intensities (softest, softer than normal, normal, louder than normal, twice as loud as normal). The solid line is a least-squares linear fit, and the dashed line is a reference with slope m = 1 and intercept b = 0.





Figures 6.10 (upper) and 6.11 (lower). Actual SPL (SPL_{mic}) versus estimated SPL from the acceleration (SPL_{acc}) for subjects 9 and 10, over five vowels (/ae/, /a/, /i/, /o/, /u/) at five different intensities (softest, softer than normal, normal, louder than normal, twice as loud as normal). The solid line is a least-squares linear fit, and the dashed line is a reference with slope m = 1 and intercept b = 0.

6.3 Discussion of estimating SPL from the acceleration

Again keeping in mind the long-term goal of calibrating the acceleration signal for a particular subject and then using that signal alone to extract information like MFDR and SPL, the subglottal system parameters listed in Table 5.2 are used with the vocal system model to give the acceleration-derived SPL, $\mathbf{SPL_{acc}}$. The vocal tract length is also fixed to the value found in Experiment II for the vowel /ae/ in normal voice, and with that length the vocal tract formants are fixed based on Equation 5.2. The bandwidths of the vocal tract formants [**F1**, **F2**, **F3**] are set to [150, 200, 250] Hz for females and [100, 150, 200] Hz for males. Like Experiment II, the values used for $\mathbf{BW_{F1}}$ are in the range given by Hanson & Chuang (1999) and Hanson (1997), and about twice as large as those given by Fant (1962), so they seem appropriate given the analysis occurs over several fundamental periods of phonation. As in Experiment II, variation in the tracheal wall parameter $\mathbf{M_{TW}}$ shifts the spectrum vertically, which is analogous to changing the error $\mathbf{SPL_{mic}}$-$\mathbf{SPL_{acc}}$. However, $\mathbf{M_{TW}}$ stayed fixed within subjects in reference to its value from Experiment II, so the error was not reduced by altering $\mathbf{M_{TW}}$ in this experiment.

Using the vocal system model, the data in Figures 6.2 through 6.11 show the potential of the acceleration signal to provide SPL estimates within a 95% confidence window of ±7dB. For a comparison to some other available data on SPL, see Table 6.3 below.

Sex, Loudness	$\overline{SPL_{acc}} \pm \sigma_{SPL_{acc}}$	$\overline{SPL_{mic}} \pm \sigma_{SPL_{mic}}$	Holmberg et al. (1994)	Holmberg et al. (1995)
female, soft	71.1 ± 3.3	68.4 ± 2.3	68.5 ± 3.3	N.A.
female, normal	75.2 ± 3.3	73.0 ± 3.1	74.0 ± 3.3	73.2 ± 3.7
female, loud	79.8 ± 3.9	76.1 ± 3.1	80.8 ± 3.8	82.3 ± 4.0
male, soft	75.7 ± 5.0	71.7 ± 4.8	71.2 ± 3.3	N.A.
male, normal	81.6 ± 3.8	80.3 ± 6.3	77.8 ± 4.4	N.A.
male, loud	82.9 ± 3.1	81.6 ± 3.6	85.6 ± 4.6	N.A.

Table 6.3. Comparison of this study's $\mathbf{SPL_{acc}}$ and $\mathbf{SPL_{mic}}$ mean and standard deviation values to two previously published reports. All values are for the vowel /ae/ only, and are in dB re $\mathbf{2\times10^{-4}}$ dyne/cm^2.

Overall, the true sound pressures produced by the subjects in this experiment are similar to those reported in Holmberg et al. (1995) and Holmberg et al. (1994), more so for the soft and normal conditions. On average, $\mathbf{SPL_{acc}}$ is greater than the $\mathbf{SPL_{mic}}$ by 1.3-4 dB. As in Experiment II, the errors in calculating $\mathbf{SPL_{acc}}$ may be from a few sources.

First, the tracheal wall parameter $\mathbf{M_{TW}}$ may not be correct. Although it was optimized for each subject in Experiment II, that procedure for choosing $\mathbf{M_{TW}}$ does not ensure that it represents the true moving mass of the tracheal wall, and thus may not be appropriate for this experiment. Some subjects whose $\mathbf{SPL_{acc}}$ and $\mathbf{SPL_{mic}}$ data is highly correlated are likely affected by errors in $\mathbf{M_{TW}}$. Specifically, subjects 2 (Figure 6.3), 7 (Figure 6.8) and 8 (Figure 6.9) could benefit from a change in $\mathbf{M_{TW}}$, which would eliminate their mean error and not affect their standard deviation. For example, changing $\mathbf{M_{TW}}$ for subject 2 from 1.1 grams to 0.7 grams would reduce her mean error to practically zero.

Second, the fixed vocal tract aspect of the model may be a source of error, since that too is determined from the results of Experiment II. However, the assumption of a neutral vocal tract does not appear to give any bias towards any particular vowel or vowel class; Figures 6.2 through 6.11 show that within subjects and across vowels there is not any obvious error due to the vowel being spoken. To test if the fixed vocal tract is producing a significant part of the error, a couple of subjects' individual sustained vowels are processed with a modified version of <u>accspl1</u> named <u>accspl2</u> (see Appendix B), which allows variable **F1**, **F2**, and **F3**. The resulting analysis screens for subject 3 sustaining /u/ and subject 10 sustaining /o/ at normal loudness are shown in Figures 6.12 and 6.13 respectively. These two examples originally produced $\mathbf{SPL_{acc}}$ = 79.6 and 80.9 dB respectively, and with the revised vocal tract transfer function **T(f)** they produce $\mathbf{SPL_{acc}}$ = 78.3 and 81.5 respectively. Because subject three's result is only marginally worse and subject ten's result is marginally better in terms of the error $\mathbf{SPL_{mic}}$-$\mathbf{SPL_{acc}}$, it appears that the fixed vocal tract transfer function is not responsible for large errors between $\mathbf{SPL_{acc}}$ and the true SPL. However, the hint of a major source of error can be seen in the acoustic and acceleration spectra from subject 3 in Figure 6.12.

Notice the prominent peak around 2100 Hz in both the acoustic and acceleration signals of Figure 6.12. This is likely due to **F3**, given that subject 3 is a male producing the vowel /u/. It may be due to the subglottal formant **F3′**, which is usually around 2200 Hz, but inspection of the acceleration spectrum in Figure 6.12 reveals two peaks above 2kHz, one around 2100 Hz and one around 2300 Hz. The peak around 2300 Hz is likely **F3′** rather than **F3**. Subject 3 must have a glottal chink that provides acoustic coupling between the supra- and subglottal systems, so that **F3** shows prominently in the acceleration spectrum. Using the vocal system model to predict SPL using this acceleration signal compounds the problem. The tracheal wall and subglottal transfer impedance inverse filters do not remove the **F3** component from the acceleration signal, and the vocal tract transfer function **T(f)** boosts it to a greater amplitude than the acoustic signal's **F3** – compare the upper right and lower right plots of Figure 6.12. In this case, the error due to **F3** does not contribute significantly to $\mathbf{SPL_{acc}}$ since the acceleration-derived speech spectrum's **F3** peak is about 10 dB down from the greatest spectral peak.

This problem of glottal coupling introducing vocal tract formants into the acceleration signal does produce significant errors for other subjects. For example, see Figure 6.14, which shows the SPL analysis screen for subject 4 sustaining the vowel /ae/ louder than normal. Her acceleration spectrum in the upper left of the screen shows a prominent peak around 1800 Hz, which is likely due to **F2**. The inverse filter created from the subglottal transfer impedance has a pole at $\mathbf{f_{Z2'}}$ = 1800 Hz, thus the ninth harmonic (**H9**) is boosted from this filtering procedure. Then, the vocal tract transfer function has a pole at 1680 Hz for **F2**, further boosting **H9**. In finally calculating $\mathbf{SPL_{acc}}$, **H9** dominates the entire spectrum and contributes much of the error. As suggested in Section 5.3, one method for improving such results is to adapt the vocal system model to include coupling between the supra- and subglottal systems, which may reduce the effects of vocal tract formants on the acceleration and lead to a more accurate estimation of the glottal waveform.

In some cases, the $\mathbf{SPL_{acc}}$ estimate changes inversely with the true SPL. For two examples of this, see Figure 6.2 where subject 1 produces a twice-as-loud-as-normal /o/ at a true 2 dB higher than the normal /o/, but the $\mathbf{SPL_{acc}}$ decreases by 3 dB; and Figure 6.4 where subject 3 produces a twice-as-loud-as-normal /a/ a true 5 dB higher than his louder-than-normal /a/, but the $\mathbf{SPL_{acc}}$ decreases by 0.4 dB. The error shown by subject 1 results from at least two problems; see Figures 6.15 and 6.16 for more details with this explanation. First, the fixed neutral vocal tract formant frequencies are clearly not appropriate for a twice-as-loud-as-normal vowel /o/. The acoustic spectrum in the upper right corner of Figure 6.15 shows an $\mathbf{f_{F1}}$ of about 420 Hz and an $\mathbf{f_{F2}}$ of about 1050 Hz, in comparison to the neutral vocal tract values of 586 and 1758 Hz. Altering these improves the magnitudes of *H2* and *H5* in the acceleration-derived microphone pressure spectrum so they more closely approximate those in the true microphone spectrum, and results in a 1 dB increase of $\mathbf{SPL_{acc}}$. Second, the magnitude of **H1** in the acceleration-derived microphone pressure spectrum is lower than in the true microphone spectrum by a few dB, and changing $\mathbf{f_{F1}}$ and $\mathbf{f_{F2}}$ will not have an appreciable effect on it. The tracheal wall parameters $\mathbf{M_{TW}}$ and $\mathbf{R_{TW}}$ have the most significant effect on **H1** in the acceleration-derived microphone pressure spectrum, so these values may also be erroneous. Increasing $\mathbf{R_{TW}}$ to 1600 grams/second raises **H1** and brings $\mathbf{SPL_{acc}}$ and $\mathbf{SPL_{mic}}$ into agreement – see Figure 6.16.

The other example of $\mathbf{SPL_{acc}}$ changing inversely with the true SPL is between subject three's louder-than-normal /a/ and his twice-as-loud-as-normal /a/. Like subject 1, the two likely sources of error are in the neutral vocal tract and the tracheal wall parameters. Altering **F1** and **F2** to 720 and 1200 Hz respectively increases $\mathbf{SPL_{acc}}$ by 3.2 dB to 86.7 dB. As mentioned in Section 5.3, one possible source of error is a dynamically changing value of $\mathbf{R_{TW}}$. At the loudest intensities, perhaps the large subglottal pressure increases $\mathbf{R_{TW}}$. Increasing $\mathbf{R_{TW}}$ from 0 to 2000 grams/second for subject 3 increases $\mathbf{SPL_{acc}}$ an additional 2.6 dB, bringing $\mathbf{SPL_{acc}}$ and $\mathbf{SPL_{mic}}$ to within 0.1 dB of each other.

One other source of error is the subglottal transfer impedance model. This model is limited in frequency to the first two subglottal poles and zeros, **F1′**, **Z1′**, **F2′**, and **Z2′**. For most subjects, this limitation is adequate because the large spectral slope of the acceleration signal results in the energy above **Z2′** being below the noise floor. However, for some subjects, this limitation leads to errors in $\mathbf{SPL_{acc}}$. Figure 6.17 shows subject 7 sustaining the vowel /u/ at twice-as-loud-as-normal loudness. Note the prominent peak around 2300 Hz in the acceleration spectrum (upper left plot), which is likely the third subglottal formant **F3′**. This peak is not removed by the subglottal transfer impedance inverse filtering, and is further boosted by the neutral vocal tract **F3** at 2445 Hz. The resulting peak in the acceleration-derived microphone pressure spectrum (lower right plot) is a few dB down from the highest peak, so its contribution to the error in $\mathbf{SPL_{acc}}$ is probably small compared to the sources of error described above.

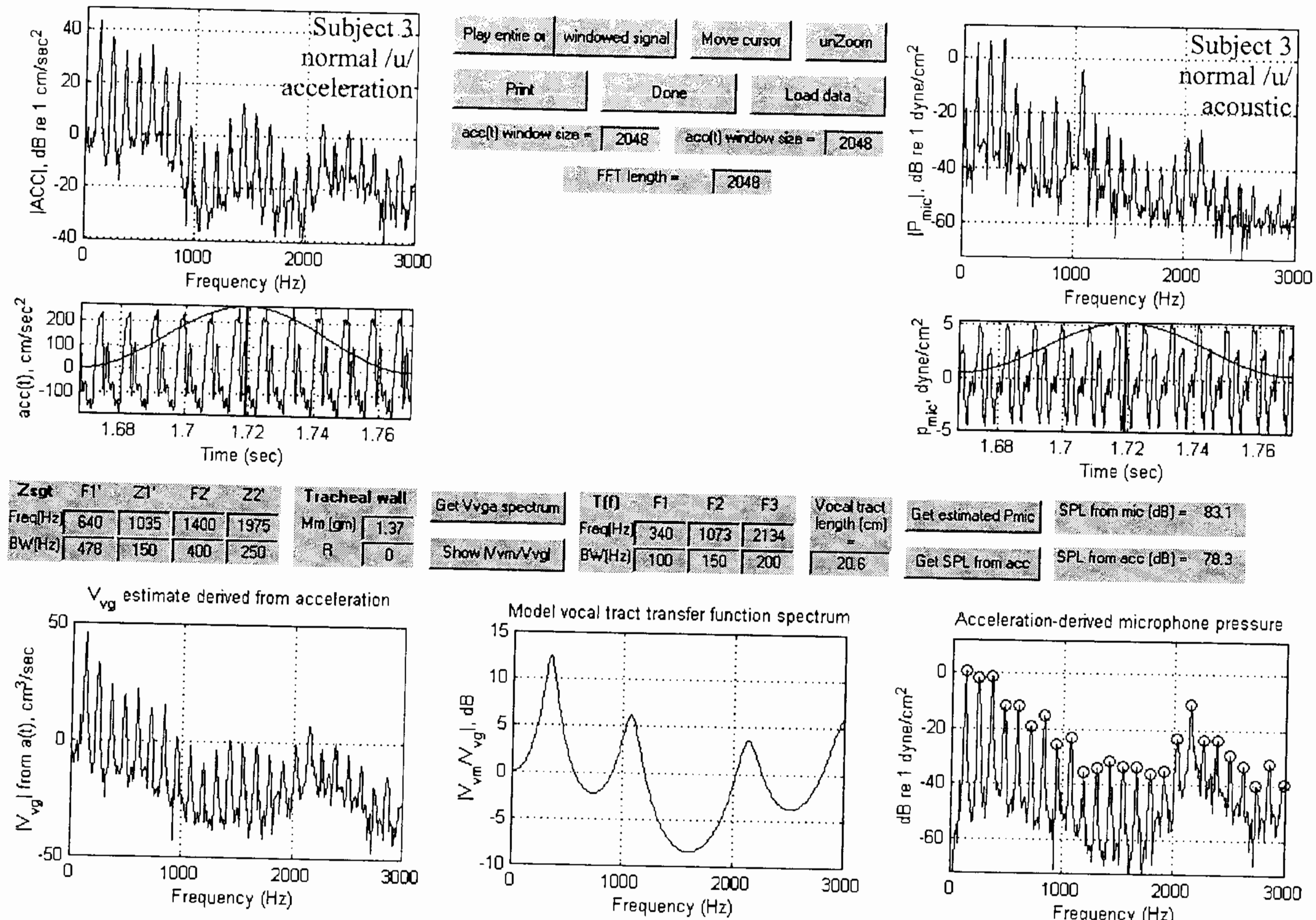


Figure 6.12. SPL analysis screen for subject 3, vowel /u/ at normal loudness. For this analysis, the frequencies of F1, F2, and F3 are allowed to vary, and are adjusted according to the peaks in the acoustic spectrum (upper right plot). Note that without the neutral vocal tract restriction, the SPL_{acc} estimate is 4.8 dB less than SPL_{mic}, and 1.3 dB less than the original SPL_{acc} estimate of 79.6 dB SPL.

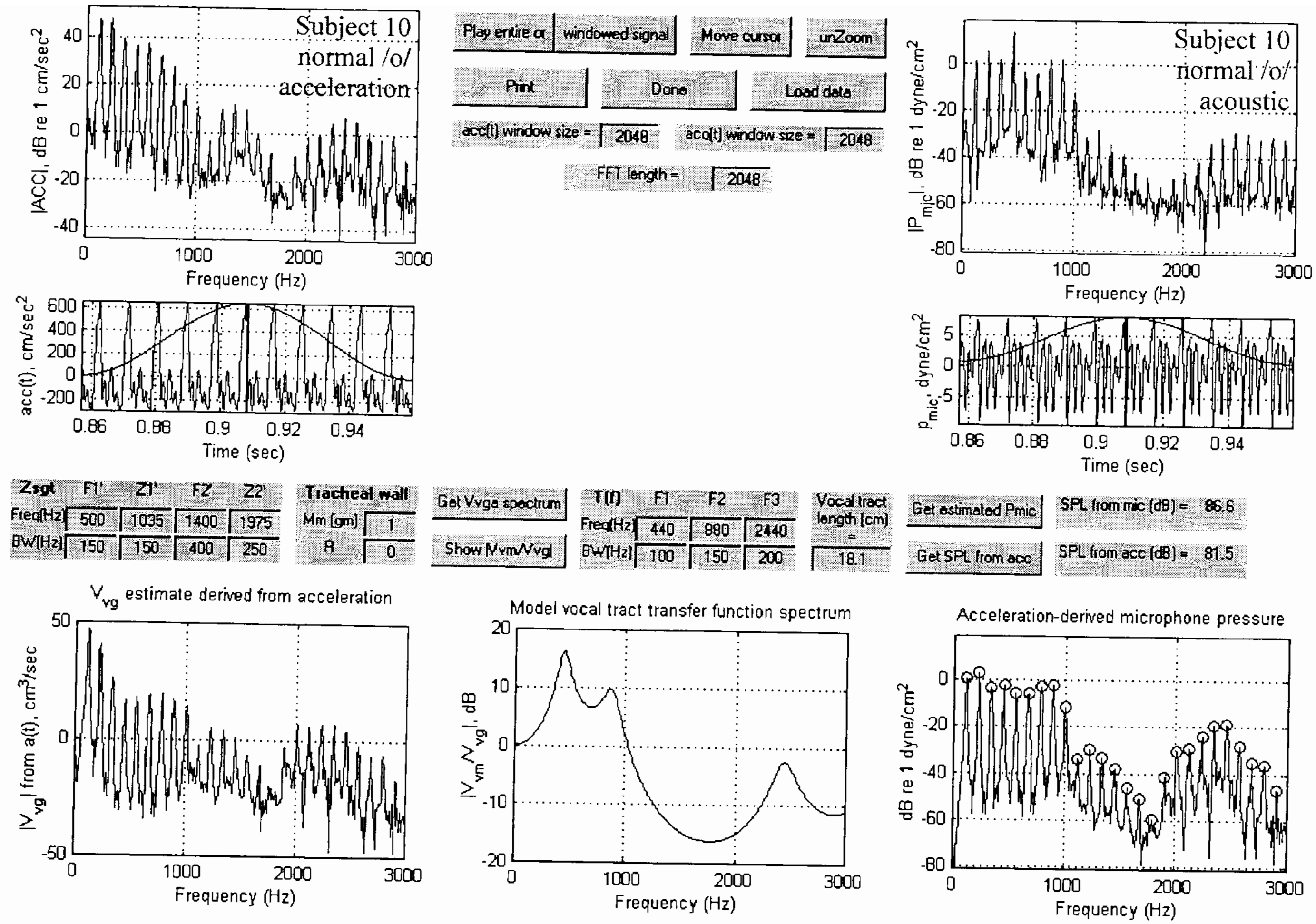


Figure 6.13. SPL analysis screen for subject 10, vowel /o/ at normal loudness. For this analysis, the frequencies of F1, F2, and F3 are allowed to vary, and are adjusted according to the peaks in the acoustic spectrum (upper right plot). Note that without the neutral vocal tract restriction, the SPL_{acc} estimate is 5.1 dB less than SPL_{mic}, and 0.6 dB more than the original SPL_{acc} estimate of 80.9 dB SPL.

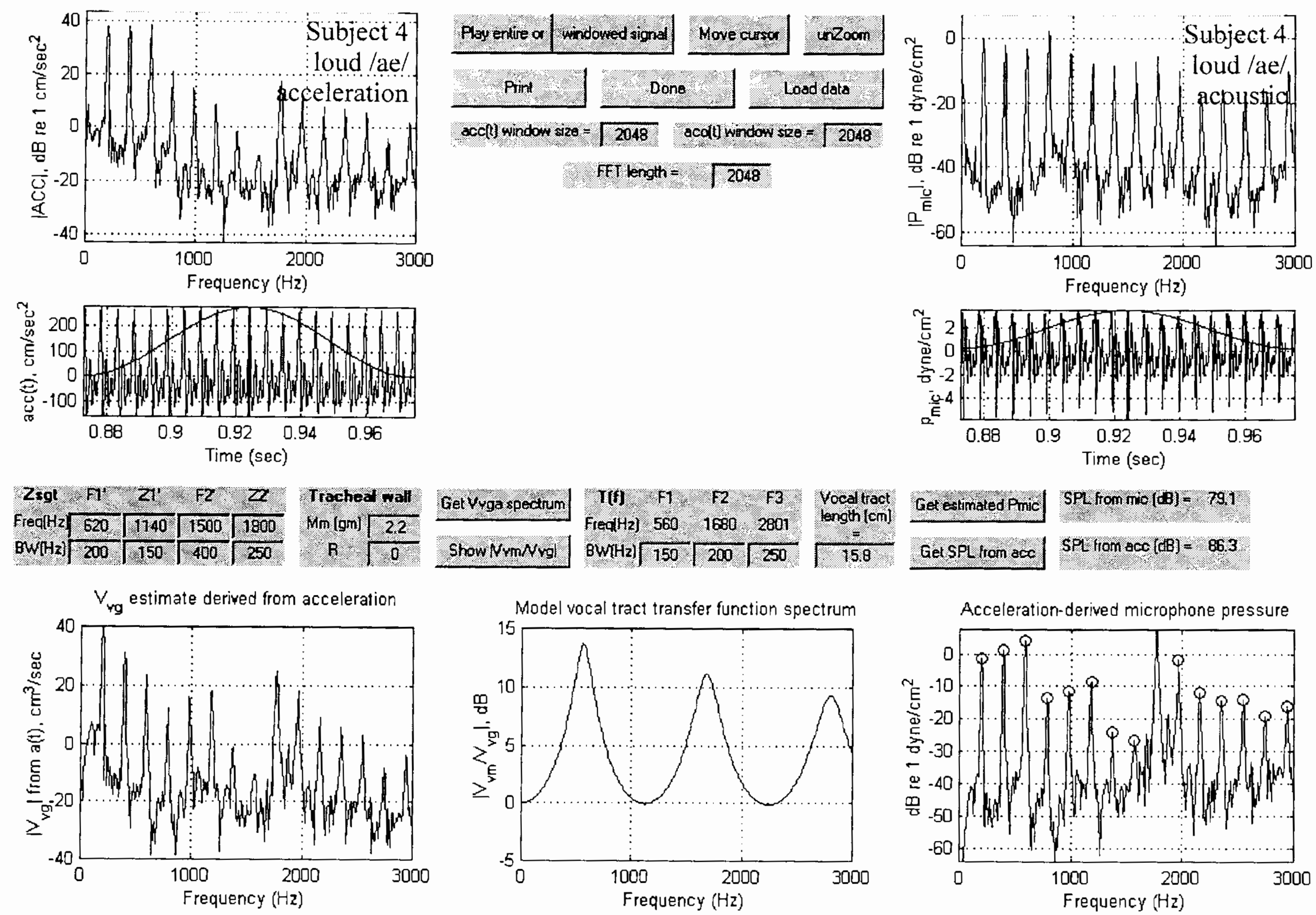


Figure 6.14. SPL analysis screen example for subject 4 sustaining the vowel /ae/ louder than normal. Note the peak in the acceleration spectrum (upper left plot) around 1800 Hz, which is likely due to the second vocal tract formant. This peak is further amplified by Z2' in the subglottal transfer impedance and F2 in the vocal tract transfer function. The result is shown in the lower right plot, where the ninth harmonic dominates the entire spectrum, and thus the estimate SPL_{acc}.

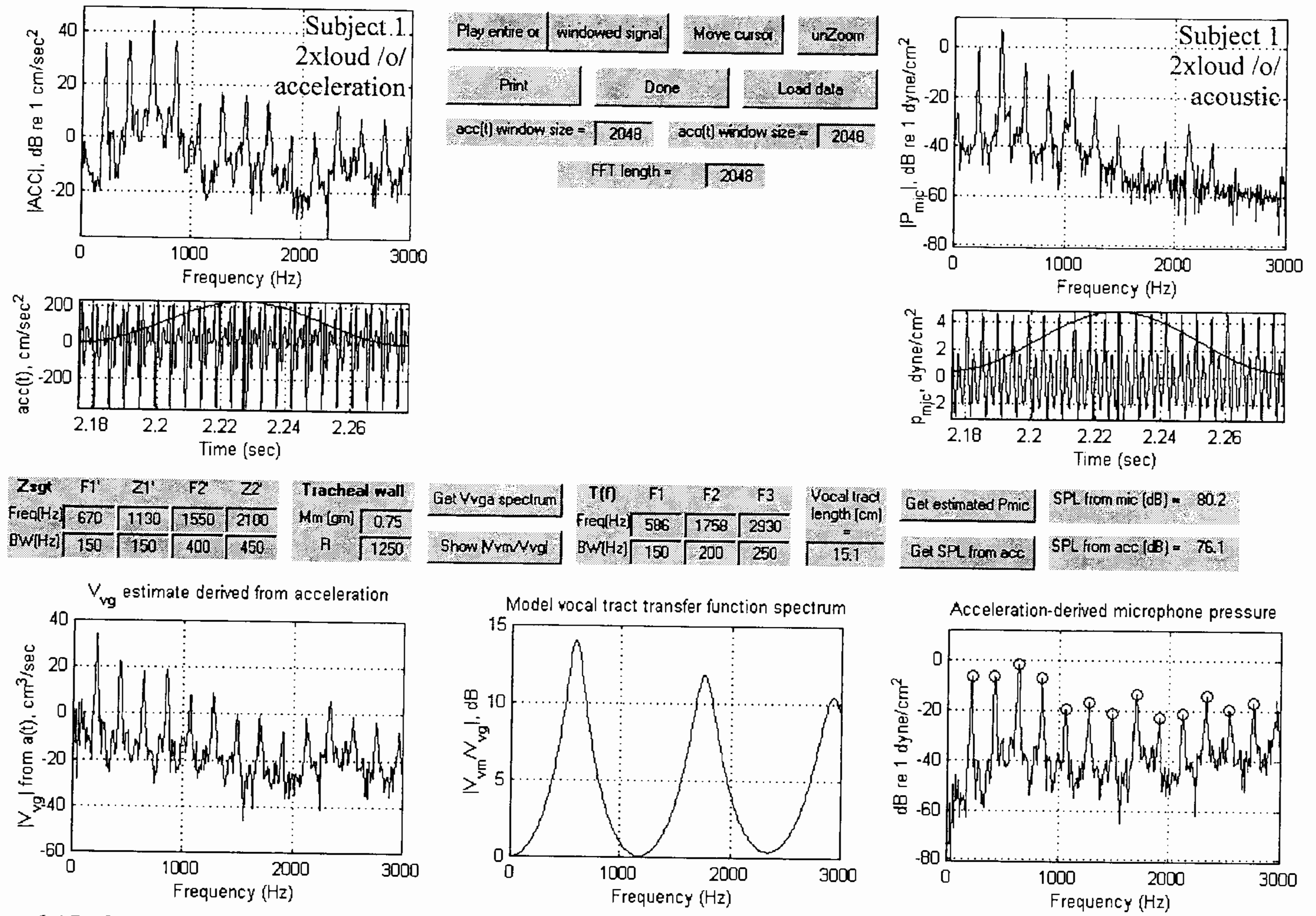


Figure 6.15. SPL analysis screen example from subject 1 sustaining the vowel /o/ twice as loud as normal. Compared to this subject's /o/ at normal loudness, the SPL_{acc} increased by 2 dB but the SPL_{mic} decreased by 3 dB. Two sources likely contribute to this error: 1) F1 and F2 of the neutral vocal tract, since those peaks are not consistent between the upper right and lower right spectra; and 2) the tracheal wall parameters, since H1 is also not consistent between those spectra.

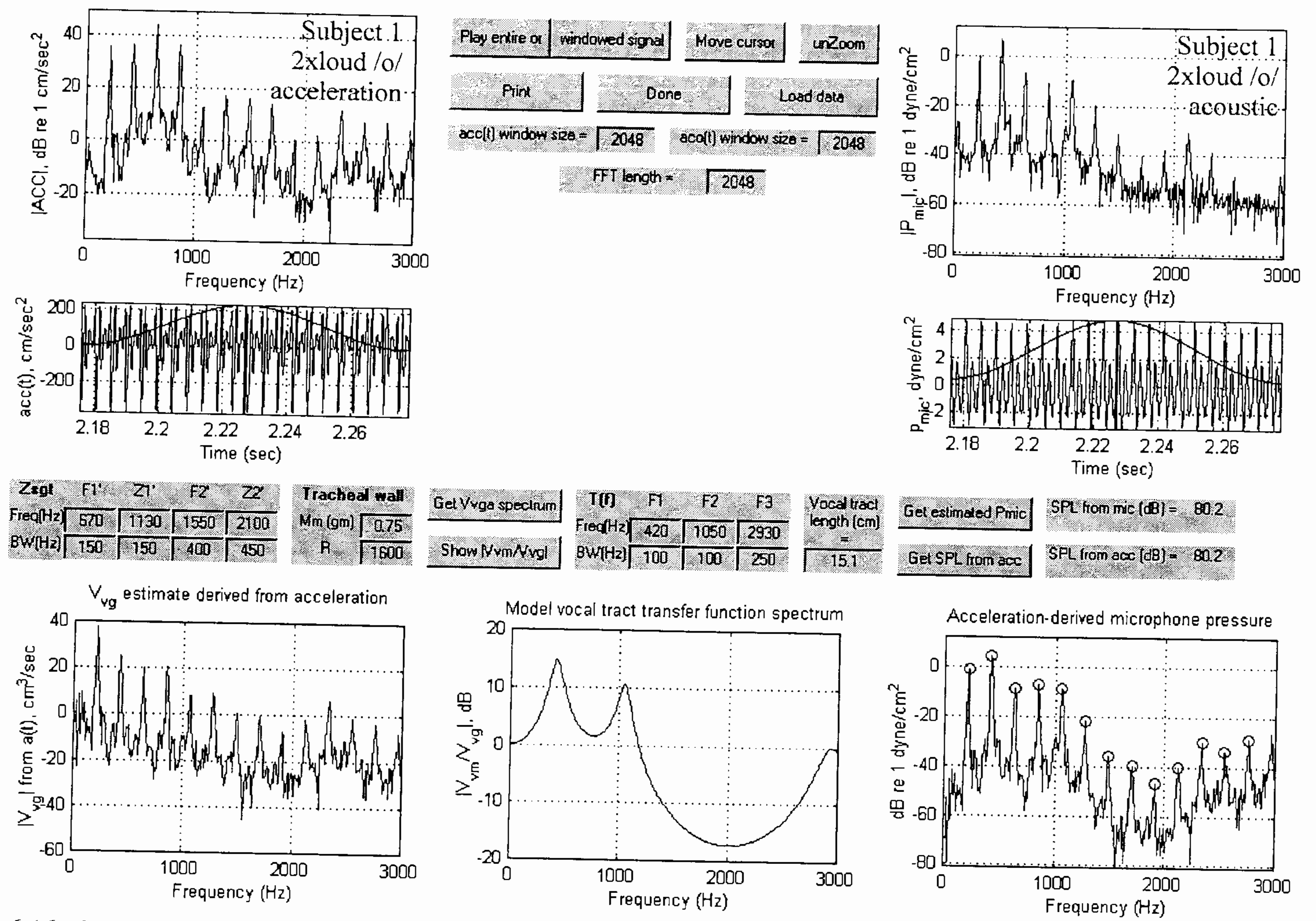


Figure 6.16. SPL analysis screen example from subject 1 sustaining the vowel /o/ twice as loud as normal. In this analysis, the vocal tract formant frequencies for F1 and F2 are altered to 420 and 1050 Hz respectively, and both of their bandwidths are reduced to 100 Hz, resulting in a 1 dB gain for SPL_{acc}. The tracheal wall resistance is increased to 1600 grams/sec, resulting in H1 increasing and bringing SPL_{acc} and SPL_{mic} into agreement.

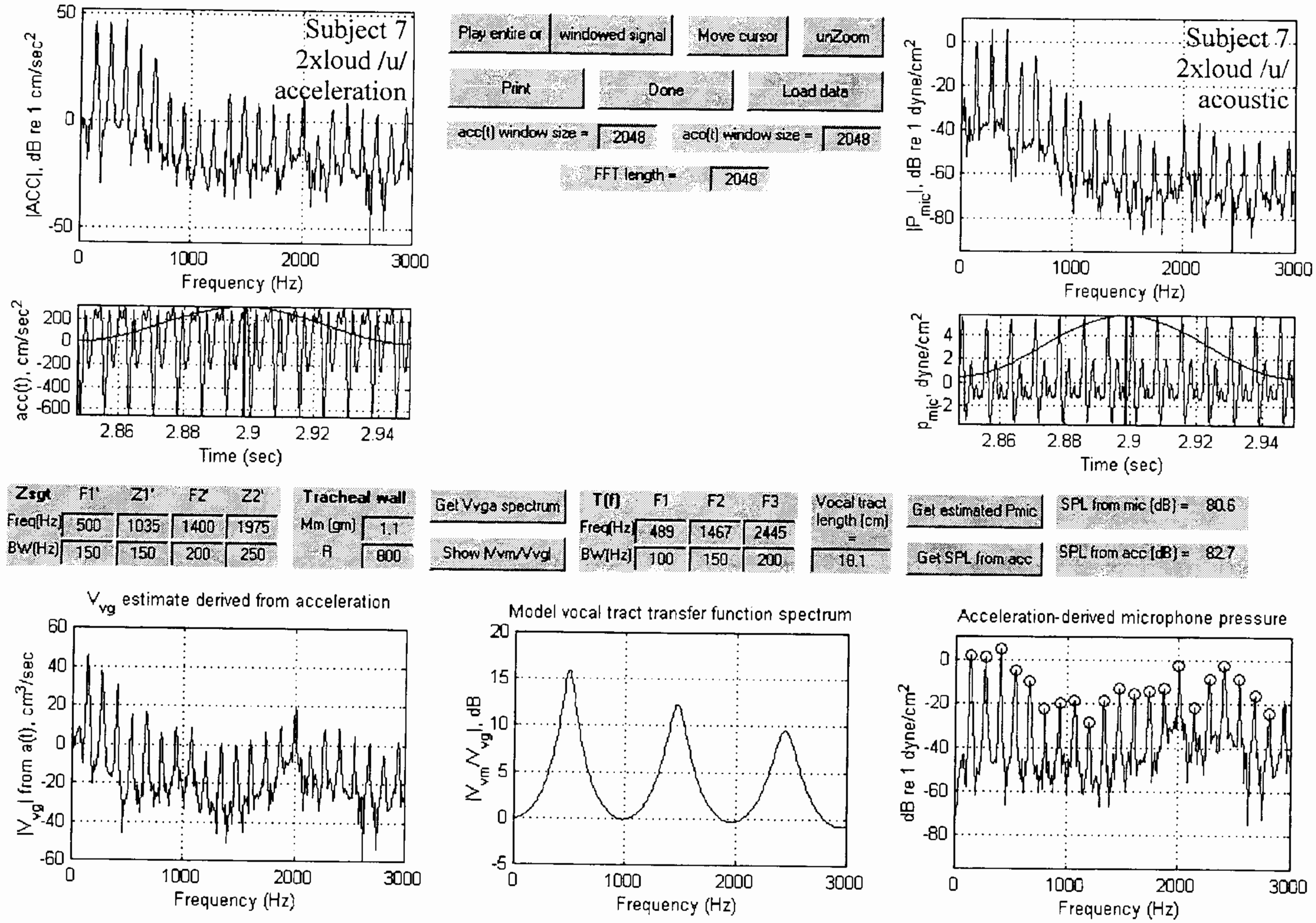


Figure 6.17. SPL analysis screen example from subject 7 sustaining the vowel /u/ twice as loud as normal. Note the peak in the acceleration spectrum (upper left plot) around 2300 Hz, which is likely F3'. This peak is carried through the processing done to calculate the acceleration-derived microphone pressure spectrum (lower right plot), and results in a peak that is a few dB down from the highest spectral peak. This contributes a small error to SPL_{acc}.

The overall results from estimating SPL using the acceleration signal are encouraging, especially considering the current use of a fixed neutral vocal tract in the model. The errors described above suggest ways to improve the vocal system model to more accurately estimate SPL – improvements that are similar to those listed in Section 5.3. First, the values of $\mathbf{M_{TW}}$ and $\mathbf{R_{TW}}$ need to be examined. They were tailored for Experiment II, and may not be correct either in the sense of the processing done in Experiment II or in the physiological sense. Second, the fixed neutral vocal tract model may be contributing small errors to the estimate $\mathbf{SPL_{acc}}$. The decision to use a fixed neutral vocal tract was made with the long-term goal of using the acceleration signal alone to estimate SPL, in which case information about the vocal tract would be unavailable and thus would need to be preset to a fixed value. Although allowing a variable vocal tract model may improve the estimate $\mathbf{SPL_{acc}}$, this change in the model would not be acceptable given this long-term goal. Third, the vocal system model does not account for any glottal coupling between the supra- and subglottal systems, leading to errors when a vocal tract formant appears in the acceleration spectrum. Modifying the model to include the provision for glottal coupling as mentioned in Section 5.3 would likely improve the estimate $\mathbf{SPL_{acc}}$ in these cases. Fourth, the model of the subglottal transfer impedance $\mathbf{Z_T(f)}$ should be expanded to include at least the third subglottal formant $\mathbf{F3'}$, and perhaps also the third subglottal zero $\mathbf{Z3'}$, to account for these peaks and valleys in the acceleration signal that now can be carried through the processing into the estimate $\mathbf{SPL_{acc}}$.

7 Supplemental Experiments: Estimating SPL and the degree of glottal closure without using the vocal system model

Results presented in Chapters 5 and 6 for the vocal system model estimations of MFDR and SPL from the acceleration are encouraging, yet they may be improved further with the model modifications suggested in the discussions of those chapters, and their computation is quite calculation-intensive. In this chapter, one measure of SPL and one measure of glottal closure, which do not use the vocal system model and thus are much less calculation-intensive, are explored for the potential of obtaining this information from the acceleration signal with significantly less signal processing.

7.1 Relationship between SPL and the acceleration spectral slope

7.1.1 Hypothesis and the method of measuring acceleration spectral slope

A second method of estimating SPL from the acceleration signal involves correlating the SPL with a measure of the acceleration's spectral slope. The working hypothesis for this analysis depends on the results of two previous investigations and an assumption. Fant & Lin (1988) related several time-domain features of their LF-model of the glottal source to corresponding frequency-domain features. In particular, their Figure 3 in that publication shows that with all other model parameters remaining fixed, an increase in the magnitude of model parameter E_e, which is equivalent to MFDR, produces an increase in the amount of high-frequency energy present in the spectrum of the flow derivative. Likewise, this would produce an increase in the high-frequency energy of the flow spectrum, although the increase would be diminished by $\frac{1}{j\omega}$, the factor between the flow and flow derivative spectrum. Thus the slope of the flow spectrum, often defined as the magnitude difference between the first harmonic and the harmonic nearest the third formant (Hanson & Chuang, 1999; Hanson, 1997; Holmberg et al., 1995), decreases with increasing MFDR. Furthermore, the results of Holmberg et al. (1994) show a correlation between MFDR and SPL across three loudness conditions for female and male speakers with normal voices. With the results from Fant & Lin (1988) and Holmberg et al. (1994), and the assumption that the subjects recorded achieve complete glottal closure when phonating such that there is no acoustic coupling between the supraglottal and subglottal systems, an increase in SPL should produce an decrease in the spectral slope of the acceleration signal.

Measuring the acoustic spectral slope requires knowing the frequencies of **H1** and **F3**, both of which vary in time during speech. Measuring the acceleration spectral slope has the advantage of a fixed $\mathbf{f_{F3'}}$ across vowels, so that only the first harmonic must be identified. It also introduces a difficulty not present in the acoustic signal: for normal loudness and quieter than normal phonations, **F3′** is often lost in the noise of the acceleration signal, making spectral slope estimates error-prone. For this reason, the acceleration spectral slope is estimated here by the difference between the first harmonic and the harmonic nearest the second subglottal formant, **H1′-A2′**. The signal analysis procedure for this measurement is simple compared to the model-based SPL estimation. First, the program accspl3 (see Appendix B) plots the acceleration and acoustic time signals, and the acceleration spectrum using a 2048-point (102.4 msec) Hanning window. The SPL for the 2048-point window is calculated from the acoustic signal and displayed. Then the subroutine geth1a2 (see Appendix B) has the user select the location of **H1′** and **A2′** with the cursor. Two 7-point (68 Hz) windows, centered on the cursor-selected points, are searched for the actual spectral peaks of **H1′** and **A2'**, then the difference **H1′-A2′** is displayed. An example of this analysis is shown in Figure 7.1. After collecting the **SPL** and **H1′-A2′** data for one subject, a least-squares linear fit is applied to describe the relationship between these two measures.

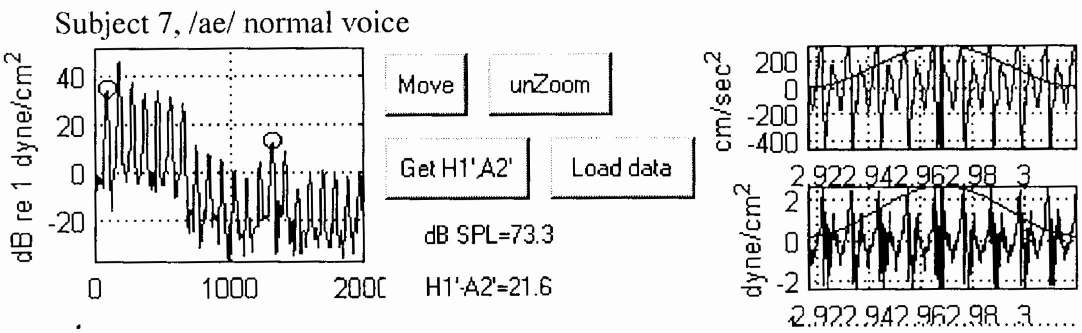


Figure 7.1. Example of the acceleration spectral slope comparison to the SPL. The left side plot shows the acceleration spectrum with H1′ and A2′ circled. The right side upper plot shows the acoustic time signal and the right side lower plot shows the acceleration time signal. For this example, 73.3 dB SPL corresponds to H1′-A2′=21.6 dB.

7.1.2 Results of comparing SPL to H1′-A2′

The technique for correlating the SPL with the acceleration spectral slope aims to do so with as little processing of the acceleration signal as possible. The spectral magnitude of the acceleration signal's first harmonic **H1′** relative to the harmonic closest to the second subglottal formant **A2′** is compared to the SPL. For each subject, a line of best fit in the least-squares sense is calculated for **H1′-A2′** (in dB) versus SPL. The correlation coefficient $\mathbf{r}^2$ and the standard deviation of the error between the true SPL ($\mathbf{SPL_{MIC}}$) and the linear fit's predicted SPL ($\hat{Y}$) are also found. Table 7.1 lists the linear best-fit equations, correlation coefficients, and error standard deviations for each subject.

For every subject except subject 10, the correlation between $\mathbf{SPL_{mic}}$ and **H1′-A2′** is weaker, and in some cases much weaker, than the correlation between $\mathbf{SPL_{acc}}$ and $\mathbf{SPL_{mic}}$ given in Table 6.2. For subject 10 the correlation values are similar at 0.73 for this technique and 0.71 for the model-based technique. Although the linear fit ensures a zero mean error for the prediction of $\mathbf{SPL_{mic}}$ from **H1′-A2′**, the standard deviation of the error is greater than that of the model-based technique for subjects 1, 2, 3, 5, 7, 8, and 9.

Subject	Line of best fit, $\hat{Y}$ =	r^2	$\sigma_{SPL_{MIC}-\hat{Y}}$, dB
1	$-0.39\times(H1'-A2') + 83.8$	0.43	3.8
2	$-0.70\times(H1'-A2') + 97.1$	0.83	2.7
3	$-0.56\times(H1'-A2') + 100.2$	0.72	3.6
4	$-0.48\times(H1'-A2') + 83.7$	0.47	3.2
5	$-0.85\times(H1'-A2') + 102.8$	0.88	3.1
6	$-0.45\times(H1'-A2') + 90.8$	0.75	2.7
7	$-0.47\times(H1'-A2') + 87.8$	0.54	4.0
8	$-0.61\times(H1'-A2') + 93.4$	0.75	4.3
9	$-0.58\times(H1'-A2') + 87.8$	0.21	6.0
10	$-0.66\times(H1'-A2') + 104.1$	0.73	3.4

Table 7.1. Line-of-best-fit equations, correlation coefficients, and the standard deviation of the error for actual SPL (SPL_{MIC}) versus H1′-A2′ from the acceleration. The error equals the true SPL from the microphone minus the predicted SPL from the line of best fit.

7.1.3 Discussion of comparing SPL to H1′-A2′

This technique abandoned the vocal system model in an attempt to correlate the SPL directly with the spectral slope of the acceleration. Table 7.1 summarizes the results of comparing SPL to **H1′-A2′**, and in general the deviation from a linear relationship between SPL and **H1′-A2′** is greater here than the relationship between $\mathbf{SPL_{acc}}$ and $\mathbf{SPL_{mic}}$ described in Section 6.2.

Table 7.2 below compares the mean and standard deviations of the **H1′-A2′** values found here to spectral slope measures of the acoustic signal reported in the literature for the vowel /ae/. Note that these values reinforce the observation of a greater spectral slope for the acceleration than the acoustic signals, as **H1′-A2′** is consistently larger than H1*-A3* or H1-F3, and A3* and F3 are always higher in frequency than **A2′**. Like the Holmberg et al. (1995) data, **H1'-A2′** on average decreases for female voice between the normal and loud conditions, although the mean decrease for **H1′-A2′** was almost 5 dB less than that of the H1-F3 decrease. The standard deviations of **H1′-A2′** and the reported data are within a factor of 2, indicating that the variation across subjects in the spectral slope of the acceleration is similar to that of the acoustic spectral slope.

The result that **H1′-A2′** does not correlate with SPL as well as does $\mathbf{SPL_{acc}}$ from the model-based technique of Section 6 may stem from errors in measuring **H1′-A2′**, or it may indicate that **H1′-A2′** reflects some other aspect of voicing that is related to SPL. Similarly to the model-based technique, some errors arise from coupling between the supra- and subglottal systems, which allows vocal tract formants to create additional peaks in the acceleration

Sex, Loudness	$\overline{H1' - A2'} \pm \sigma_{H1'-A2'}$ (dB)	Hanson (1997) H1*-A3* (dB)	Hanson & Chuang (1999) H1*-A3* (dB)	Holmberg et al. (1995) H1-F3 (dB)
female, normal	30.9 ± 6.6	24.1 ± 3.4	N.A.	22 ± 8.0
female, loud	26.2 ± 5.2	N.A.	N.A.	12.6 ± 8.9
male, normal	28.3 ± 6.1	N.A.	15.5 ± 4.7	N.A.

Table 7.2. Comparison of this study's H1′-A2′ mean and standard deviation values to three previously published reports of acoustic spectral slope measures. All values are for the vowel /ae/ only. Note the similar values of standard deviation, suggesting that variations in the spectral slope of the acceleration and acoustic signals are similar.

spectrum. This is illustrated in Figure 7.2, which compares the acceleration spectrum of subject 9 producing the vowel /a/ in softer-than-normal and softest conditions. The expected increase in **H1′-A2′** from softer-than-normal to softest does not occur most likely because the amount of glottal coupling in the softest condition is greater than that in the softer-than-normal condition (Holmberg et al., 1988), which allows her **F2** at 1500 Hz to boost the acceleration spectrum around **F2′**. The other possibility suggested by these results, that **H1′-A2′** reflects an aspect of voicing other than SPL, is consistent with the hypotheses put forth about H1-F3 by Holmberg et al. (1995) and about H1*-A3* by Hanson (1997) that these spectral tilt measures are related to how abruptly the glottal airflow decreases and the speed quotient, respectively.

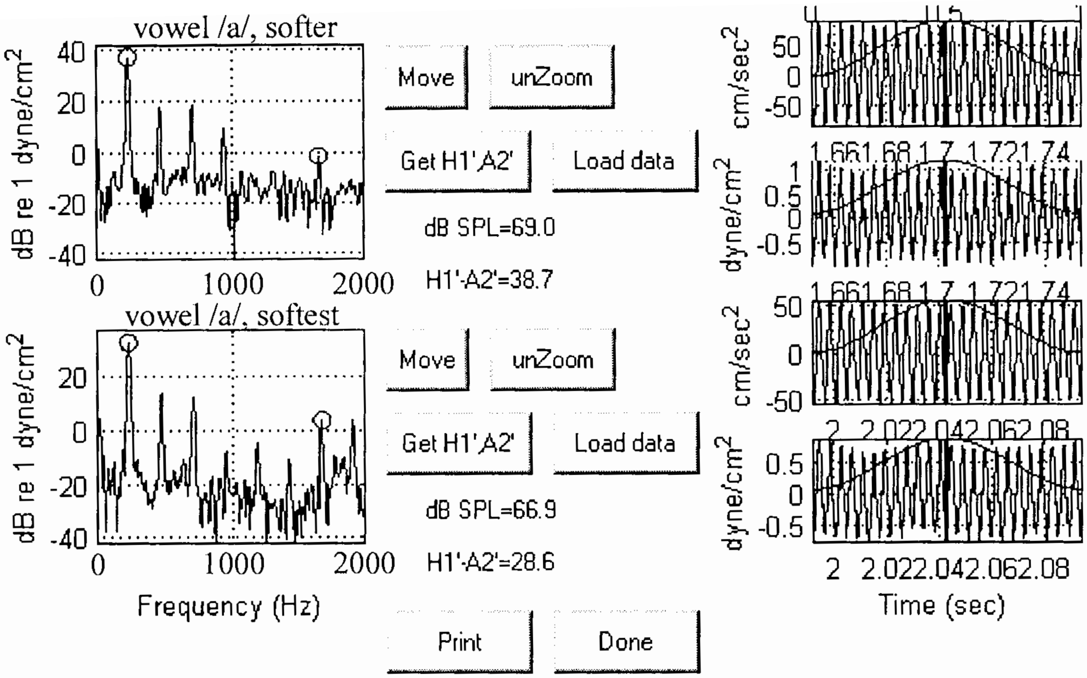


Figure 7.2. Analysis of H1′-A2′ and SPL from subject 9, vowel /a/ in the softer-than-normal (upper spectrum and upper two time plots) and softest (lower spectrum and lower two time plots) conditions. The unexpected decrease in H1′-A2′ from the softer-than-normal to softest conditions may be due to an increase in glottal coupling, which allows F2 to influence A2′.

7.2 Relationship between the degree of glottal closure and the bandwidth of F1′

7.2.1 Hypothesis and the method of measuring the degree of glottal closure and the bandwidth of F1′

Hanson (1999, 1997) proposed a model relating the glottal chink area during closure to the bandwidth of the first vocal tract formant. The goal of this investigation is to extend these findings to the acceleration signal by analyzing the data from vocal task 1. Stevens (1998) and Klatt & Klatt (1990) point out that from normal (modal) to breathy phonation the DC component of the airflow increases significantly. Thus, estimates of the first subglottal formant bandwidth from the acceleration signal are compared to the minimum flow from the Rothenberg mask to assess whether a correlate of the degree of glottal closure exists in the acceleration signal. Figures 7.3a through 7.3c are intended to show the sequence of the processing done on the acceleration and airflow signals in this experiment. As in Experiments II and III, the first step is to select stable segments from both signals as shown in Figure 7.3a.

The minimum flow was determined using the MATLAB® program accflo4 (see Appendix B), which was adapted from the program hiband by Qi (1996b) and based on the technique of Perkell, Holmberg & Hillman (1991). To obtain the minimum flow estimate from the raw airflow signal, first the vocal tract transfer function was estimated through Linear Prediction, as done in Experiment II for MFDR. The bandwidths of the formants above **F3** were fixed at 250 Hz. The higher formants were included up to half the sampling rate, or 10kHz. The aliased version of the digital filter produced the required higher-pole correction, as described above in the analysis methods for Experiment II (Gold & Rabiner, 1968).

The airflow signal is inverse filtered to give $\mathbf{v_{VGF}(t)}$ with the goal of minimizing the ripple during the closed phase, which is achieved by manipulating the frequencies and bandwidths of **F1**, **F2**, and **F3** as described in Section 5.1. Then nine to ten consecutive periods of $\mathbf{v_{VGF}(t)}$ are displayed and the experimenter selects the first two peaks of $\mathbf{v_{VGF}(t)}$ to give accflo4 an estimate of the fundamental period **T0**. The program then searches for the actual first two peaks within two 11-point (0.55 msec) windows centered on each cursor-selected point. The time difference between the two actual peaks gives **T0**. Then the third peak is searched for in an 11-point window centered at 3×**T0**, and once found the value of **T0** is modified to one-half the difference in time between the first and third peaks. This iterative procedure of finding the next airflow peak then modifying **T0** was fashioned after the one described in Section 6.1. In this way, the first nine consecutive peaks of $\mathbf{v_{VGF}(t)}$ are identified, and circled on the analysis display. Between each two adjacent peaks, accflo4 circles the locations where $\mathbf{v_{VGF}(t)}$ drops below a threshold of 30% of its adjacent peak values, and then searches for the places below that 30% threshold where the signal first changes slope. For example, when searching to the left of a peak, where the slope first changes from positive to negative below the 30% threshold would be marked. Between the marks of slope change, accflo4 finds the median minimum value of $\mathbf{v_{VGF}(t)}$. The mean of eight consecutive median values is taken as the minimum flow. This minimum flow estimation process is summarized in Figure 7.3b.

The final step in the glottal closure analysis is determining the bandwidth of the first subglottal formant, fashioned after the method of Hanson (1997) and shown in Figure 7.3c. The program accflo4 applies a 512-point FIR digital band-pass filter to the raw acceleration time signal **a(t)**. The center frequency of the filter was chosen as the frequency used for each individual's **F1′** in Experiment II (see Table 5.2). The filter bandwidth was fixed at 400 Hz, less than Hanson's bandwidth of 600 Hz, because often the first harmonic of the acceleration spectrum is greater in amplitude than the harmonic nearest **F1′** and a filter bandwidth of 600 Hz would pass the first harmonic of some females whose **F0** is 200 Hz or higher. Nine to ten consecutive periods of the filtered acceleration, centered in time on the periods of $\mathbf{v_{VGF}(t)}$ analyzed, are displayed in the lower left plot of the analysis screen. The experimenter then selects the first two peaks of the filtered acceleration signal, and their spacing is used to find the mean value of the first eight consecutive peaks, $\mathbf{A_1}$, with an iterative process similar to the one described above. Then the experimenter selects the next-highest peaks in the **F1′** oscillation that follow the two peaks selected in the last step, and the program uses the same

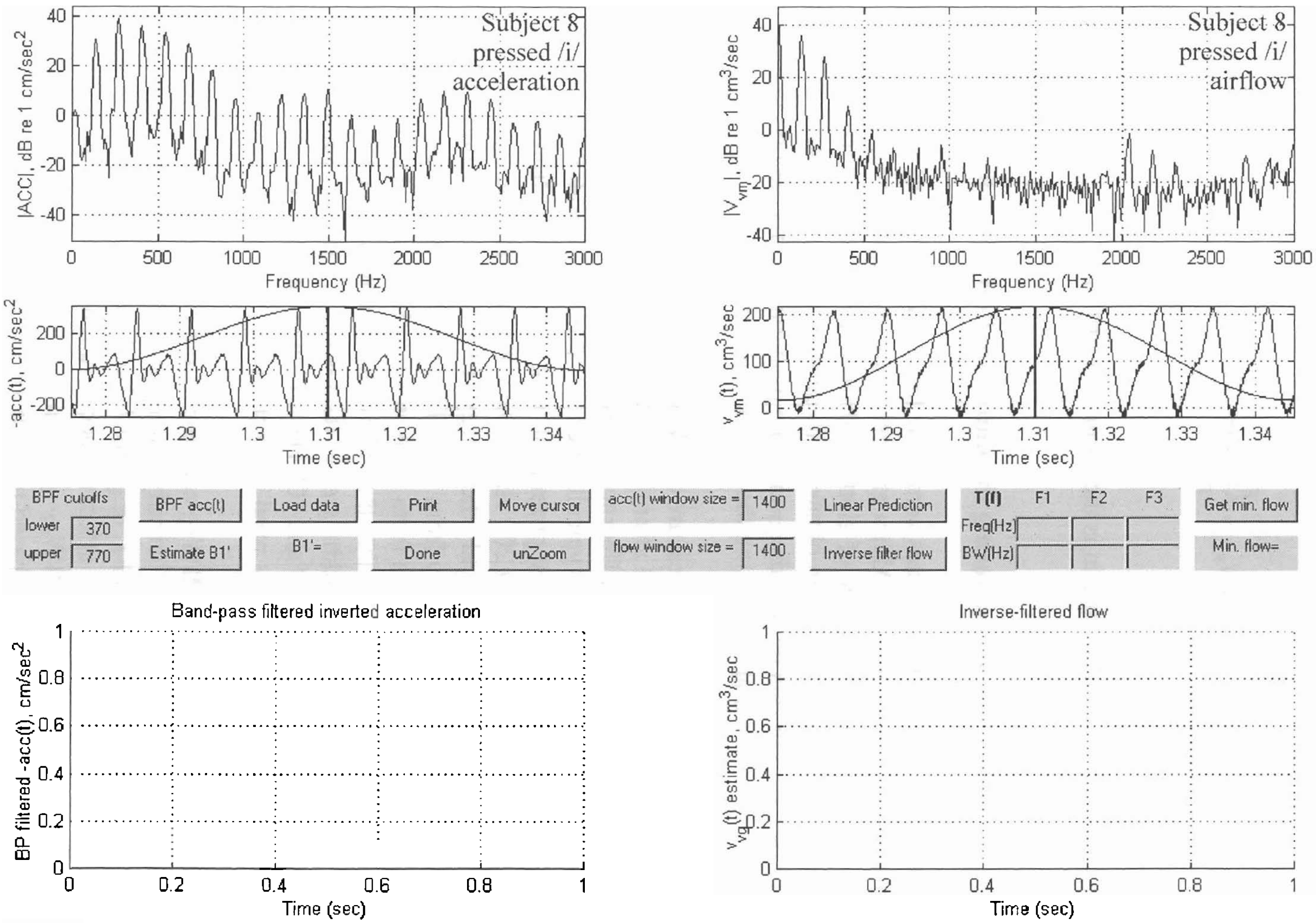


Figure 7.3a. Glottal closure analysis screen example, step 1: elect stable signal segments from both the acceleration (left) and airflow (right). Adjust the window size on the sig ıals (center text boxes) such that 9-10 periods of each signal are s' own.

iterative process with their spacing to find the mean value of the first eight next-highest peaks, $\mathbf{A_2}$. The peaks used to calculate $\mathbf{A_1}$ and $\mathbf{A_2}$ are circled on the filtered acceleration signal in the analysis screen. The mean amplitudes $\mathbf{A_1}$ and $\mathbf{A_2}$ are used to estimate the **F1′** bandwidth according to the equation:

$$BW_{F1'} = \frac{f_{F1'}}{\pi} \times \ln\left(\frac{A_1}{A_2}\right)$$

Equation 3.6

where $f_{F1'}$ is the frequency of *F1′*, and ln is the natural logarithm.

7.2.2 Results and discussion of comparing the minimum flow to the bandwidth of F1′

The raw data of the minimum flow values versus the bandwidth of F1′ estimates are not presented here but will be discussed. Although these results suggest some potential in obtaining information about the degree of glottal adduction from the acceleration, they predominantly expose the difficulties in obtaining these measures and basing the measure on the bandwidth of **F1′**. For this reason, discussing the problems of this method in light of the results is more appropriate than presenting the results themselves.

The difficulty in obtaining a reliable measure of $\mathbf{BW_{F1'}}$ is the main source of error in this technique. This analysis was fashioned after the estimates of vocal tract first formant bandwidths by Hanson (1997). Two differences between Hanson's analysis and this work that contribute to the error are 1) **F1** is typically the most prominent peak in the acoustic spectrum while **H1′** is typically the most prominent peak in the acceleration spectrum, and 2) $\mathbf{f_{F1}}$ is typically higher than $\mathbf{f_{F1'}}$ by about 200 Hz for the vowels /a/ and /ae/, so band-pass filtering around **F1** will likely capture only the energy around **F1** while band-pass filtering around **F1′** will likely pass some energy from **H1′**. To avoid this, the band-pass filter used in this experiment is reduced in bandwidth from Hanson's 600 Hz to 400 Hz. This filter bandwidth reduction may compound rather than alleviate the errors, because the past reports of $\mathbf{BW_{F1'}}$ (see Table 5.2) show bandwidths greater than 400 Hz. As a consequence, most of the estimates for $\mathbf{BW_{F1'}}$ are between 50 and 100 Hz, which are clearly too small given both the past reports and the values of $\mathbf{BW_{F1'}}$ used in Experiment II.

Another source of error in measuring $\mathbf{BW_{F1'}}$ can come from acoustic coupling between the supra- and subglottal systems, which ironically is assumed to be a correlate of the attempted measure. If **F1** is near in frequency to **F1′** (e.g., for /a/ and /ae/ in males, and /o/ in females), then glottal coupling could allow **F1** to affect the acceleration spectrum, which in turn would make the time signal measurement of $\mathbf{BW_{F1'}}$ problematic. However, this observation suggests the possibility of a spectrally-based measure on the acceleration of the amount of vocal tract influence in the **F1** region. For example, if the magnitude of **F1′** could be

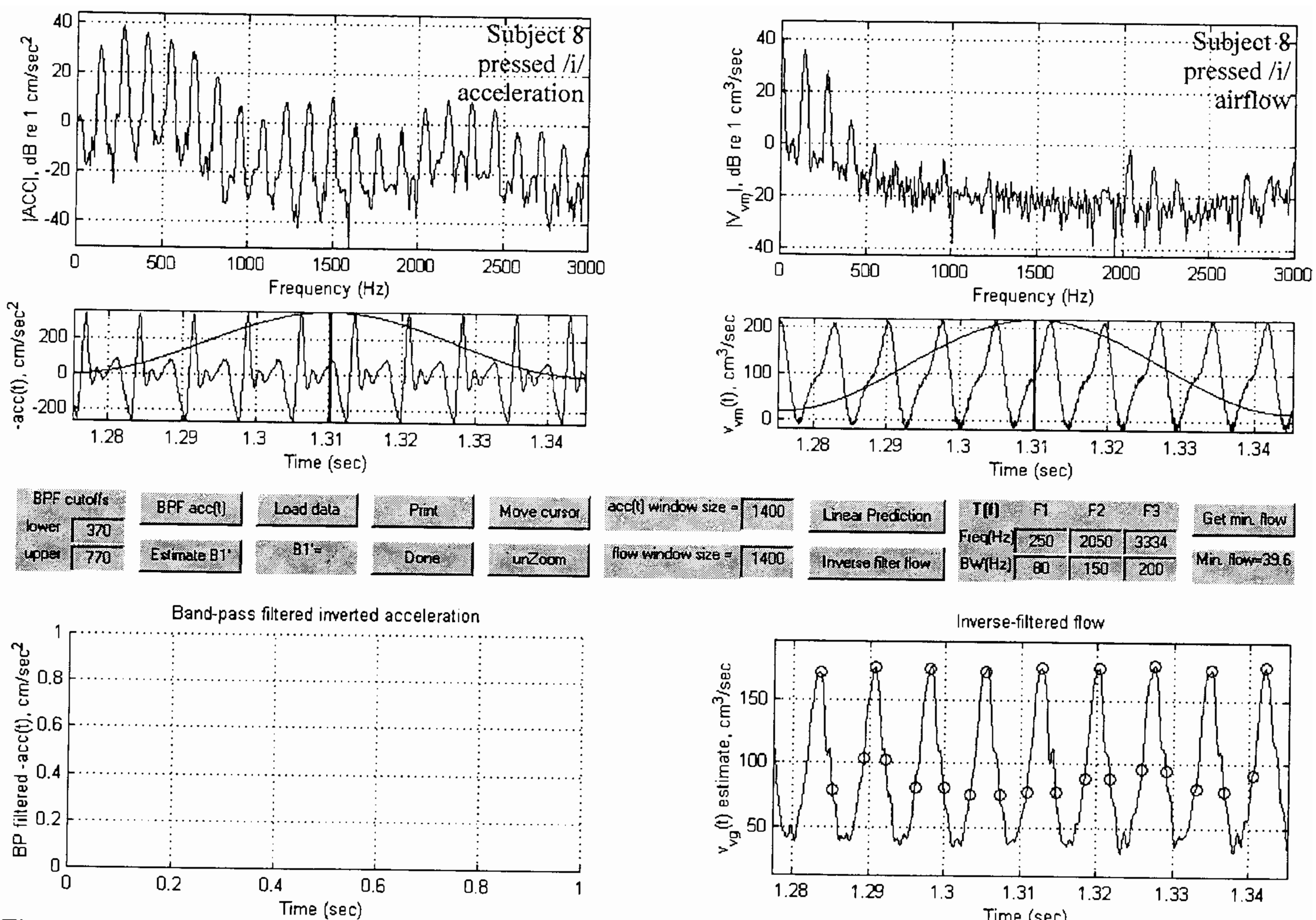


Figure 7.3b. Glottal closure analysis screen example, step 2: inverse filter the airflow using the method described in Section 5.1 for Experiment II. After minimizing the ripple in the inverse-filtered flow, calculate the minimum flow using the algorithm outlined in Section 7.2.1, and list that minimum in the rightmost lower text box.

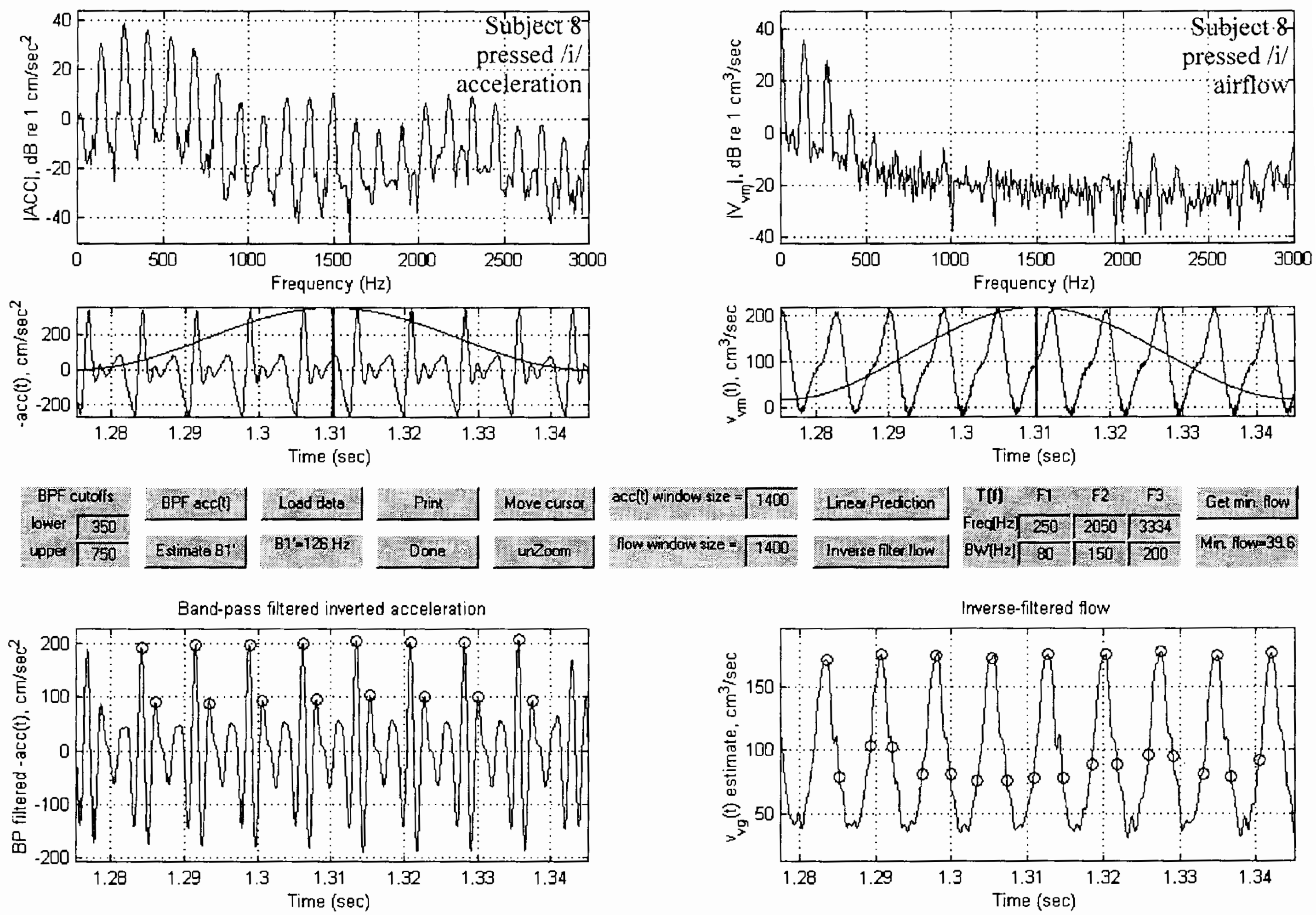


Figure 7.3c. Glottal analysis example screen, step 3: band-pass filter the acceleration signal with a 400 Hz wide filter centered on the frequency of F1' listed in Table 5.2. Calculate the mean difference between the primary and secondary peaks of the filtered acceleration time signal (circles in lower left plot) and find the bandwidth of F1' using Eqn. 7.1.

compared to **F1** in the acceleration spectrum, then a measure such as **F1′-F1** may correlate with the minimum airflow.

Other investigators have noted problems with using the Rothenberg mask to measure estimate minimum flow (Holmberg et al., 1995; Rothenberg, 1973). The minimum flow errors are obvious only for the vowels /i/ and /a/ of subject 2 because her inverse-filtered flow signal gave negative minimum flow. However, that does not preclude errors elsewhere in the data. To minimize the likelihood of these errors, the experimenter checked for obvious mask leaks at the time that the mask was fitted for the recordings, and continually monitored the signal from the mask for abrupt changes in the DC component. A confounding factor in this is the Rothenberg mask signal's tendency to drift, such that the DC component is slowly changing over time. To counter these problems, a more direct measure of glottal closure such as endoscopy may be warranted.

Also, four of the nine subjects exhibited difficulty performing the pressed voice condition correctly, which may be due to subject naivete with regards to increasing glottal adduction. Subject 1 produced almost identical minimum airflows for the pressed /ae/ and the normal /ae/. The minimum airflows for the pressed vowels /i/ and /a/ by subject 6 are about three times those of his normal vowels. Subject 8 produced a pressed /ae/ and /a/ with twice the minimum airflow of those normal vowels. Lastly, subject 9 produced the pressed vowels /ae/ and /i/ with slightly more minimum airflow than her normal versions. In contrast, all 9 subjects increased their minimum airflow when asked to increase breathiness as compared to normal voice.

In spite of the many errors in this experiment, seven of the nine subjects displayed increases in both the minimum airflow and the $\mathbf{BW_{F1'}}$ when going from a normal /i/ to a breathy /i/. This suggests that the acceleration has some potential for providing a measure of glottal adduction, although modifying a method suited for speech signal processing is clearly inadequate for this task.

8 Conclusions

The results of this work have further developed the potential of the acceleration as measured from the skin on the neck as a useful tool in quantifying vocal function. A vocal system model is proposed and assumptions about the model are tested in Experiment I. Modeling exercises and data from the ten subjects recorded for this thesis work show that vertical laryngeal position may alter the first subglottal transfer impedance zero by about 100 Hz and 5 dB. Changes in the subglottal transfer impedance of this order will not significantly affect the estimates of MFDR and SPL made in Experiments II and III. Similarly, lung volume does not show a consistent effect on the frequency of the first subglottal zero, so it is unlikely to have an appreciable effect on the measured acceleration signal.

The model is first used to estimate the Maximum Flow Declination Rate or MFDR in Experiment II. Results of comparing the airflow-derived MFDR to the acceleration-derived MFDR show a high correlation ($r^2 \geq 0.75$) for 9 out of 10 subjects. The subject whose correlation fell below 0.75 (subject 6) was informally diagnosed with bilateral vocal fold sulci after the data analysis, so the interpretation of his data should be tempered with that knowledge. The error between the estimated and true MFDR is less than 2 dB for all subjects, and has a mean of –0.5 dB across all subjects. The error standard deviation across all subjects is 2.9 dB, suggesting that if the error is assumed to be Gaussian with zero mean, ninety-five percent of the acceleration-derived MFDR estimates will be within ±5.8dB of the true MFDR. Given that the range of normal MFDR is about 60 dB, the acceleration-derived MFDR may be clinically useful with this error standard deviation.

Sound Pressure Level (SPL) is estimated from the acceleration using the model in Experiment III. Correlation coefficients between the true SPL and the acceleration-derived SPL are greater than 0.75 for 8 out of 10 subjects. Again, subject 6 shows the lowest r^2 value. The mean error between the true and estimated SPL for each subject is typically larger than the mean error from Experiment II. Likewise, the mean error across subjects is –1.9 dB, almost four times that of Experiment II. The error standard deviation across all subjects is similar to that of Experiment II at 3.5 dB.

The measures based solely on the acceleration signal, unlike the model-based measures above, produce less favorable results than Experiments II or III. Correlating SPL with **H1′-A2′** from the acceleration signal gives both lower correlation coefficients and greater error standard deviations than the model-based SPL estimates of Experiment III. Adopting an acoustic measure of glottal closure for the acceleration signal shows several limitations of the technique and gives poor results due to underestimated bandwidths of the first subglottal formant. However, the increase in minimum flow and the bandwidth of the first subglottal formant from a normal /i/ to a breathy /i/ for 7 out of 9 subjects suggests some potential of the acceleration to measure glottal closure.

Systematic errors from these experiments suggest several ways to modify the vocal system model to improve the results. Incorporating glottal coupling, making direct measures of the tracheal wall parameters, averaging the subglottal transfer impedance parameters over vowels and loudness conditions, and including higher formants or zeros in the subglottal transfer impedance are all modifications to the model that will likely improve MFDR and SPL estimates.

9 Future work

This work is guided by the long-term goal of a device capable of monitoring a patient's voice use over an entire day - a "Portable Vocal Accumulator" as mentioned in Section 1.1. Applying the vocal system model developed here to the real-time acquisition of MFDR and SPL estimates using an accelerometer-based Portable Vocal Accumulator will be the next step in this work. Having MFDR and SPL data for a patient's typical work day will greatly improve a clinician's ability to assess a patient's voice use demands, and thus tailor that patient's vocal therapy appropriately.

The model modifications described in the conclusions will be implemented one at a time, to test the improvement gained by each. Relatively simple improvements, such as averaging the subglottal transfer impedance parameters over vowels and loudness conditions, will be incorporated first. Making direct measurements of the tracheal wall parameters is also an important next step, as it may show that variation across individuals is not as great as this work suggests, which could allow these parameters to remain fixed across individuals.

Lastly, measuring the degree of glottal closure with the acceleration remains an important goal. By incorporating glottal closure into the vocal system model, a model-based measure of glottal closure using the acceleration may be possible.

10 Acknowledgements

Thanks go to all of these people who helped me achieve this work:

My wife, Deborah, who loves and supports me through both my celebration and frustration;

My thesis committee – Bob Hillman, Ken Stevens, and Jeff Fredberg – for being persistently helpful, regularly questioning, and for relentlessly building my confidence;

Jim Kobler, James Heaton, Geoff Meltzner and Ehab Goldstein for much figure editing and writing advice; and

John Rosowski, for much help with calibrating the accelerometers used in the recordings.

Appendix A

On the following four pages appears the consent form used for the subjects to provide informed consent before being recorded.

Note that this form covers many different experiments that take place in the Voice and Speech Laboratory, and that not all of the procedures listed in the form (e.g., endoscopy) were performed for the experiments described in Section 3.1.

MASSACHUSETTS EYE & EAR INFIRMARY

INFORMED CONSENT

TITLE: Objective Assessment of Vocal Hyperfunction

INVESTIGATOR: DATE:

DESCRIPTION AND EXPLANATION OF PROCEDURES:

You, ________________________________, are being asked to participate in a research study which uses a variety of techniques to study various aspects of the human voice. For the purposes of this study, all of these techniques are considered to be experimental. The overall goal of this project is to provide new information concerning the etiology (causation) and treatment of disorders which affect the voice.

This project is being conducted in the Voice and Speech Laboratory at the Massachusetts Eye and Ear Infirmary. In the lab, you will have measuring devices attached to the area around your mouth, nose, neck and chest. With these devices in place, you will be asked to pronounce a number of utterances in a variety of ways (e.g., loud voice, soft voice, high pitch voice, low pitch voice) and read from some prepared text while recordings are made of the outputs from the measuring devices and of your voice.

The following measurements will be made. Air pressure inside your mouth will be measured with a tube that will be placed between your lips and held in position in the middle of your mouth. The outside end of the tube will be attached to a mask that fits over your mouth and nose. The mask is made of rubber, and it has a number of holes drilled in it which are covered by a fine wire mesh. It contains a flow-measuring device. Movements of your vocal cords will be measured using electrodes that are held against your neck by an adjustable band. Other electrodes will be taped to the surface of your neck to monitor the activity of your throat muscles. Measurement of your breathing patterns will be made using two elastic bands that strap around your rib cage and abdomen. In addition, a small metal disk (5.6 mm by 8 mm) may be attached to the front of your neck just above your collar bone using an adhesive that is specially designed for attaching devices to the skin (Skin Bond). This device will monitor the vibrations of your neck as you phonate. All of the above measuring devices are connected to recording equipment. None of these devices interfere with normal function or cause any discomfort.

You will also be given a brief hearing test (screening) through headphones and asked to fill out several forms which gather information about your health, voice use and level of emotional stress.

In addition, your larynx will be examined and photographed. This will involve the placement of a rigid laryngoscope (a tube approximately 5 mm. in diameter) through your mouth and into the back of your throat and/or the passage of a flexible laryngoscope (a tube approximately 3.4 mm. in diameter) up through your nose and down the back of your throat. A camera will be attached to the end of these scopes outside of your body to take pictures of your larynx.

You will receive $35 each time you participate in an assessment session involving the measurements described above. You will probably be asked to participate in only one assessment session, but you could be asked if you can return for additional assessment sessions if it is deemed necessary. Each assessment session will involve from one to two hours of your time, or less.

You may also be asked to undergo the placement of additional devices between your vocal cords and/or into your esophagus (food pipe). This will entail the application of additional topical anesthetic via spraying and/or swabbing of your mouth, throat and nose. A small pressure sensing device on a curved rod will then be passed through your mouth and throat, and positioned between your vocal cords in your larynx. This procedure will be completed in 15 minutes or less. In addition, a small, collapsed balloon will be passed through your nose and into your throat, at which point you will swallow it into your food tube by drinking some water. This additional procedure will be completed in 45 minutes or less. You will be paid an additional $20 to $40 for participating in one or both of these this extra testing procedures.

<u>RISKS AND DISCOMFORTS:</u>

You may experience slight discomfort (irritation) in your nasal passages associated with the passage of the flexible scope. To reduce this discomfort, a light topical anesthetic (solution of 3% lidocaine and 0.25% phenylephrine) will be sprayed into your nose prior to examination with the flexible scope. The anesthetic will normally wear off in approximately 15 to 20 minutes. There is also the possibility that the flexible scope could cause a nosebleed. If this occurs, testing will be suspended and appropriate medical attention will be provided. Placement in your throat of either scope may cause gagging. If this becomes a problem, your throat will be sprayed with the topical anesthetic (10% lidocaine). There is the possibility that you could experience an allergic reaction to the topical anesthetic. If this occurs, testing will be suspended and appropriate medical attention will be provided.

Participation in the additional testing that entails placement of a pressure sensing device between your vocal cords may increase the likelihood of gagging, coughing and/or choking. Also, the introduction of the balloon through your nose could cause increased coughing of choking, or a nosebleed. If any of this occurs, testing will be suspended and appropriate medical attention will be provided as needed.

POTENTIAL BENEFITS:

You will receive no health or other benefits from your participation in these assessments.

However, it is expected that the results from this project will assist in better diagnosis and treatment of voice disorders.

There will be no costs to you for any of the services that you receive as part of participating in this study.

CONFIDENTIALITY:

All of the data gathered in this investigation will be held in strict confidence. Any reports or publications will not identify individual participants by name or initials.

RIGHT TO WITHDRAW:

You are not obligated to participate in this study. If you choose not to participate your present or future medical care will not be affected in any way. Also, if you participate, you may withdraw your consent and discontinue participation at any time without affecting your medical care.

COMPENSATION:

In the unlikely event that you should be injured as a direct result of this study, you will be provided with medical treatment. This treatment does not imply any negligence on the part of the Massachusetts Eye and Ear Infirmary or any of the physicians involved. When applicable, the Massachusetts Eye and Ear Infirmary reserves the right to bill third party payers for any emergency services rendered. The Massachusetts Eye and Ear Infirmary does not have any program to provide compensation as a result of any injuries.

RIGHT TO ASK QUESTIONS:

If you have questions regarding this research or your participation in it, either now or at any time in the future, please feel free to ask them. The research team, particularly Dr. Hillman, who can be reached at (617) 573-4050, will be happy to answer any questions you may have. You may obtain further information about your rights as a research subject by calling Carl Finn, Director of Research Administration at the Massachusetts Eye and Ear Infirmary at (617) 573-4080. If any problems arise as a result of your participation in this study, including research-related injuries, please call the principal investigator, Dr. Hillman at (617) 573-4050 immediately.

CONSENT:

I have read the above description of this research study, and I understand it. I have been informed of the risks and benefits involved, and all of my questions have been answered to my satisfaction. Furthermore, I have been assured that any future questions I may have will also be answered to my satisfaction. Furthermore, I have been assured that any future questions I may have will also be answered by a member of the research team. I understand that I will receive a copy of this form.

I understand that I am free to withdraw this consent and discontinue participation in the described research study.

I voluntarily consent to my participation in the described research study.

Date	Name of Subject	Signature of Subject
Date	Name of Witness	Signature of Witness
Date	Name of Investigator	Signature of Investigator

Investigators:

Robert E. Hillman, Ph.D.
Voice and Speech Laboratory
Massachusetts Eye & Ear Infirmary
617-573-4050

James Kobler, Ph.D.
Voice and Speech Laboratory
Massachusetts Eye & Ear Infirmary
617-573-4050

Katherine Verdolini, Ph.D.
Voice and Speech Laboratory
Massachusetts Eye & Ear Infirmary
617-573-4050

Steven Zeitels, M.D.
Voice and Speech Laboratory
Massachusetts Eye & Ear Infirmary
617-573-4050

Harold Cheyne
Voice and Speech Laboratory
Massachusetts Eye & Ear Infirmary
617-573-4050

Ann McLean-Muse, Ph.D.
Voice and Speech Laboratory
Massachusetts Eye & Ear Infirmary
617-573-4050

11.2 Appendix B - MATLAB® programs created for this work

Program name	Page
abf2mat.m	116
acc2vvg.m	119
accflo3.m	121
accflo4.m	125
accspl1.m	127
accspl2.m	129
accspl3.m	132
autolpc.m	134
bpfacc.m	135
derivf.m	136
domain.m	136
dVdtplot3.m	137
dVspec3.m	139
gdp.m	140
gdplong.m	140
getaccspl.m	141
getbf1.m	142
getF0.m	143
getfb2.m	144
getfbl3.m	144
getfz.m	145
geth1a2.m	146
getHvt.m	147
getmfdr3.m	147
getminflo.m	149
invfilt3.m	150
invfilt4.m	152
invfiltat3.m	153
linespec.m	155
LPbp125.m	156
lplv.m	156
lplvlong.m	161
movespec.m	166
newdata.m	167
newdata2.m	169
newdata3.m	171
newdata4.m	173
newtime3.m	175
newtime4.m	177
newtime5.m	179
newtime6.m	181
newwindow.m	182
pkpsd.m	184
rad.m	185
s2z.m	186
setF1F2F3.m	187
setF3orL.m	187
update.m	188
vtractz.m	188
vvg2mic.m	189
zoominout3.m	190
zoominout4.m	191
zoominout5.m	193
zoominout6.m	194

```
% abf2mat.m
%
% Usage:  abf2mat
% Input:  1) the name of an Axoscope .ABF file to process
%       2) the gain factors on the Axoscope acceleration,
%          audio and flow signals respectively
% Output: several MATLAB .MAT files containing data that are
%       converted from amplitude bits to the appropriate units
%       (e.g., cm/s^2 for the acceleration).
%
% This MATLAB m-file allows editing of the data from Harold
% Cheyne's thesis recordings.  Prompts are provided for the user
% to input MATLAB file names following the editing of the data.
%
% portions were adapted from "abfread.m", by Geoffrey Meltzner (8/21/97)
% modified by Harold Cheyne (11/1/2000)

another = 1;
count = 0;
gainquest1 = 'Did any signal gains change during the experiment?';
gainquest2 = 'Were any signal gains changed for this file?';
gainquest3 = 'Enter new acceleration gain factor, where New Gain = Factor*(Old Gain)';
gainquest4 = 'Enter new microphone gain factor, where New Gain = Factor*(Old Gain)';
gainquest5 = 'Enter new airflow gain factor, where New Gain = Factor*(Old Gain)';
gainchange = questdlg(gainquest1,'Signal gains','Yes','No','No');
g1 = inputdlg('Enter the Cyberamp gain for the accleration signal.');
v2a = 25472./str2num(char(g1))./1.152;       % v2a = V->cm/s^2 factor for white acc
[infile,inpath] = uigetfile('*.abf','Enter the microphone calibration filename');
while another                                          % Keep processing until no files are left.
  count = count + 1;                              % count = # of files processed
  % This first section sets up variables for reading the .ABF file.
  m = [8;10;14;40;100;120;122;126;244;252;378;410;442;602;730;1178];
  s = [1;1;1;1;1;1;1;1;1;1;1;15;160;128;1;4];
  t = ['short';'long ';'short';'long ';'short';'short';'float';'float';'float'];
  t(10:16,:) = ['long ';'short';'short';'char ';'char ';'float';'float'];
  v = ['nOperationMode          ';'lActualAcqLength        ';'nNumPointsIgnored       '];
  v(4:5,:) = ['lDataSectionPtr         ';'nDataformat             '];
  v(6:7,:) = ['nADCNumChannels         ';'fADCSampleInterval      '];
  v(8:9,:) = ['fADCSecondSampleInterval';'fADCRange               '];
  v(10:11,:) = ['lADCResolution          ';'nADCPtoLChannelMap      '];
  v(12:13,:) = ['nADCSamplingSeq         ';'sADCChannelName         '];
  v(14:15,:) = ['sADCUnits               ';'fADCProgrammableGain    '];
  v(16:17,:) = ['fSignalLowpassFilter    ';'offset                  '];
  v(18:19,:) = ['format                  ';'fs                      '];
    fid=fopen(infile,'r');                     % Open the .ABF file for reading only
    for n=1:16,
    fseek(fid,m(n),-1);
    eval([deblank(v(n,:)) '=fread(fid,s(n),deblank(t(n,:)));']);
    end
    offset=lDataSectionPtr*512;
    if nDataformat==0
    format='integer*2';
    else
    format='float';
    end
```

```
   fs=1/(nADCNumChannels*fADCSampleInterval*0.000001);
   fseek(fid,offset,-1);
   alldata=fread(fid,[nADCNumChannels,lActualAcqLength/nADCNumChannels],format);
   for n=1:nADCNumChannels,
   varname(n,:) = char(sADCChannelName(1+10*nADCSamplingSeq(n):10+10*nADCSamplingSeq(n)))';
        eval([varname(n,:) '=alldata(n,:).*fADCRange./lADCResolution;']);
end
clear alldata m s t v nOperationMode lActualAcqLength nNumPointsIgnored;
clear lDataSectionPtr nDataformat nADCNumChannels fADCSampleInterval;
clear fADCSecondSampleInterval fADCRange lADCResolution nADCPtoLChannelMap;
clear nADCSamplingSeq sADCChannelName sADCUnits fADCProgrammableGain;
clear fSignalLowpassFilter offset format;
   % End of .ABF file reading section.

% Next, plot the relevant data from the .ABF file for editing.
figure(1)
clf
set(gcf,'Renderer','OpenGL','Position',[1 29 1152 768],'Resize','off')
xmax = length(accel);
if count == 1
  clear EGGtrack flow accel;
  plot(audio(1:10:xmax))
  axis([1 xmax/10 -10 10])
  hold
  for m = 1:2,
    if m == 1
      h = msgbox('Click start & end points of the microphone calibration');
    else h = msgbox('Click start & end points of the SPL reading');
    end
    waitfor(h,'ButtonDownFcn');
    [x,y] = ginput(2);
    x1 = round((x(1)-1).*10+1);
    x2 = round((x(2)-1).*10+1);
    plot([x(1) x(1)],[-10 10],'r',[x(2) x(2)],[-10 10],'r')
    if m == 1
      Vrms = sqrt(mean(audio(x1:x2).^2));
    else
      soundsc(audio(x1:x2),fs);
    end
  end
  spl = inputdlg('What was the sound meter reading in dB SPL?','Measured SPL');
  Prms = 2e-4.*10.^(str2num(char(spl))./20);% rms sound pressure (dyne/cm^2)
  v2p = Prms./Vrms;                                   % v2p = voltage to pressure factor
        [infile,inpath] = uigetfile('*.abf','Enter the flow calibration filename');
elseif count == 2
  clear EGGtrack accel;
  subplot(2,1,1)
  plot(flow(1:10:xmax))
  axis([1 xmax/10 -10 10])
  hold on
  subplot(2,1,2)
  plot(audio(1:10:xmax))
  axis([1 xmax/10 -10 10])
  hold on
  scale = ['0 ';'25';'18';'12';'8 ';'4 '];
```

```
  ccps = [0;391.6;272.5;160.6;95.2;41.9];
  for m = 1:6,
    repeat = 1;
    while repeat
      h = msgbox(strcat('Click start & end points for flow=',scale(m,:)));
      waitfor(h,'ButtonDownFcn');
      [x,y] = ginput(2);
         x1 = round((x(1)-1).*10+1);
       x2 = round((x(2)-1).*10+1);
      subplot(2,1,1)
      plot([x(1) x(1)],[-10 10],'r',[x(2) x(2)],[-10 10],'r')
      subplot(2,1,2)
      plot([x(1) x(1)],[-10 10],'r',[x(2) x(2)],[-10 10],'r')
      soundsc(audio(x1:x2),fs);
      a = questdlg('Are those the points you want?');
      if length(char(a)) == 3
        repeat = 0;
        Vflow(m) = mean(flow(x1:x2));
      end
    end
  end
  figure(2)
  clf
  plot(ccps,Vflow,'ro')
  v2f = Vflow\ccps;             % v2f = voltage to flow factor, assumes flow=0 @ V=0
  hold
  plot(v2f*[0:0.1:2],0:0.1:2,'b-')
  text(30,1.8,strcat('cc/s=',num2str(v2f),'V_{flow}'));
  grid
  title('Mask voltage to flow transfer function')
  ylabel('Mask voltage (V_{flow})')
  xlabel('Actual flow (cc/s)')
  [infile,inpath] = uigetfile('*.abf','Enter the next filename for analysis.');
else
  if length(char(gainchange)) == 3
    changehere = questdlg(gainquest2,'Signal Gains','Yes','No','No');
    if length(char(changehere)) == 3
      v2adelta = inputdlg(gainquest3);
      v2a = v2a.*str2num(char(v2adelta));
      v2pdelta = inputdlg(gainquest4);
      v2p = v2p.*str2num(char(v2pdelta));
      v2fdelta = inputdlg(gainquest5);
      v2f = v2f.*str2num(char(v2fdelta));
    end
  end
  subplot(4,1,1)
  plot(accel(1:10:xmax))
  axis([1 xmax/10 -10 10])
  hold on
  ylabel('accel')
  subplot(4,1,2)
  plot(audio(1:10:xmax))
  axis([1 xmax/10 -10 10])
  hold on
  ylabel('audio')
```

```
    subplot(4,1,3)
    plot(flow(1:10:xmax))
    axis([1 xmax/10 -10 10])
    hold on
    ylabel('flow')
    subplot(4,1,4)
    plot(EGGtrack(1:10:xmax))
    axis([1 xmax/10 -10 10])
    hold on
    ylabel('EGG track')
    repeat = 1;
    while repeat
      h = msgbox('Click start & end points for next portion to edit & save.');
      waitfor(h,'ButtonDownFcn');
      [x,y] = ginput(2);
      x1 = round((x(1)-1).*10+1);
         x2 = round((x(2)-1).*10+1);
      for m = 1:4,
        subplot(4,1,m)
        plot([x(1) x(1)],[-10 10],'r',[x(2) x(2)],[-10 10],'r')
      end
      soundsc(audio(x1:x2),fs);
      a = questdlg('Are those the points you want?');
      if length(char(a)) == 3
        repeat = 0;
        flo = flow(x1+8:x2+8).*v2f;                % Get edited flow in cc/s
        aco = audio(x1+11:x2+11).*v2p;             % Get edited audio in dyne/cm^2
        acc = accel(x1:x2).*v2a;                   % Get edited accel in cm/s^2
        egg = EGGtrack(x1-2:x2-2);                     % Get edited EGG tracking signal
        outfile = inputdlg('Enter the name of the MATLAB file to be saved.');
        save(char(outfile),'flo','aco','acc','egg','fs');
        a = questdlg('Do these data need more editing & saving?');
           if length(char(a)) == 3
           repeat = 1;
        elseif length(char(a)) == 2
          a = questdlg('Is there another file to be analyzed?');
          if length(char(a)) == 3
               [infile,inpath] = uigetfile('*.abf','Enter the next filename for analysis.');
          else another = 0;
          end
           end
      end
    end
  end
end
```

```
function VVG = acc2vvg(acc,fs,h,t,wa,wcn)

% FUNCTION ACC2VVG.M - Acceleration to glottal volume velocity calculation
%
% Usage:  VVG = acc2vvg(acc,fs,h,t,wcn)
% Input:  acc, the measured acceleration at the neck (cm/sec^2)
%         fs, the sampling frequency (Hz)
```

```
%       h, the graphic object handles created by ACCFLO.M
%       t, the time values for each sample of accel (sec)
%       wa, the window for use when calculating the acceleration FFT
%       wcn, the window center index
% Output: VVG, the FFT of the acceleration-derived glottal volume velocity
%
% adapted from INVFILTAT3.M
% Harold Cheyne
% 4 April 2001 - original invfiltat3.m creation
% revised 23 May 2001 - added high-pass filter to tracheal wall model
% revised 25 May 2001 - changed HPF in tracheal wall model to 4th-order Butterworth
% revised 30 May 2001 - altered to acc2vvg.m
% revised 25 October 2001 - changed DVVG = 2.*fft(dvvga.*wa./2,n)./n; to
%   DVVG = 2.*fft(dvvga.*wa,n)./n; The factor of 1/2 is not correct!

Spf = str2double(get(h([28 32]),'String'));    % Get F1' and F2' frequencies
Spbw = str2double(get(h([29 33]),'String')); % Get F1' and F2' bandwidths
Szf = str2double(get(h([30 34]),'String'));    % Get Z1' and Z2' frequencies
Szbw = str2double(get(h([31 35]),'String')); % Get Z1' and Z2' bandwidths
n = str2double(get(h(20),'String'));            % Get FFT length from GUI.
f = 0:fs/n:fs/2-fs/n;

% ============== Tracheal wall model ==================
% The tracheal wall model includes a resistance and a mass.  It ignores the
% compliance of the wall because past reports of tracheal wall input impedance
% and vocal tract wall input impedance show the wall's resonance to be below
% 80 Hz.  It converts the acceleration to pressure in the tracheal deep to the
% accelerometer.
Mm = str2double(get(h(39),'String'));           % Get model mass value from GUI.
Rm = str2double(get(h(40),'String'));           % Get model resistance from GUI.
A = 0.8.*0.56;                                          % A = area under accelerometer
zp1 = 1;                                                % z-domain pole at z = 1, and zero at
zz1 = cos(2.*pi.*Rm./(Mm.*fs)) + j.*sin(2.*pi.*Rm./(Mm.*fs)); % e^(j*2*pi*Rm/(Mm*fs))
[b1,a1] = zp2tf(zz1,zp1,1);                     % 1st estimate of digital filter
H1 = freqz(b1,a1,[1 3000],fs);                  % Get filter response at 1, 3000 Hz
G1 = Mm./(A.*abs(H1(2)));                               % Set gain at 3000 Hz to Mm/A
[b2,a2] = zp2tf(zz1,zp1,G1)                         % re-calculate filter with gain G1
wla = length(wa);
pa = filter(b2,a2,acc(wcn-wla/2:wcn+wla/2-1)); % Filter acceleration signal.
% The following is the filter from 23 May 2001.
% pa2 = filter([1 -1],[1.0031 -0.9969],pa);     % High-pass pa at 20 Hz.
[b3,a3] = butter(4,75/1e4,'high');              % 4th-order Butterworth HPF with Fc=75Hz
pa2 = filter(b3,a3,pa);                             % HPF the estimated pressure signal
% ========== End of tracheal wall model ===============

[pz,zz] = s2z(Szf,Szbw,Spf,Spbw,fs);         % For inverse filter, swap poles & zeros
[b4,a4] = zp2tf(zz,pz,1);                       % Get transfer fxn with gain = 1
H4 = freqz(b4,a4,[0 1],fs);
G4 = 10^(50/20)./abs(H4(1));
[b5,a5] = zp2tf(zz,pz,G4)
dvvga = filter(b5,a5,pa2);
DVVG = 2.*fft(dvvga.*wa,n)./n;
VVG = DVVG(1:n/2)./(j.*2.*pi.*f);              % Divide by jw to get Vvg spectrum

% Plot results.
```

```
subplot(h(5))
plot(f,20.*log10(abs(VVG(1:n/2))),'k-')
grid
xlim([0 3000])
xlabel('Frequency (Hz)')
ylabel('|V_{vg}| from a(t), cm^3/sec');
title('V_{vg} estimate derived from acceleration');
function VVG = acc2vvg(acc,fs,h,t,wa,wcn)
```

```
% M-File ACCFLO3.M - ACCeleration and FLOw data analysis with Spectra
% and using a mass+resistance model of the tracheal wall & skin
% adapted from m-file LPLV.M
%
% Harold Cheyne
% 21 December 2000
% revised 5 June 2001
% revised 28 August 2001 for producing thesis document-compatible plots
% (i.e., black & white)
% revised 17 September 2001 to include Coherent Power Gain (i.e., gain
% due to Hanning/Hamming windows)

more off
[fname,fpath] = uigetfile('*.mat','Enter the filename for the analysis.');
if fname ~= 0
  load(strcat(fpath,fname));
  lacc = length(acc);                        % lacc = length of data (samples)
  wn = 1024;                                     % wn = window length for fft (1024=default)
  w = zeros(1,wn);                               % Initialize the window vector.
  w(1:wn-1) = hanning(wn-1,'periodic').*2;  % Use a Hanning window for the spectra
  % Note that the factor of 2 in the Hanning window corrects for the window's gain.
  wcn = round(lacc./2);                      % Initialize wcn = window center index
  n1 = wcn - wn/2 + 1;                           % n1 = window start point
    n2 = n1 + wn - 1;                            % n2 = window end point
  x = acc - mean(acc);                       % operate on the AC acceleration only
  y = egg;                                           % rename egg signal for functions below
  z = flo;                                           % rename flow signal for functions below
  afn = zeros(4,2);                              % Initialize amp. & freq. output vector.
  t = 0:1/fs:(lacc-1)/fs;                    % Create a time index for the signal.
  n = 2048;                                      % n = length of fft
  S = ones(n,1);                                 % Initialize spectrum vector.
  f = 0:fs/n:fs/2-fs/n;                   % Create the frequency vector for the spectra
  fmax = round(3000.*n./fs);                 % fmax = highest freq. index to display
  in = 0;                                        % in = boolean indicating zoomed/not zoomed
  dc = 0;                                        % dc = boolean indicating use/don't use d(cm)
  lp = 1;                                        % lp = boolean indicating LPF dVvg/dt or not
  dvvga = [];
  dVvg = [];
  h = zeros(69,1);                           % Initialize handle vector for all handles.

  % Create the figure for the GUI.
  figure(1)
  clf
  set(1,'Renderer','OpenGL','Position',[1 29 1152 768],'Resize','off')
```

```
  % Upper left subplot for the acceleration spectrum.
  h(1) = subplot(3,3,1);                         % h(1) = acceleration spectrum
  A = 2.*fft(x(n1:n2).*w,n)./n;                 % A = spectrum of acceleration
  plot(f,20.*log10(abs(A(1:n./2))),'k-');% for thesis document only
%  plot(f,20.*log10(abs(A(1:n./2))));
  grid
  xlim([0 3000])
  ytickmin = round(min(20.*log10(abs(A(1:fmax))))./10).*10;
    ytickmax = round(max(20.*log10(abs(A(1:fmax))))./10).*10;
    set(gca,'YTick',ytickmin:10:ytickmax);
    xlabel('Frequency (Hz)')
  ylabel('dB re 1 cm/sec^2')
  title(strcat(fname,',AC acceleration signal'));

  % Upper right plot for flow spectrum.
  h(2) = subplot(3,3,3);                         % h(2) = airflow spectrum
  F = 2.*fft(z(n1:n2).*w,n)./n;                 % F = spectrum of airflow
  plot(f,20.*log10(abs(F(1:n./2))),'k-');% for thesis document plots only
%  plot(f,20.*log10(abs(F(1:n./2))));
  grid
  xlim([0 3000])
  ytickmin = round(min(20.*log10(abs(F(1:fmax))))./10).*10;
    ytickmax = round(max(20.*log10(abs(F(1:fmax))))./10).*10;
    set(gca,'YTick',ytickmin:10:ytickmax);
    xlabel('Frequency (Hz)')
  ylabel('dB re 1 cm^3/sec')
  title(strcat(fname,',airflow signal'));

  % Middle left subplot for the acceleration time signal.
  h(3) = subplot(6,3,7);                         % h(3) = half-size plot showing
  plot(t,x,'k-');                                % This line only for thesis document plots
%  plot(t,x,'c-');                               % acceleration signal with window center
hold on                                            % line and window outline
    % The next line is only for thesis document plots.
    line([wcn/fs wcn/fs],[min(x) max(x)],'Color',[0 0 0],'LineWidth',2)
%   line([wcn/fs wcn/fs],[min(x) max(x)],'Color',[1 0 0],'LineWidth',2)
  grid
  xlabel('Time (sec)');
  ylabel('acc(t), cm/sec^2');
  axis([0 t(length(t)) min(x) max(x)])

  % Middle right subplot for the airflow time signal.
  h(4) = subplot(6,3,9);                         % h(4) = half-size plot showing airflow
  plot(t,z,'k-');                                % This line only for thesis document plots
%  plot(t,z,'c-');                               % time signal with window center line
    hold on                                        % and window outline
    % The next line is only for thesis document plots.
    line([wcn/fs wcn/fs],[min(x) max(x)],'Color',[0 0 0],'LineWidth',2)
%   line([wcn/fs wcn/fs],[min(z) max(z)],'Color',[1 0 0],'LineWidth',2)
  grid
  xlabel('Time (sec)');
  ylabel('v_{vm}, cm^3/sec');
  axis([0 t(length(t)) min(z) max(z)])
```

```
% Lower left plot for the acceleration-derived glottal flow derivative.
h(5) = subplot(3,3,7);
   grid
xlabel('Time (sec)')
ylabel('dV_{vg}/dt from a(t), litre/sec^2')

% Lower right plot for the first derivative of the glottal flow.
h(6) = subplot(3,3,9);                              % h(6) = 1st derivative of inv. filt. flow
grid

% Middle center plot for the inverse filtered flow signal.
h(7) = subplot(3,3,5);
   grid
   xlabel('Time (sec)');
ylabel('V_{vg}, cm^3/sec');
title('Inverse-filtered flow signal')

% Middle lower plot for the spectra of dVvga/dt and dVvgf/dt.
h(8) = subplot(3,3,8);
grid
xlabel('Frequency(Hz)')
ylabel('dB re 1 litre/sec^2')

% Create the spectral measurement interactive display.

% buttonstyles holds the text names of all the style types for the GUI buttons.
% stylelist holds the number of the corresponding style to put in buttonstyles:
% 1 = none(plot), 2 = pushbutton, 3 = text, 4 = edit, 5 = checkbox
stylelist = [1 1 1 1 1 1 1 1 2 2 2 2 3 4 2 2 3 3 3 3 3 3 4 4 4 4 4 4 4 2 3 3];
stylelist(33:62) = [3 3 3 3 4 4 4 4 4 4 4 4 2 2 3 2 3 4 2 3 4 3 4 3 4 3 4 3 3 4];
stylelist(63:69) = [2 3 3 3 2 4 2];
for m = 1:69,
  switch stylelist(m)
  case 1,    buttonstyles(m,1:10) = '          ';
  case 2,    buttonstyles(m,1:10) = 'pushbutton';
  case 3,    buttonstyles(m,1:10) = 'text      ';
  case 4,    buttonstyles(m,1:10) = 'edit      ';
  end
end
% buttonstring holds all of the strings of the buttons, text, and editable
% text to be created in the figure.  The first six rows are blanks so that the
% index for the handles h is the same as that for buttonstring.
stringblank = [1:8 23:29];
for m = 1:length(stringblank),
  buttonstring(stringblank(m),1:25) = blanks(25);
end
buttonstring(9:10,1:25) = ['Play entire or' blanks(11);'windowed signal' blanks(10)];
buttonstring(11:12,1:25) = ['Move cursor' blanks(14);'Zoom' blanks(21)];
buttonstring(13:14,1:25) = ['a(t) window size =' blanks(7);'2048' blanks(21)];
buttonstring(15:16,1:25) = ['L.P. Go!' blanks(17);'I.F. Go!' blanks(17)];
buttonstring(17:18,1:25) = ['Freq(Hz)' blanks(17);'BW(Hz)' blanks(19)];
buttonstring(19:20,1:25) = ['F1' blanks(23);'F2' blanks(23)];
buttonstring(21:22,1:25) = ['F3' blanks(23);'Vocal tract length (cm) ='];
buttonstring(30:31,1:25) = ['Get dVvga/dt' blanks(13);'Freq(Hz)' blanks(17)];
buttonstring(32:33,1:25) = ['BW(Hz)' blanks(19); 'F1''' blanks(22)];
```

```
buttonstring(34:35,1:25) = ['Z1''' blanks(22);'F2''' blanks(22)];
buttonstring(36:37,1:25) = ['Z2''' blanks(22);'640' blanks(22)];
buttonstring(38:39,1:25) = ['478' blanks(22);'1035' blanks(21)];
buttonstring(40:41,1:25) = ['150' blanks(22);'1400' blanks(21)];
buttonstring(42:43,1:25) = ['400' blanks(22);'1975' blanks(21)];
buttonstring(44:45,1:25) = ['250' blanks(22);'Get dVvgf/dt' blanks(13)];
buttonstring(46:47,1:25) = ['Get acc MFDR' blanks(13);'MFDR =' blanks(19)];
buttonstring(48:49,1:25) = ['Done' blanks(21);'flow window size =' blanks(7)];
buttonstring(50:51,1:25) = ['2048' blanks(21);'Print' blanks(20)];
buttonstring(52:53,1:25) = ['R' blanks(24);'0' blanks(24)];
buttonstring(54:55,1:25) = ['Mm (gm)' blanks(18);'1' blanks(24)];
buttonstring(56:57,1:25) = ['FFT length =' blanks(13);'2048' blanks(21)];
buttonstring(58:59,1:25) = ['Vvg display size =' blanks(7);'512' blanks(22)];
buttonstring(60:61,1:25) = ['Tracheal wall' blanks(12);'dVvg/dt display size =   '];
buttonstring(62:63,1:25) = ['1024' blanks(21);'Get flow MFDR' blanks(12)];
buttonstring(64:65,1:25) = ['MFDR =' blanks(19);'T(f)' blanks(21)];
buttonstring(66:67,1:25) = ['Zsgt' blanks(21);'Get F0' blanks(19)];
buttonstring(68:69,1:25) = ['0' blanks(24);'Load more data' blanks(11)];
% buttonpos holds all of the positions of the buttons, text, and editable
% text to be created in the figure. The first six rows are zeros so that the
% index for the handles h is the same as that for buttonpos.
buttonpos=zeros(69,4);
buttonpos(9:12,:)=[421 681 75 30;496 681 90 30;596 681 75 30;681 681 60 30];
buttonpos(13:16,:)=[421 651 105 20;526 651 45 20;771 311 50 25;771 346 50 25];
buttonpos(17:20,:)=[831 331 40 20;831 311 40 20;871 351 40 20;911 351 40 20];
buttonpos(21:24,:)=[951 351 40 20;1001 331 60 40;871 331 40 20;871 311 40 20];
buttonpos(25:28,:)=[911 331 40 20;911 311 40 20;951 331 40 20;951 311 40 20];
buttonpos(29:32,:)=[1001 311 60 20;101 271 100 30;101 331 40 20;101 311 40 20];
buttonpos(33:36,:)=[141 351 40 20;181 351 40 20;221 351 40 20;261 351 40 20];
buttonpos(37:40,:)=[141 331 40 20;141 311 40 20;181 331 40 20;181 311 40 20];
buttonpos(41:44,:)=[221 331 40 20;221 311 40 20;261 331 40 20;261 311 40 20];
buttonpos(45:48,:)=[771 271 100 30;211 271 80 30;301 271 100 30;421 571 150 30];
buttonpos(49:52,:)=[581 651 105 20;686 651 45 20;421 611 150 30;311 311 50 20];
buttonpos(53:56,:)=[361 311 40 20;311 331 50 20;361 331 40 20;581 621 105 20];
buttonpos(57:60,:)=[686 621 45 20;581 591 105 20;686 591 45 20;311 351 90 20];
buttonpos(61:64,:)=[581 561 105 20;686 561 45 20;881 271 80 30;971 271 100 30];
buttonpos(65:68,:)=[831 351 40 20;101 351 40 20;581 526 105 25;686 526 45 25];
buttonpos(69,:)=[421 531 150 30];
% buttoncall holds all of the callbacks of the buttons, text, and editable
% text to be created in the figure. The first six rows are blanks so that the
% index for the handles h is the same as that for buttoncall.
callblank = [1:8 13 17:22 31:44 47 49 50 52:62 65:66 68];
for m = 1:length(callblank),
  buttoncall(callblank(m),1:51) = blanks(51);
end
buttoncall(9,1:51) = ['soundsc(aco,fs);' blanks(35)];
buttoncall(10,1:51) = ['soundsc(aco(n1:n2),fs);                            '];
buttoncall(11,1:51) = ['wcn=newtime3(x,z,fname,fs,h,t);                    '];
buttoncall(12,1:51) = ['in=zoominout3(fname,fs,h,in,t,wcn,x,z);            '];
buttoncall(15,1:51) = ['fbw=getfbl3(z,fs,h,wcn);                           '];
buttoncall(16,1:51) = ['[vg,vglp]=invfilt3(fbw,z,1250,fname,fs,h,in,t,wcn);'];
buttoncall(27,1:51) = ['setF3orL(str2double(get(h(27),''String'')),[],h);    '];
buttoncall(29,1:51) = ['setF3orL([],str2double(get(h(29),''String'')),h);    '];
buttoncall(30,1:51) = ['dvvga=invfiltat3(x,dVvg,fs,h,t,wcn);               '];
buttoncall(45,1:51) = ['dVvg=dVdtplot3(z,dvvga,fname,fs,h,in,vglp,t,wcn);  '];
```

```
  buttoncall(46,1:51) = ['getmfdr3(1,dvvga,fs,h,t,wcn);                    '];
  buttoncall(48,1:51) = ['clear;close(1);more on;                          '];
  buttoncall(51,1:51) = ['print(1,''-dmeta'',''-zbuffer'');                '];
  buttoncall(63,1:51) = ['getmfdr3(2,dVvg,fs,h,t,wcn);                     '];
  buttoncall(67,1:51) = ['getF0(h);                                        '];
  buttoncall(69,1:51) = ['[A,x,y,z,aco,fname,fs,F,t,wa,wcn,wf]=newdata3(h); '];
  for m = 9:69,                                      % Create all of the buttons & text
    h(m) = uicontrol('Style',strcat(buttonstyles(m,:)),...
      'String',strcat(buttonstring(m,:)),...
      'Position',buttonpos(m,:),...
      'Callback',strcat(buttoncall(m,:)),...
      'BackgroundColor',[1 1 1]);
  end
  set(h(60),'FontWeight','bold');
  set(h(65),'FontWeight','bold');
  set(h(66),'FontWeight','bold');
end
```

```
% M-File ACCFLO4.M - ACCeleration and FLOw data analysis
% for comparing the estimated F1' bandwidth to the estimated minimum flow,
% adapted from m-file ACCSPL1.M
%
% Harold Cheyne
% 26 June 2001
% revised 12 July 2001

more off
h = zeros(39,1);                              % Initialize handle vector for all handles.

% Create the figure for the GUI.
figure(1)
clf
set(1,'Renderer','OpenGL','Position',[1 29 1152 768],'Resize','off')
sp = [3 2 1;3 2 2;6 2 5;6 2 6;3 2 5;3 2 6];
xaxis = ['Frequency (Hz)';'Frequency (Hz)';'Time (sec)    ';'Time (sec)    '];
xaxis(5:6,1:14) = ['Time (sec)    ';'Time (sec)    '];
yaxis = ['|ACC|, dB re 1 cm/sec^2      ';'|V_{vm}|, dB re 1 cm^3/sec  '];
yaxis(3:4,1:29) = ['-acc(t), cm/sec^2            ';'v_{vm}(t), cm^3/sec          '];
yaxis(5:6,1:29) = ['BP filtered -acc(t), cm/sec^2';'v_{vg}(t) estimate, cm^3/sec '];
titletext(1:4,1:40) = [blanks(40);blanks(40);blanks(40);blanks(40)];
titletext(5,1:40) = 'Band-pass filtered inverted acceleration';
titletext(6,1:40) = 'Inverse-filtered flow                   ';
for m = 1:6,
  h(m) = subplot(sp(m,1),sp(m,2),sp(m,3));
  grid
  xlabel(strcat(xaxis(m,:)));
  ylabel(strcat(yaxis(m,:)));
  title(strcat(titletext(m,:)));
end

% Create the spectral measurement interactive display.

% buttonstyles holds the text names of all the style types for the GUI buttons.
```

```
% stylelist holds the number of the corresponding style to put in buttonstyles:
% 1 = none(plot), 2 = pushbutton, 3 = text, 4 = edit, 5 = checkbox
stylelist = [1 1 1 1 1 1 3 3 4 3 4 2 2 2 3 2 2 2 2 3 4 3 4 2 2 3 3 3 3 3 3];
stylelist(32:39) = [4 4 4 4 4 4 2 3];
for m = 1:39,
  switch stylelist(m)
  case 1,buttonstyles(m,1:10) = '          ';
  case 2,buttonstyles(m,1:10) = 'pushbutton';
  case 3,buttonstyles(m,1:10) = 'text      ';
  case 4,buttonstyles(m,1:10) = 'edit      ';
  end
end

% buttonstring holds all of the strings of the buttons, text, and editable
% text to be created in the figure.  The first six rows are blanks so that the
% index for the handles h is the same as that for buttonstring.
stringblank = [1:6 32:37];
for m = 1:length(stringblank),
  buttonstring(stringblank(m),1:25) = blanks(25);
end
buttonstring(7:8,1:25)=['BPF cutoffs' blanks(14);'lower' blanks(20)];
buttonstring(9:11,1:25)=['370' blanks(22);'upper' blanks(20);'770' blanks(22)];
buttonstring(12:13,1:25)=['BPF acc(t)' blanks(15);'Estimate B1''' blanks(13)];
buttonstring(14:15,1:25)=['Load data' blanks(16);'B1''=' blanks(21)];
buttonstring(16:17,1:25)=['Print' blanks(20);'Done' blanks(21)];
buttonstring(18:19,1:25)=['Move cursor' blanks(14);'Zoom' blanks(21)];
buttonstring(20:21,1:25)=['acc(t) window size =     ';'1024' blanks(21)];
buttonstring(22:23,1:25)=['flow window size =       ';'1024' blanks(21)];
buttonstring(24:25,1:25)=['Linear Prediction        ';'Inverse filter flow      '];
buttonstring(26:28,1:25)=['T(f)' blanks(21);'F1' blanks(23);'F2' blanks(23)];
buttonstring(29:31,1:25)=['F3' blanks(23);'Freq(Hz)' blanks(17);'BW(Hz)' blanks(19)];
buttonstring(38:39,1:25)=['Get min. flow' blanks(12);'Min. flow=' blanks(15)];

% buttonpos holds all of the positions of the buttons, text, and editable
% text to be created in the figure.  The first six rows are zeros so that the
% index for the handles h is the same as that for buttonpos.
buttonpos=zeros(62,4);
buttonpos(7:10,:)=[111 341 80 20;111 321 40 20;151 321 40 20;111 301 40 20];
buttonpos(11:14,:)=[151 301 40 20;201 336 75 25;201 301 75 25;286 336 75 25];
buttonpos(15:18,:)=[286 301 75 25;371 336 75 25;371 301 75 25;456 336 75 25];
buttonpos(19:22,:)=[456 301 75 25;541 336 100 25;641 336 40 25;541 301 100 25];
buttonpos(23:26,:)=[641 301 40 25;691 336 100 25;691 301 100 25;801 341 40 20];
buttonpos(27:30,:)=[841 341 40 20;881 341 40 20;921 341 40 20;801 321 40 20];
buttonpos(31:34,:)=[801 301 40 20;841 321 40 20;841 301 40 20;881 321 40 20];
buttonpos(35:38,:)=[881 301 40 20;921 321 40 20;921 301 40 20;971 336 75 25];
buttonpos(39,:)=[971 301 75 25];

% buttoncall holds all of the callbacks of the buttons, text, and editable
% text to be created in the figure.  The first seven rows are blanks so that the
% index for the handles h is the same as that for buttoncall.
callblank = [1:13 20:39];
for m = 1:length(callblank),
  buttoncall(callblank(m),1:51) = blanks(51);
end
buttoncall(12,1:51) = ['y=bpfacc(acc,fs,h,t,wcn);                          '];
```

```
buttoncall(13,1:51) = ['getbfl(y,fs,h,t,wcn);                    '];
buttoncall(14,1:51) = ['[A,acc,F,flo,fname,fs,t,wa,wcn,wf]=newdata4(h);   '];
buttoncall(16,1:51) = ['print(1,''-dmeta'',''-zbuffer'');              '];
buttoncall(17,1:51) = ['clear;close(1);more on;                    '];
buttoncall(18,1:51) = ['wcn=newtime6(acc,flo,fname,fs,h,t);          '];
buttoncall(19,1:51) = ['zoominout6(acc,flo,fname,fs,h,t,wcn);          '];
buttoncall(21,1:51) = ['newwindow(acc,flo,fname,fs,h,t,wcn);          '];
buttoncall(23,1:51) = ['newwindow(acc,flo,fname,fs,h,t,wcn);          '];
buttoncall(24,1:51) = ['getfb2(flo,fs,h,wcn);                    '];
buttoncall(25,1:51) = ['[vg,vglp]=invfilt4(flo,1250,fs,h,t,wcn);       '];
buttoncall(38,1:51) = ['getminflo(vglp,fs,h,t,wcn);                 '];
for m = 7:39,                                   % Create all of the buttons & text
  h(m) = uicontrol('Style',strcat(buttonstyles(m,:)),...
    'String',strcat(buttonstring(m,:)),...
    'Position',buttonpos(m,:),...
    'Callback',strcat(buttoncall(m,:)));
end
set(h(26),'FontWeight','bold');
%set(h(36),'FontWeight','bold');
%set(h(43),'FontWeight','bold');
```

```
% M-File ACCSPL1.M - ACCeleration and SPL data analysis with Spectra
% and using a mass+resistance model of the tracheal wall & skin,
% adapted from m-file ACCFLO3.M
%
% Harold Cheyne
% ACCFLO3.M - 21 December 2000, revised 21 May 2001
% revised 31 May 2001

more off
h = zeros(62,1);                                % Initialize handle vector for all handles.

% Create the figure for the GUI.
figure(1)
clf
set(1,'Renderer','OpenGL','Position',[1 29 1152 768],'Resize','off')
sp = [3 3 1;3 3 3;6 3 7;6 3 9;3 3 7;3 3 8;3 3 9];
xaxis = ['Frequency (Hz)';'Frequency (Hz)';'Time (sec)    ';'Time (sec)    '];
xaxis(5:7,1:14) = ['Frequency (Hz)';'Frequency (Hz)';'Frequency (Hz)'];
yaxis = ['|ACC|, dB re 1 cm/sec^2    ';'|P_{mic}|, dB re 1 dyne/cm^2'];
yaxis(3:4,1:28) = ['acc(t), cm/sec^2            ';'p_{mic}, dyne/cm^2          '];
yaxis(5:6,1:28) = ['|V_{vg}| from a(t), cm^3/sec';'|V_{vm}/V_{vg}|, dB         '];
yaxis(7,1:28) = 'dB re 1 dyne/cm^2           ';
titletext(1:4,1:44) = [blanks(44);blanks(44);blanks(44);blanks(44)];
titletext(5,1:44) = 'V_{vg} estimate derived from acceleration   ';
titletext(6,1:44) = 'Model vocal tract transfer function spectrum';
titletext(7,1:44) = blanks(44);
for m = 1:7,
  h(m) = subplot(sp(m,1),sp(m,2),sp(m,3));
  grid
  xlabel(strcat(xaxis(m,:)));
  ylabel(strcat(yaxis(m,:)));
  title(strcat(titletext(m,:)));
```

```
end

% Create the spectral measurement interactive display.

% buttonstyles holds the text names of all the style types for the GUI buttons.
% stylelist holds the number of the corresponding style to put in buttonstyles:
% 1 = none(plot), 2 = pushbutton, 3 = text, 4 = edit, 5 = checkbox
stylelist = [1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 3 4 3 4 3 4 3 3 3 3 3 3 3 4 4 4 4 4 4 4];
stylelist(36:62) = [3 3 3 4 4 2 2 3 3 3 3 3 3 3 4 3 4 3 4 3 4 2 2 3 3 3 3];
for m = 1:62,
  switch stylelist(m)
  case 1, buttonstyles(m,1:10) = '          ';
  case 2, buttonstyles(m,1:10) = 'pushbutton';
  case 3, buttonstyles(m,1:10) = 'text      ';
  case 4, buttonstyles(m,1:10) = 'edit      ';
  end
end

% buttonstring holds all of the strings of the buttons, text, and editable
% text to be created in the figure. The first seven rows are blanks so that the
% index for the handles h is the same as that for buttonstring.
stringblank = [1:7 61:62];
for m = 1:length(stringblank),
  buttonstring(stringblank(m),1:25) = blanks(25);
end
buttonstring(8:9,1:25)=['Play entire or' blanks(11);'windowed signal' blanks(10)];
buttonstring(10:11,1:25)=['Move cursor' blanks(14);'Zoom' blanks(21)];
buttonstring(12:13,1:25)=['Print' blanks(20);'Done' blanks(21)];
buttonstring(14:15,1:25)=['Load data' blanks(16);'acc(t) window size =     '];
buttonstring(16:17,1:25)=['2048' blanks(21);'aco(t) window size =     '];
buttonstring(18:19,1:25)=['2048' blanks(21);'FFT length =' blanks(13)];
buttonstring(20:22,1:25)=['2048' blanks(21);'Zsgt' blanks(21);'F1 ''' blanks(22)];
buttonstring(23:25,1:25)=['Z1 ''' blanks(22);'F2 ''' blanks(22);'Z2 ''' blanks(22)];
buttonstring(26:28,1:25)=['Freq(Hz)' blanks(17);'BW(Hz)' blanks(19);'640' blanks(22)];
buttonstring(29:31,1:25)=['478' blanks(22);'1035' blanks(21);'150' blanks(22)];
buttonstring(32:34,1:25)=['1400' blanks(21);'400' blanks(22);'1975' blanks(21)];
buttonstring(35:36,1:25)=['250' blanks(22);'Tracheal wall' blanks(12)];
buttonstring(37:39,1:25)=['Mm (gm)' blanks(18);'R' blanks(24);'1' blanks(24)];
buttonstring(40:41,1:25)=['0' blanks(24);'Get Vvga spectrum        '];
buttonstring(42:43,1:25)=['Show |Vvm/Vvg|' blanks(11);'T(f)' blanks(21)];
buttonstring(44:46,1:25)=['F1' blanks(23);'F2' blanks(23);'F3' blanks(23)];
buttonstring(47:49,1:25)=['Freq(Hz)' blanks(17);'BW(Hz)' blanks(19);'500' blanks(22)];
buttonstring(50:52,1:25)=['100' blanks(22);'1500' blanks(21);'150' blanks(22)];
buttonstring(53:54,1:25)=['2500' blanks(21);'200' blanks(22)];
buttonstring(55:56,1:25)=['Vocal tract length (cm) =';'17.7' blanks(21)];
buttonstring(57:58,1:25)=['Get estimated Pmic       ';'Get SPL from acc         '];
buttonstring(59:60,1:25)=['SPL from mic (dB) =      ';'SPL from acc (dB) =      '];
% buttonpos holds all of the positions of the buttons, text, and editable
% text to be created in the figure. The first seven rows are zeros so that the
% index for the handles h is the same as that for buttonpos.
buttonpos=zeros(62,4);
buttonpos(8:11,:)=[421 681 75 30;496 681 90 30;596 681 75 30;681 681 60 30];
buttonpos(12:15,:)=[421 641 100 30;531 641 100 30;641 641 100 30;421 611 110 20];
buttonpos(16:19,:)=[531 611 45 20;586 611 110 20;696 611 45 20;504 581 110 20];
buttonpos(20:23,:)=[614 581 45 20;101 341 40 20;141 341 40 20;181 341 40 20];
```

```
buttonpos(24:27,:)=[221 341 40 20;261 341 40 20;101 321 40 20;101 301 40 20];
buttonpos(28:31,:)=[141 321 40 20;141 301 40 20;181 321 40 20;181 301 40 20];
buttonpos(32:35,:)=[221 321 40 20;221 301 40 20;261 321 40 20;261 301 40 20];
buttonpos(36:39,:)=[311 341 90 20;311 321 50 20;311 301 50 20;361 321 40 20];
buttonpos(40:43,:)=[361 301 40 20;411 336 100 25;411 301 100 25;521 341 40 20];
buttonpos(44:47,:)=[561 341 40 20;601 341 40 20;641 341 40 20;521 321 40 20];
buttonpos(48:51,:)=[521 301 40 20;561 321 40 20;561 301 40 20;601 321 40 20];
buttonpos(52:55,:)=[601 301 40 20;641 321 40 20;641 301 40 20;691 321 60 40];
buttonpos(56:59,:)=[691 301 60 20;761 336 100 25;761 301 100 25;871 336 100 25];
buttonpos(60:62,:)=[871 301 100 25;971 336 40 25;971 301 40 25];
% buttoncall holds all of the callbacks of the buttons, text, and editable
% text to be created in the figure. The first seven rows are blanks so that the
% index for the handles h is the same as that for buttoncall.
callblank = [1:7 15:40 47:55 59:62];
for m = 1:length(callblank),
  buttoncall(callblank(m),1:51) = blanks(51);
end
buttoncall(8,1:51) = ['soundsc(aco,fs);' blanks(35)];
buttoncall(9,1:51) = ['soundsc(aco(n1:n2),fs);                    '];
buttoncall(10,1:51) = ['wcn = newtime4(acc,aco,fname,fs,h,t);         '];
buttoncall(11,1:51) = ['zoominout4(acc,aco,fname,fs,h,t,wcn);         '];
buttoncall(12,1:51) = ['print(1,''-dmeta'',''-zbuffer'');              '];
buttoncall(13,1:51) = ['clear;close(1);more on;                     '];
buttoncall(14,1:51) = ['[A,acc,aco,fname,fs,P,t,wa,wcn,wp]=newdata(h);   '];
buttoncall(41,1:51) = ['vvg=acc2vvg(acc,fs,h,t,wa,wcn);               '];
buttoncall(42,1:51) = ['Hvt=getHvt(fs,h);                           '];
buttoncall(56,1:51) = ['setF1F2F3(h);' blanks(38)];
buttoncall(57,1:51) = ['p=vvg2mic(fs,h,Hvt,vvg);                     '];
buttoncall(58,1:51) = ['getaccspl(fs,h,p);                         '];
for m = 8:62,                                  % Create all of the buttons & text
  h(m) = uicontrol('Style',strcat(buttonstyles(m,:)),...
    'String',strcat(buttonstring(m,:)),...
    'Position',buttonpos(m,:),...
    'Callback',strcat(buttoncall(m,:)));
end
set(h(21),'FontWeight','bold');
set(h(36),'FontWeight','bold');
set(h(43),'FontWeight','bold');
```

```
% M-File ACCSPL2.M - ACCeleration and SPL data analysis with Spectra
% and using a mass+resistance model of the tracheal wall & skin,
% adapted from m-file ACCFLO3.M, and allowing variable F1, F2 and F3
%
% Harold Cheyne
% ACCFLO3.M - 21 December 2000, revised 21 May 2001
% revised 31 May 2001
% revised 13 June 2001

more off
h = zeros(62,1);                              % Initialize handle vector for all handles.

% Create the figure for the GUI.
figure(1)
```

```
clf
set(1,'Renderer','OpenGL','Position',[1 29 1152 768],'Resize','off')
sp = [3 3 1;3 3 3;6 3 7;6 3 9;3 3 7;3 3 8;3 3 9];
xaxis = ['Frequency (Hz)';'Frequency (Hz)';'Time (sec)    ';'Time (sec)    '];
xaxis(5:7,1:14) = ['Frequency (Hz)';'Frequency (Hz)';'Frequency (Hz)'];
yaxis = ['|ACC|, dB re 1 cm/sec^2     ';'|P_{mic}|, dB re 1 dyne/cm^2'];
yaxis(3:4,1:28) = ['acc(t), cm/sec^2            ';'p_{mic}, dyne/cm^2          '];
yaxis(5:6,1:28) = ['|V_{vg}| from a(t), cm^3/sec';'|V_{vm}/V_{vg}|, dB         '];
yaxis(7,1:28) = 'dB re 1 dyne/cm^2           ';
titletext(1:4,1:44) = [blanks(44);blanks(44);blanks(44);blanks(44)];
titletext(5,1:44) = 'V_{vg} estimate derived from acceleration   ';
titletext(6,1:44) = 'Model vocal tract transfer function spectrum';
titletext(7,1:44) = blanks(44);
for m = 1:7,
  h(m) = subplot(sp(m,1),sp(m,2),sp(m,3));
  grid
  xlabel(strcat(xaxis(m,:)));
  ylabel(strcat(yaxis(m,:)));
  title(strcat(titletext(m,:)));
end

% Create the spectral measurement interactive display.

% buttonstyles holds the text names of all the style types for the GUI buttons.
% stylelist holds the number of the corresponding style to put in buttonstyles:
% 1 = none(plot), 2 = pushbutton, 3 = text, 4 = edit, 5 = checkbox
stylelist = [1 1 1 1 1 1 1 2 2 2 2 2 2 2 3 4 3 4 3 4 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4];
stylelist(36:62) = [3 3 3 4 4 2 2 3 3 3 3 3 3 4 4 4 4 4 4 3 4 2 2 3 3 3 3];
for m = 1:62,
  switch stylelist(m)
  case 1, buttonstyles(m,1:10) = '          ';
  case 2, buttonstyles(m,1:10) = 'pushbutton';
  case 3, buttonstyles(m,1:10) = 'text      ';
  case 4, buttonstyles(m,1:10) = 'edit      ';
  end
end

% buttonstring holds all of the strings of the buttons, text, and editable
% text to be created in the figure.  The first seven rows are blanks so that the
% index for the handles h is the same as that for buttonstring.
stringblank = [1:7 61:62];
for m = 1:length(stringblank),
  buttonstring(stringblank(m),1:25) = blanks(25);
end
buttonstring(8:9,1:25)=['Play entire or' blanks(11);'windowed signal' blanks(10)];
buttonstring(10:11,1:25)=['Move cursor' blanks(14);'Zoom' blanks(21)];
buttonstring(12:13,1:25)=['Print' blanks(20);'Done' blanks(21)];
buttonstring(14:15,1:25)=['Load data' blanks(16);'acc(t) window size =    '];
buttonstring(16:17,1:25)=['2048' blanks(21);'aco(t) window size =    '];
buttonstring(18:19,1:25)=['2048' blanks(21);'FFT length =' blanks(13)];
buttonstring(20:22,1:25)=['2048' blanks(21);'Zsgt' blanks(21);'F1''' blanks(22)];
buttonstring(23:25,1:25)=['Z1''' blanks(22);'F2''' blanks(22);'Z2''' blanks(22)];
buttonstring(26:28,1:25)=['Freq(Hz)' blanks(17);'BW(Hz)' blanks(19);'640' blanks(22)];
buttonstring(29:31,1:25)=['478' blanks(22);'1035' blanks(21);'150' blanks(22)];
buttonstring(32:34,1:25)=['1400' blanks(21);'400' blanks(22);'1975' blanks(21)];
```

```
buttonstring(35:36,1:25)=['250' blanks(22);'Tracheal wall' blanks(12)];
buttonstring(37:39,1:25)=['Mm (gm)' blanks(18);'R' blanks(24);'1' blanks(24)];
buttonstring(40:41,1:25)=['0' blanks(24);'Get Vvga spectrum        '];
buttonstring(42:43,1:25)=['Show |Vvm/Vvg|' blanks(11);'T(f)' blanks(21)];
buttonstring(44:46,1:25)=['F1' blanks(23);'F2' blanks(23);'F3' blanks(23)];
buttonstring(47:49,1:25)=['Freq(Hz)' blanks(17);'BW(Hz)' blanks(19);'500' blanks(22)];
buttonstring(50:52,1:25)=['100' blanks(22);'1500' blanks(21);'150' blanks(22)];
buttonstring(53:54,1:25)=['2500' blanks(21);'200' blanks(22)];
buttonstring(55:56,1:25)=['Vocal tract length (cm) =';'17.7' blanks(21)];
buttonstring(57:58,1:25)=['Get estimated Pmic       ';'Get SPL from acc         '];
buttonstring(59:60,1:25)=['SPL from mic (dB) =      ';'SPL from acc (dB) =      '];
% buttonpos holds all of the positions of the buttons, text, and editable
% text to be created in the figure.  The first seven rows are zeros so that the
% index for the handles h is the same as that for buttonpos.
buttonpos=zeros(62,4);
buttonpos(8:11,:)=[421 681 75 30;496 681 90 30;596 681 75 30;681 681 60 30];
buttonpos(12:15,:)=[421 641 100 30;531 641 100 30;641 641 100 30;421 611 110 20];
buttonpos(16:19,:)=[531 611 45 20;586 611 110 20;696 611 45 20;504 581 110 20];
buttonpos(20:23,:)=[614 581 45 20;101 341 40 20;141 341 40 20;181 341 40 20];
buttonpos(24:27,:)=[221 341 40 20;261 341 40 20;101 321 40 20;101 301 40 20];
buttonpos(28:31,:)=[141 321 40 20;141 301 40 20;181 321 40 20;181 301 40 20];
buttonpos(32:35,:)=[221 321 40 20;221 301 40 20;261 321 40 20;261 301 40 20];
buttonpos(36:39,:)=[311 341 90 20;311 321 50 20;311 301 50 20;361 321 40 20];
buttonpos(40:43,:)=[361 301 40 20;411 336 100 25;411 301 100 25;521 341 40 20];
buttonpos(44:47,:)=[561 341 40 20;601 341 40 20;641 341 40 20;521 321 40 20];
buttonpos(48:51,:)=[521 301 40 20;561 321 40 20;561 301 40 20;601 321 40 20];
buttonpos(52:55,:)=[601 301 40 20;641 321 40 20;641 301 40 20;691 321 60 40];
buttonpos(56:59,:)=[691 301 60 20;761 336 100 25;761 301 100 25;871 336 100 25];
buttonpos(60:62,:)=[871 301 100 25;971 336 40 25;971 301 40 25];
% buttoncall holds all of the callbacks of the buttons, text, and editable
% text to be created in the figure.  The first seven rows are blanks so that the
% index for the handles h is the same as that for buttoncall.
callblank = [1:7 15:40 47:56 59:62];
for m = 1:length(callblank),
  buttoncall(callblank(m),1:51) = blanks(51);
end
buttoncall(8,1:51) = ['soundsc(aco,fs);' blanks(35)];
buttoncall(9,1:51) = ['soundsc(aco(n1:n2),fs);                            '];
buttoncall(10,1:51) = ['wcn = newtime4(acc,aco,fname,fs,h,t);              '];
buttoncall(11,1:51) = ['zoominout4(acc,aco,fname,fs,h,t,wcn);              '];
buttoncall(12,1:51) = ['print(1,''-dmeta'',''-zbuffer'');                      '];
buttoncall(13,1:51) = ['clear;close(1);more on;                            '];
buttoncall(14,1:51) = ['[A,acc,aco,fname,fs,P,t,wa,wcn,wp]=newdata(h);     '];
buttoncall(41,1:51) = ['vvg=acc2vvg(acc,fs,h,t,wa,wcn);                    '];
buttoncall(42,1:51) = ['Hvt=getHvt(fs,h);                                  '];
buttoncall(57,1:51) = ['p=vvg2mic(fs,h,Hvt,vvg);                           '];
buttoncall(58,1:51) = ['getaccspl(fs,h,p);                                 '];
for m = 8:62,                                 % Create all of the buttons & text
  h(m) = uicontrol('Style',strcat(buttonstyles(m,:)),...
    'String',strcat(buttonstring(m,:)),...
    'Position',buttonpos(m,:),...
    'Callback',strcat(buttoncall(m,:)));
end
set(h(21),'FontWeight','bold');
set(h(36),'FontWeight','bold');
```

```
set(h(43),'FontWeight','bold');
```

```
% M-File ACCSPL3.M - ACCeleration and SPL data analysis with H1'-A2'
% adapted from m-file ACCSPL1.M
%
% Harold Cheyne
% 21 June 2001
% revised 22 June 2001

more off
h = zeros(47,1);                                   % Initialize handle vector for all handles.

% Create the figure for the GUI.
figure(1)
clf
set(1,'Renderer','OpenGL','Position',[1 1 612 792],'Resize','off','Color',[1 1 1])
sp = [5 3 1;5 3 4;5 3 7;5 3 10;5 3 13;10 3 3;10 3 6;10 3 9;10 3 12;10 3 15];
sp(11:15,:) = [10 3 18;10 3 21;10 3 24;10 3 27;10 3 30];
xaxis = [blanks(14);blanks(14);blanks(14);blanks(14);'Frequency (Hz)';blanks(14)];
xaxis(7:11,1:14) = [blanks(14);blanks(14);blanks(14);blanks(14);blanks(14);];
xaxis(12:15,1:14) = [blanks(14);blanks(14);blanks(14);'Time (sec)    '];
yaxis = ['dB re 1 cm/sec^2';'dB re 1 cm/sec^2';'dB re 1 cm/sec^2'];
yaxis(4:6,1:16) = ['dB re 1 cm/sec^2';'dB re 1 cm/sec^2';'cm/sec^2        '];
yaxis(7:9,1:16) = ['dyne/cm^2       ';'cm/sec^2        ';'dyne/cm^2       '];
yaxis(10:12,1:16) = ['cm/sec^2        ';'dyne/cm^2       ';'cm/sec^2        '];
yaxis(13:15,1:16) = ['dyne/cm^2       ';'cm/sec^2        ';'dyne/cm^2       '];
for m = 1:15,
  h(m) = subplot(sp(m,1),sp(m,2),sp(m,3));
  grid
  xlabel(strcat(xaxis(m,:)));
  ylabel(strcat(yaxis(m,:)));
end

% Create the spectral measurement interactive display.

% buttonstyles holds the text names of all the style types for the GUI buttons.
% stylelist holds the number of the corresponding style to put in buttonstyles:
% 1 = none(plot), 2 = pushbutton, 3 = text, 4 = edit, 5 = checkbox
stylelist = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 3 3 2 2 2 2 3 3 2 2 2 2 3 3];
stylelist(34:47) = [2 2 2 2 3 3 2 2 2 2 3 3 2 2];
for m = 1:47,
  switch stylelist(m)
  case 1,buttonstyles(m,1:10) = '          ';
  case 2,buttonstyles(m,1:10) = 'pushbutton';
  case 3,buttonstyles(m,1:10) = 'text      ';
  end
end

% buttonstring holds all of the strings of the buttons, text, and editable
% text to be created in the figure. The first 15 rows are blanks so that the
% index for the handles h is the same as that for buttonstring.
stringblank = [1:15];
for m = 1:length(stringblank),
```

```
  buttonstring(stringblank(m),1:11) = blanks(11);
end
buttonstring(16:18,1:11)=['Move      ';'unZoom     ';'Get H1",A2"'];
buttonstring(19:21,1:11)=['Load data  ';'dB SPL=    ';'H1"-A2"=   '];
buttonstring(22:24,1:11)=['Move      ';'unZoom     ';'Get H1",A2"'];
buttonstring(25:27,1:11)=['Load data  ';'dB SPL=    ';'H1"-A2"=   '];
buttonstring(28:30,1:11)=['Move      ';'unZoom     ';'Get H1",A2"'];
buttonstring(31:33,1:11)=['Load data  ';'dB SPL=    ';'H1"-A2"=   '];
buttonstring(34:36,1:11)=['Move      ';'unZoom     ';'Get H1",A2"'];
buttonstring(37:39,1:11)=['Load data  ';'dB SPL=    ';'H1"-A2"=   '];
buttonstring(40:42,1:11)=['Move      ';'unZoom     ';'Get H1",A2"'];
buttonstring(43:45,1:11)=['Load data  ';'dB SPL=    ';'H1"-A2"=   '];
buttonstring(46:47,1:11)=['Print      ';'Done       '];

% buttonpos holds all of the positions of the buttons, text, and editable
% text to be created in the figure.  The first 15 rows are zeros so that the
% index for the handles h is the same as that for buttonpos.
buttonpos=zeros(47,4);
buttonpos(16:19,:)=[221 701 40 30;271 701 60 30;221 661 70 30;301 661 70 30];
buttonpos(20:23,:)=[221 631 100 20;221 606 100 20;221 566 40 30;271 566 60 30];
buttonpos(24:27,:)=[221 526 70 30;301 526 70 30;221 496 100 20;221 471 100 20];
buttonpos(28:31,:)=[221 431 40 30;271 431 60 30;221 391 70 30;301 391 70 30];
buttonpos(32:35,:)=[221 361 100 20;221 336 100 20;221 296 40 30;271 296 60 30];
buttonpos(36:39,:)=[221 256 70 30;301 256 70 30;221 226 100 20;221 201 100 20];
buttonpos(40:43,:)=[221 161 40 30;271 161 60 30;221 121 70 30;301 121 70 30];
buttonpos(44:47,:)=[221 91 100 20;221 66 100 20;221 26 70 30;301 26 70 30];

% buttoncall holds all of the callbacks of the buttons, text, and editable
% text to be created in the figure.  The first 15 rows are blanks so that the
% index for the handles h is the same as that for buttoncall.
callblank = [1:15];
for m = 1:length(callblank),
  buttoncall(callblank(m),1:41) = blanks(41);
end
buttoncall(16,1:41) = ['[A,wcn]=newtime5(acc,aco,fname,fs,h,1,t);'];
buttoncall(22,1:41) = ['[A,wcn]=newtime5(acc,aco,fname,fs,h,2,t);'];
buttoncall(28,1:41) = ['[A,wcn]=newtime5(acc,aco,fname,fs,h,3,t);'];
buttoncall(34,1:41) = ['[A,wcn]=newtime5(acc,aco,fname,fs,h,4,t);'];
buttoncall(40,1:41) = ['[A,wcn]=newtime5(acc,aco,fname,fs,h,5,t);'];
buttoncall(17,1:41) = ['zoominout5(acc,aco,fname,fs,h,1,t,wcn);  '];
buttoncall(23,1:41) = ['zoominout5(acc,aco,fname,fs,h,2,t,wcn);  '];
buttoncall(29,1:41) = ['zoominout5(acc,aco,fname,fs,h,3,t,wcn);  '];
buttoncall(35,1:41) = ['zoominout5(acc,aco,fname,fs,h,4,t,wcn);  '];
buttoncall(41,1:41) = ['zoominout5(acc,aco,fname,fs,h,5,t,wcn);  '];
buttoncall(18,1:41) = ['geth1a2(A,fname,fs,h,1);                 '];
buttoncall(24,1:41) = ['geth1a2(A,fname,fs,h,2);                 '];
buttoncall(30,1:41) = ['geth1a2(A,fname,fs,h,3);                 '];
buttoncall(36,1:41) = ['geth1a2(A,fname,fs,h,4);                 '];
buttoncall(42,1:41) = ['geth1a2(A,fname,fs,h,5);                 '];
buttoncall(19,1:41) = ['[A,acc,aco,fname,fs,t,wcn]=newdata2(h,1);'];
buttoncall(25,1:41) = ['[A,acc,aco,fname,fs,t,wcn]=newdata2(h,2);'];
buttoncall(31,1:41) = ['[A,acc,aco,fname,fs,t,wcn]=newdata2(h,3);'];
buttoncall(37,1:41) = ['[A,acc,aco,fname,fs,t,wcn]=newdata2(h,4);'];
buttoncall(43,1:41) = ['[A,acc,aco,fname,fs,t,wcn]=newdata2(h,5);'];
buttoncall(46,1:41) = ['print(1,"-dmeta","-zbuffer");            '];
```

```
buttoncall(47,1:41) = ['clear;close(1);more on;           '];
for m = 16:47,                                    % Create all of the buttons & text
  h(m) = uicontrol('Style',strcat(buttonstyles(m,:)),...
    'String',strcat(buttonstring(m,:)),...
    'Position',buttonpos(m,:),...
    'BackgroundColor',[1 1 1],...
    'Callback',strcat(buttoncall(m,:)));
end
```

```
function a = autolpc(x,p,w,e)

% FUNCTION AUTOLPC.M
%
% Usage:    a = autolpc(x,p,w,e)
% Input:    x = the signal to be processed (e.g, speech)
%           p = the number of linear prediction coefficients
%           w (OPTIONAL) = window to be used on the signal,
%             if omitted the default is a Hamming window with
%             length equal to the signal length; if the string
%             'none' is passed in argument w, no window is used
%           e (OPTIONAL, but requires passing argument w)
%             pre-emphasis filter coefficients to be used on
%             the signal, with the MATLAB function
%             FILTER(e,1,x); if omitted the default is
%             e = [1 -0.95]; if the string 'none' is passed in
%             argument e, no pre-emphasis is applied to the
%             signal
% Output:   a = the linear prediction coefficients
%
% This MATLAB function performs Linear Prediction Coding (LPC)
% using the autocorrelation method as outlined in Rabiner &
% Schafer (1993) _Fundamentals of Speech Recognition_. It uses
% a Hamming window on the data, as suggested on page 114.
%
% Harold Cheyne
% 19 January 2000
% revised 4 April 2001

N = length(x);
switch nargin
case 2                                  % If only 2 arguments are passed, then
  e = [1 -0.95];                        % set the pre-emphasis and window
  w = hamming(N)';                      % parameters to the default values.
  y = filter(e,1,x);                    % y = pre-emphasized signal
    s = y.*w;                           % s = windowed signal
case 3                                  % If 3 arguments are passed, then the user
  e = [1 -0.95];                        % wants the default pre-emphasis, and
    y = filter(e,1,x);                  % only the argument w needs to be checked.
  if ~ischar(w)                         % If w is not a string, use it to window
    s = y.*w(:)';                       % the signal (ensure it's a row vector),
  elseif w == 'none'                    % otherwise do no windowing on the signal.
    s = y;
  end
```

```
case 4                                          % If 4 arguments are passed, then check
  if ~ischar(e)                                 % to see if e is 'none' or a vector. If
    y = filter(e,1,x);                          % e is not a string then use it as filter
  elseif e == 'none'                            % coefficients for pre-emphasis, otherwise
    y = x;                                      % do no pre-emphasis.
  end                                           % Next check w...
  if ~ischar(w)                                 % If w is not a string, use it to window
    s = y.*w(:)';                               % the signal (ensure it's a row vector),
  elseif w == 'none'                            % otherwise do no windowing on the signal.
    s = y;
  end
end
% Step #4 in Rabiner & Schafer, p. 114.
r = zeros(p+1,1);                               % Initialize autocorrelation matrix.
for m = 0:1:p,                                  % For each coefficient from 0 to p,
  for n = 0:1:N-1-m,                            % and for each lag value,
    r(m+1) = r(m+1) + s(n+1).*s(n+1+m);         % calculate the autocorrelation.
  end
end
% Matrix form of autocorrelation method, from Rabiner & Schafer, p.106.
Rn = zeros(p);                                  % Initialize the left-hand matrix.
for k = 1:p,                                    % Loop through each row, and
  for i = 1:p,                                  % loop through each column, placing
    Rn(k,i) = r(abs(k-i)+1);                    % the autocorrelation as computed
  end                                           % above into the correct cell of the
end                                             % matrix.
a = Rn\r(2:p+1);                                % Solve for the coefficients.
%a = [1;-a];                                    % Make a(0) equal to 1, and place the rest
% of the coefficients in a vector ready for digital filtering.
```

```
function y=bpfacc(acc,fs,h,t,wcn)

% FUNCTION BPFACC.M
%
% Usage:  y=bpfacc(acc,fs,h,t,wcn)
% Input:  acc, the acceleration signal to be filtered
%         fs, the sampling frequency of acc (Hz)
%         h, the GUI handles from ACCFLO4.M
%         t, the vector of time values for acc (sec)
%         wcn, the center index of the windowed portion of acc (samples)
% Output: y, the band-pass filtered acceleration signal
%         The output y is also plotted in the GUI from ACCFLO4.M.
%
% This MATLAB function is called by pressing the "BPF acc(t)" button
% in the GUI from ACCFLO4.M.  It takes the acceleration signal and
% band-pass filters it with the cutoff frequencies appearing in the
% GUI.  Then it outputs the filtered version of the acceleration, and
% plots it in the GUI.
%
% Harold Cheyne
% 27 June 2001
% revised 13 July 2001
```

```
fc1 = str2double(get(h(9),'String'));        % Get the lower cutoff frequency from GUI
fc2 = str2double(get(h(11),'String'));          % Get the upper cutoff frequency from GUI
Wn = [fc1/(fs/2) fc2/(fs/2)];                     % Define the normalized cutoff frequencies
b = fir1(512,Wn);                                       % Create a 512-point band-pass filter
wn = str2double(get(h(21),'String'));          % Get the acceleration window size
n1 = wcn - fix(3.*wn/2);                          % Define the start index for filtering
n2 = n1 + 2.*wn - 1;                                   % Define the end index for filtering
% Use FILTFILT to filter 2 window lengths of the acceleration signal - one length
% prior to selected window + selected window - forwards & backwards to produce
% zero phase shift.
x = filtfilt(b,1,acc(n1:n2));                  % Zero-phase shift filtered acceleration
% Time align the filtered acceleration.
y = -x(wn+1:2.*wn);
n3 = wcn - fix(wn/2);                             % Define the start time index
n4 = n3 + wn - 1;                                       % Define the end time index
subplot(h(5));
hold off
plot(t(n3:n4),y,'k-')                                         % Plot the negative of the filtered
grid                                                          % acceleration because the sign of the
hold on                                                       % flow and the acceleration are inverses.
axis([t(n3) t(n4) min(y).*1.1 max(y).*1.1])
title('Band-pass filtered inverted acceleration')
xlabel('Time (sec)')
ylabel('BP filtered -acc(t), cm/sec^2')
```

```
function hh = derivf(n)
% Make derivative filter (ref. Rabiner and Gold, pp 119-122)
% usage: hh = derivf(n)
%  hh --- impulse response of the filter
%  n  --- order of filter (even number)
% Yingyong Qi, 1996

if rem(n,2) ~= 0
  error('Order of filter should be an even number');
end;

% design a n-pt differentiator
mag = [0:n/2 (n/2-1):-1:1]/(n/2);
ph = [n/2-1:-1:-n/2]*pi/n;
j = sqrt(-1);

% Get impulse resp. of ideal differentiator and 1/2 sample delay
h = ifft(mag.*cos(ph)+j.*mag.*sin(ph),n);

% Shift n/2 pt for a causal system
hh = real([h h]);
hh = hh(n/2+(1:n));
%END of deriv.m
```

```
function [dom,gd]=domain(analysischoice,dom,gd,gdtext,h,sm);

% This function is used exclusively by the MATLAB function LPLV.M,
```

```
% in response to the user activating the "Click to analyze..." button.
%
% Harold Cheyne
% 1 February 2001

dom = ~dom;                                          % Alternate between Time/Freq domain
set(h(28),'String',analysischoice(dom+1,:)); % Change the text in the domain button
if dom                                           % If domain is frequency,
  gd = sm;                                       % set the get data button text to
else                                             % the selected subglottal pole or zero,
  gd = 5;                                      % otherwise set it to "Get Twc"
end                                              % for centering the time window.
set(h(29),'String',gdtext(gd,:));            % Set the text in the get data button.
```

```
function dVvg = dVdtplot3(flow,dvvga,fname,fs,h,in,invflow,t,wcn)

% Function DVDTPLOT3.M
%
% Usage:  dVvg = dVdtplot(flow,dvvga,fname,fs,h,in,invflow,t,wcn)
% Input:  flow, the flow time signal
%         dvvga, the acceleration-derived glottal flow derivative
%         fname, the filename of the data being processed
%         fs, the sampling frequency of the signal invflow
%         h, the handles of the graphics objects created by ACCFLO.M
%         in, a boolean describing whether the time plots are
%           zoomed in (in = 1) or not (in = 0)
%         invflow, the inverse-filtered airflow waveform of which to
%           take the first derivative (using the first difference as
%           an approximate first derivative) and to plot for the
%           funtion ACCFLO.M
%         t, the vector holing the time values for the time signals
%         wcn, the center index of the windowed portion of the flow &
%           acceleration signal being processed by ACCFLO.M
% Output: dVvg, the approximate first derivative of the signal invflow
%
% Harold Cheyne
% 3 April 2001
% revised 15 May 2001
% revised 28 August 2001 to produce plots for the thesis document, i.e., black & white

% Yingyong Qi (1996) used a differentiator filter implemented by
% function DERIVF.M
dtfilt = derivf(512);                        % Get the impulse response of the filter
d = filter(dtfilt,1,[invflow zeros(1,256)]);
n = length(invflow);
dVvg = fs.*pi.*d(256+(1:n))./1e3;           % Get dVvg/dt in litres/sec^2
dVvg(1:64)=zeros(1,64);
dVvg(n-63:n)=zeros(1,64);
dn = str2double(get(h(59),'String'));      % Get Vvg display length from GUI.
vt1 = wcn - fix(dn/2);
vt2 = vt1 + dn - 1;
ddl = str2double(get(h(62),'String'));      % Get dVvg display length from the GUI
dt1 = wcn - fix(ddl/2);
```

```
dt2 = dt1 + ddl - 1;
dvt1 = fix(length(dVvg)/2) - fix(ddl/2);
dvt2 = dvt1 + ddl - 1;
subplot(h(6))
hold off
plot(t(dt1:dt2),dVvg(dvt1:dvt2),'k')
%plot(t(dt1:dt2),dVvg(dvt1:dvt2),'r')
hold on
mindV = min(dVvg(dvt1:dvt2));
maxdV = max(dVvg(dvt1:dvt2));
line([t(dt1) t(dt1)],[mindV maxdV],'Color',[0 0 0],'LineWidth',2);
line([t(dt2) t(dt2)],[mindV maxdV],'Color',[0 0 0],'LineWidth',2);
grid
axis([t(dt1-5) t(dt2+5) mindV*1.1 maxdV*1.1])
xlabel('Time(sec)')
ylabel('dVvg/dt, liter/sec^2');
subplot(h(4));
hold off
if in                                      % If the flow plot is already zoomed in, then
  wnf = str2double(get(h(50),'String')); % get the flow window length from the
  wf = hamming(wnf)';                    % GUI and make a Hamming window for the LP.
  t3 = wcn - fix(wnf/2);               % Calculate the correct start & end points for the
  t4 = t3 + wnf - 1;                   % plot, and show the flow signal in blue,
  plot(t(t3:t4),flow(t3:t4),'k');   % the Hamming window in red, the line on the
%   plot(t(t3:t4),flow(t3:t4),'b');% the Hamming window in red, the line on the
  hold on                                  % center of the window in red, and the time limits
  minflow = min(flow(t3:t4));  % of the inverse-filtered flow display in green.
  maxflow = max(flow(t3:t4));
%   plot(t(t3:t4),wf.*maxflow,'r');
%    line([wcn/fs wcn/fs],[minflow maxflow],'Color',[1 0 0],'LineWidth',2);
%    line([t(vt1) t(vt1)],[minflow maxflow],'Color',[0 1 0],'LineWidth',2);
%    line([t(vt2) t(vt2)],[minflow maxflow],'Color',[0 1 0],'LineWidth',2);
  plot(t(t3:t4),wf.*maxflow,'k');
    line([wcn/fs wcn/fs],[minflow maxflow],'Color',[0 0 0],'LineWidth',2);
    line([t(vt1) t(vt1)],[minflow maxflow],'Color',[0 0 0],'LineWidth',2);
    line([t(vt2) t(vt2)],[minflow maxflow],'Color',[0 0 0],'LineWidth',2);
    line([t(dt1) t(dt1)],[minflow maxflow],'Color',[0 0 0],'LineWidth',2);
    line([t(dt2) t(dt2)],[minflow maxflow],'Color',[0 0 0],'LineWidth',2);
    axis([t(t3) t(t4) minflow maxflow])
    grid
  xlabel('Time (sec)');
  ylabel('cm^3/sec');
else                                       % Otherwise if the entire flow signal is showing,
  plot(t,flow,'k-');                 % then replot the entire flow signal, along with
%   plot(t,flow,'c-');                  % then replot the entire flow signal, along with
    hold on                                % the line on the center of the Hamming window in
  minflow = min(flow);                  % red and the time limits of the inverse-filtered
  maxflow = max(flow);                  % flow display in green.
%    line([wcn/fs wcn/fs],[minflow maxflow],'Color',[1 0 0],'LineWidth',2);
%    line([t(vt1) t(vt1)],[minflow maxflow],'Color',[0 1 0],'LineWidth',2);
%    line([t(vt2) t(vt2)],[minflow maxflow],'Color',[0 1 0],'LineWidth',2);
    line([wcn/fs wcn/fs],[minflow maxflow],'Color',[0 0 0],'LineWidth',2);
    line([t(vt1) t(vt1)],[minflow maxflow],'Color',[0 0 0],'LineWidth',2);
    line([t(vt2) t(vt2)],[minflow maxflow],'Color',[0 0 0],'LineWidth',2);
  line([t(dt1) t(dt1)],[minflow maxflow],'Color',[0 0 0],'LineWidth',2);
```

```
    line([t(dt2) t(dt2)],[minflow maxflow],'Color',[0 0 0],'LineWidth',2);
    grid
  xlabel('Time (sec)');
  ylabel('cm^3/sec');
  axis([0 t(length(t)) minflow maxflow])
end
dVspec3(dvvga,dVvg,fs,h);
```

```
function dVspec3(dvvga,dvvgf,fs,h)

% Function DVSPEC3.M - Calculate and display the spectrum of dVg/dt
%
% Usage:  DVVG=dVspec3(dvvga,dvvgf,fs,h)
% Input:  dvvga, the glottal flow derivative time signal from the acceleration
%           (Set to zero if only dvvgf is being plotted.)
%         dvvgf, the glottal flow derivative time signal from the airflow
%           (Set to zero if only dvvga is being plotted.)
%         fs, the sampling frequency
%         h, the vector of graphics handles for the GUI created by ACCFLOS.M
% Output: The spectral magnitude of the glottal flow derivative time signal(s)
%           is plotted in the middle left subplot of the GUI.
%
% Harold Cheyne
% 8 May 2001
% revised 16 May 2001
% revised 28 August 2001 to produce plots for the thesis document, i.e., black & white

fftlength = str2double(get(h(57),'String'));      % Get the FFT length from the GUI.
f = 0:fs/fftlength:fs/2-fs/fftlength;
f1500 = fix(1500.*fftlength./fs);                   % The index of the frequency <1500Hz.
subplot(h(8));
hold off
if ~isempty(dvvga)                                   % As long as dvvga is not empty,
  n = length(dvvga);                               % Get the length of the input so that
  w = zeros(n,1);                                  % the appropriate Hanning window can
  w(2:n-1) = hanning(n-2);                         % be constructed.
  DVVGA = 2.*fft(dvvga.*w',fftlength)./fftlength;  % Get the spectrum of dvvga.
  DVVGAdB = 20.*log10(abs(DVVGA(1:fftlength./2)));% Calculate the spectrum in dB.
  maxD = max(DVVGAdB(1:f1500));
  [fout,yout] = linespec(f,1500,fs,h,fftlength,DVVGAdB);
  %plot(f(1:f3000),DVVGAdB(1:f3000),'b');
  stem(f(fout),yout,'ko-');
%   stem(f(fout),yout,'bo-');
  hold
end
if ~isempty(dvvgf)
  n = length(dvvgf);                               % Get the length of the input so that
  w = zeros(n,1);                                  % the appropriate Hanning window can
  w(2:n-1) = hanning(n-2);                         % be constructed.
  DVVGF = 2.*fft(dvvgf.*w',fftlength)./fftlength;  % Get the spectrum of dvvgf.
  DVVGFdB = 20.*log10(abs(DVVGF(1:fftlength./2)));% Calculate the spectrum in dB.
  maxD = max(DVVGFdB(1:f1500));
  %plot(f(1:f3000),DVVGFdB(1:f3000),'r');
```

```
  [fout,yout] = linespec(f,1500,fs,h,fftlength,DVVGFdB);
  stem(f(fout),yout,'ko')
%  stem(f(fout),yout,'ro')
end
grid
%if ~isempty(dvvga) & ~isempty(dvvgf)
%  maxD = max([DVVGAdB(1:f1500) DVVGFdB(1:f1500)]);
%end
xlim([0 1500])
xlabel('Frequency (Hz)')
ylabel('|dV_{vg}/dt| in dB re 1 litre/sec^2')
```

```
function [afn,S,wcn]=gdp(afn,dom,f,fmax,fname,fs,h,n,S,sm,t,w,wn,wcn,x,y)

% This function is used exclusively by the MATLAB function LPLV.M,
% in response to the user activating the "Get..." button.
%
% Harold Cheyne
% 3 January 2001
% revised 2 February 2001

[x1,y1]=ginput(1);                              % Get data point from figure.
if ~dom                                             % If current analysis domain is time,
  wcn = round(fs*x1);                          % Get index nearest to the clicked point.
  S = movespec(f,fmax,fname,fs,h,n,0,t,w,wcn,x,y);  % Then replot the acceleration plots.
else                                                % If the analysis domain is frequency,
  Sn1 = round(x1*n/fs);                        % find the index of the chosen frequency
  if sm == 1 | sm == 3                         % and if the current parameter is F1' or F2'
    [a1,f1]=max(20.*log10(S(Sn1-3:Sn1+3)));   % search for the nearest peak,
  else                                                   % or if it is Z1' or Z2',
    [a1,f1]=min(20.*log10(S(Sn1-3:Sn1+3)));   % search for the nearest valley.
  end
  afn(sm,1) = a1;                               % Save the amplitude
  afn(sm,2) = f(Sn1-4+f1);                   % and frequency of the peak/valley.
  subplot(h(4));                                     % Then replot the spectrum.
  an = find(afn(:,2));                          % Get the indeces of non-zero elements.
  plot(f(2:fmax),20.*log10(S(2:fmax)),'k',afn(an,2),afn(an,1),'ko');
  ylim([min(20.*log10(S(2:fmax)))-5 max(20.*log10(S(2:fmax)))+5]);
  grid
  ylabel('dB');
  xlabel('Frequency (Hz)');
  title(strcat('|fft(V_{acc}-mean(V_{acc}))|*2*pi*f, subject:',fname(1:2)));
  set(h(sm+11),'String',num2str(afn(sm,2),'%4.1f'));
  set(h(sm+15),'String',num2str(afn(sm,1),'%3.1f'));
end
```

```
function [afn,S,wcn]=gdplong(afn,dom,f,fmax,fname,fs,h,n,ns,S,sm,t,w,wn,wcn,x,y)

% This function is used exclusively by the MATLAB function LPLVLONG.M,
% in response to the user activating the "Get..." button.
%
% Harold Cheyne
```

```
% 3 January 2001
% revised 30 March 2001

[x1,y1]=ginput(1);                              % Get data point from figure.
if ~dom                                          % If current analysis domain is time,
  wcn = round(fs*x1);                           % Get index nearest to the clicked point.
  S = movespec(f,fmax,fname,fs,h,n,ns,t,w,wcn,x,y); % Replot the acceleration plots.
else                                             % If the analysis domain is frequency,
  Ss = smooth([S S(n/2:-1:1)],ns);              % smooth the spectrum,
  Sn1 = round(x1*n/fs);                         % find the index of the chosen frequency
    if sm == 1 | sm == 3                         % and if the current parameter is F1' or F2'
    [a1,f1]=max(20.*log10(Ss(Sn1-fix(ns/2)-3:Sn1-fix(ns/2)+3))); % search for the
  else                                           % nearest peak, or if it is Z1' or Z2',
    [a1,f1]=min(20.*log10(Ss(Sn1-fix(ns/2)-3:Sn1-fix(ns/2)+3))); % search for the
  end                                            % nearest valley.
  afn(sm,1) = a1+20;                            % Save the amplitude + 20 dB (as plotted)
  afn(sm,2) = f(Sn1-4+f1);                      % and frequency of the peak/valley.
  subplot(h(4));                                % Then replot the spectrum.
  hold off
  an = find(afn(:,2));                          % Get the indeces of non-zero elements.
  plot(f(2:fmax),20.*log10(S(2:fmax)),'k')
  hold on
  plot(f(round(ns/2):fmax-fix(ns/2)),20.*log10(Ss(1:fmax-(ns-1)))+20,'k');
    plot(afn(an,2),afn(an,1),'ko');
    ylim([min(20.*log10(S(2:fmax)))-5 max(20.*log10(Ss(1:fmax-(ns+1))))+25]);
  grid
  ylabel('dB');
  xlabel('Frequency (Hz)');
  title(strcat('|fft(V_{acc}-mean(V_{acc}))|*2*pi*f, subject:',fname(1:2)));
  set(h(sm+11),'String',num2str(afn(sm,2),'%4.1f'));
  set(h(sm+15),'String',num2str(afn(sm,1),'%3.1f'));
end
```

```
function getaccspl(fs,h,p)

% Function GETACCSPL.M - Get the ACCeleration-derived Sound Pressure Level
%
% Usage:  getaccspl(fs,h,p)
% Input:
% Output:
%
% This MATLAB function, called by m-file ACCSPL1.M, calculates the estimated
% SPL from the acceleration-derived microphone pressure spectral magnitude.
%
% Harold Cheyne
% 30 May 2001
% revised 31 May 2001 - parts adapted from LINESPEC.M

h1 = msgbox('Use the cursor to select F0 from the lower right spectrum.');
uiwait(h1);
[f0,a0]=ginput(1);                              % Get frequency and magnitude of cursor point (F0)
n = str2double(get(h(20),'String')); % Get the FFT length from the GUI
f = 0:fs/n:fs/2-fs/n;                          % f = vector of frequencies in the FFT
```

```
f0i = round(f0.*n./fs);                     % Calculate index of nearest freq. to cursor
[a1,f0] = max(p(f0i-3:f0i+3));             % Find peak in pressure spectrum around cursor
f0 = f0 + f0i - 4;                          % Set index of f0 to the peak
numh = fix(3000./f(f0));                    % numh = # of harmonics in line spectrum
peaks = zeros(numh-1,2);                    % Initialize vector to store harmonic peak values
peaks(1,1) = a1;                                % Set first peak value to a1
peaks(1,2) = f0;
for m = 2:numh-1,                                   % For each harmonic,
  fm = round(m.*f0);                        % find the frequency index of the harmonic,
  [peaks(m,1),peaks(m,2)] = max(p(fm-2:fm+2));      % find the maximum around that index,
  peaks(m,2) = peaks(m,2) + fm - 3;     % adjust the index value accordingly, and
  f0 = peaks(m,2)./m;                       % and adjust the index of F0 for accuracy.
end
SPL = 20.*log10(sum(peaks(:,1))./2e-4);% Sum the peak values and get the sum in dB.
set(h(62),'String',num2str(SPL,'%3.1f'));     % Display the SPL value.
subplot(h(7))                                   % And draw red circles on each peak value as
plot(f,20.*log10(p),'k-',f(peaks(:,2)),20.*log10(peaks(:,1)),'ko')
y = ylim(h(2));
axis([0 3000 y(1) y(2)])
grid
xlabel('Frequency (Hz)');
ylabel('dB re 1 dyne/cm^2');
title('Acceleration-derived microphone pressure')
```

```
function getbf1(y,fs,h,t,wcn)

% FUNCTION GETFB1.M
%
% Usage:  getbf1(y,fs,h,t,wcn)
% Input:  y, the band-pass filtered acceleration signal
%         fs, the sampling frequency of y (Hz)
%         h, the graphics handles for the ACCFLO4.M GUI
%         t, the vector holding the time values for y (sec)
%         wcn, the center index of the windowed portion of
%           the acceleration (samples)
% Output: With some graphical (mouse click) inputs from the
%         user, this function estimates the bandwidth of the
%         first subglottal formant and prints that estimate
%         in the GUI.
%
% Harold Cheyne
% 27 June 2001
% revised 12 July 2001

wn = str2double(get(h(21),'String'));       % Get the acceleration window size
n1 = wcn - fix(wn/2);                           % Define the starting point for y
n2 = n1 + wn - 1;                                   % Define the ending point for y
for p = 1:2,                                        % Loop through this operation twice, once
  switch p                                              % for the maximum in each period, and once
  case 1, h1 = msgbox('Click on the first 2 maxima in the lower left plot.');
  case 2, h1 = msgbox('Click on the first 2 secondary oscillation maxima.');
  end                                                   % for the 2nd oscillation maximum in each.
  uiwait(h1);
```

```
  [tmax,amax]=ginput(2);                    % Get two maxima from cursor clicks.
  tmax = sort(tmax);                        % Make sure tmax entries are ascending.
  n = round(tmax(1)*fs)-n1;                 % Convert tmax(1) to an index for y.
  [amax,nmax] = max(y(n-5:n+5));          % Find maxima around first chosen point.
  tmax(1) = t(nmax + n - 6 + n1);         % Adjust tmax(1) to time of 1st maxima.
  n = round(tmax(2)*fs)-n1;                 % Convert tmax(2) to an index for y.
  [amax(2),nmax(2)] = max(y(n-5:n+5));    % Find maxima around second chosen point.
  tmax(2) = t(nmax(2) + n - 6 + n1);      % Adjust tmax(2) to time of 2nd maxima.
  T0 = tmax(2) - tmax(1);                   % Estimate the fundamental period T0.
  for m = 3:8,                                % Find the next five maxima based on T0.
    tmax(m) = tmax(m-1) + T0;               % Estimate the time of the next maximum.
    n = round(tmax(m)*fs)-n1;             % Convert that time to an index in samples.
    [amax(m),nmax(m)] = max(y(n-5:n+5));% Find the next maximum.
    tmax(m) = t(nmax(m) + n - 6 + n1); % Adjust the time estimate based on nmax.
    T0 = (tmax(m) - tmax(1))./(m-1);      % Update the estimate of T0.
  end
  subplot(h(5));
  hold off
  switch p
  case 1, t1 = tmax;, a1 = amax;, plot(t(n1:n2),y,'k-',t1,a1,'ko');
  case 2, t2 = tmax;, a2 = amax;, plot(t(n1:n2),y,'k-',t1,a1,'ko',t2,a2,'ko');
  end
  grid
  axis([t(n1) t(n2) min(y).*1.1 max(y).*1.1])
  title('Band-pass filtered inverted acceleration')
  xlabel('Time (sec)')
  ylabel('BP filtered -acc(t), cm/sec^2')
end
bw = log(mean(a1)./mean(a2))./(pi.*mean(t2-t1));
set(h(15),'String',strcat('B1''=',num2str(bw,'%3.0f'),' Hz'));
```

```
function getF0(h)

% Function GETF0.M
%
% Usage:  getF0(h)
% Input:  h, the graphics handles for the accflo3.m GUI
% Output: the F0 value (Hz) corresponding to the peak selected
%         by the cursor in either the acceleration or airflow
%         raw signal spectrum
%
% Harold Cheyne
% 16 May 2001

again = 1;
m1lim = get(h(2),'Ylim');                  % Get the magnitude and frequency limits
f1lim = get(h(2),'Xlim');                  % of the airflow spectrum.
while again
  [fn,an] = ginput(1);                     % Get cursor input for one point.
  if fn < f1lim(1) | fn > f1lim(2) | an < m1lim(1) | an > m1lim(2)
    errordlg('Choose F0 from the airflow spectrum only.');
  else
    set(h(68),'String',num2str(fn,'%3.0f'));
```

```
    again = 0;
  end
end
```

```
function getfb2(flo,fs,h,wcn);

% Function GETFB2.M - GET vocal tract Formants and Bandwidths
%
% Usage:  getfb2(flo,fs,h,wcn)
% Input:  flo, the flow signal to be used for extraction of
%            vocal tract formant frequencies & bandwidths
%         fs, the sampling frequency (Hz)
%         h, the handles of the GUI objects created by ACCFLO4.M
%         wcn, the window center index
% Output: The vocal tract formant frequencies and bandwidths (Hz)
%            for F1, F2 and F3 are written to their appropriate
%            locations in the GUI.
%
% Harold Cheyne
% 27 June 2001
% revised

wn = str2double(get(h(23),'String'));          % Get analysis window length from GUI.
n1 = wcn - fix(wn/2);                              % n1 = start index of flow window
n2 = n1 + wn - 1;                                     % n2 = end index of flow window
a = autolpc(flo(n1:n2),20,hamming(wn),'none'); % Perform LPC:Hamming, no pre-emph.
a = [1;-a];                                              % a(0) = 1, and negate rest for filtering
r = roots(a);
[flpc,I] = sort(angle(r).*fs./(2.*pi));     % Get the pole frequencies in Hz.
posflpc = find(flpc>0);                               % Then find the positive ones.
fbw(1:3,1) = flpc(posflpc(1:3));                   % Assign F1,F2,F3 frequencies and
r2 = abs(r(I(posflpc(1:3))));                       % get the vector lengths for those roots
fbw(1:3,2) = fs.*acos(-r2./2+2-1./(2.*r2))./pi;    % to compute the bandwidths.

% Now display these findings in the GUI display created by ACCFLO3.M
for m = 1:6,                                              % Display formant freq.'s & bandwidths
  a = fix((m+1)/2);                                      % a = row number of fbw
  b = 2-rem(m,2);                                      % b = column number of fbw
  set(h(31+m),'String',num2str(fbw(a,b),'%4.0f'));
end
```

```
function fbw = getfbl3(flow,fs,h,wcn);

% Function GETFBL3.M - GET vocal tract Formants, Bandwidths, and Length
%
% Usage:  getfbl3(flow,fs,h)
% Input:  flow, the flow signal to be used for extraction of
%            vocal tract formant frequencies & bandwidths, and
%            estimation of vocal tract length
%         fs, the sampling frequency (Hz)
%         h, the handles of the GUI objects created by ACCFLO.M
%         wcn, the window center index
```

```
% Output: fbw, the vocal tract formant frequencies and bandwidths (Hz)
%
% Harold Cheyne
% 2 April 2001
% revised 14 May 2001

wn = str2double(get(h(50),'String'));           % Get analysis window length from GUI.
n1 = wcn - fix(wn/2);                             % n1 = start index of flow window
n2 = n1 + wn - 1;                                   % n2 = end index of flow window
a = autolpc(flow(n1:n2),20,hamming(wn),'none');  % Perform LPC:Hamming, no pre-emph.
a = [1;-a];                                           % a(0) = 1, and negate rest for filtering
r = roots(a);
[flpc,I] = sort(angle(r).*fs./(2.*pi));     % Get the pole frequencies in Hz.
posflpc = find(flpc>0);                             % Then find the positive ones.
fbw(1:3,1) = flpc(posflpc(1:3));                 % Assign F1,F2,F3 frequencies and
r2 = abs(r(I(posflpc(1:3))));                     % get the vector lengths for those roots
fbw(1:3,2) = fs.*acos(-r2./2+2-1./(2.*r2))./pi;  % to compute the bandwidths.
vtl = 5.*35400./(4.*fbw(3,1));                   % Estimate vocal tract length
nf = ceil((fs/2)./(2*fbw(1,1)));                 % Determine the number of upper formants
for m = 4:nf,                                         % Estimate the upper formants
  fbw(m,1) = (2.*m-1).*35400./(4.*vtl);     % Assume upper formants depend only on
  fbw(m,2) = 200;                                   % vocal tract length & upper bandwidths
end                                                      % are fixed.

% Now display these findings in the GUI display created by ACCFLO3.M
for m = 1:6,                                          % Display formant freq.'s & bandwidths
  a = fix((m+1)/2);                                 % a = row number of fbw
  b = 2-rem(m,2);                                   % b = column number of fbw
  set(h(22+m),'String',num2str(fbw(a,b),'%4.0f'));
end
set(h(29),'String',num2str(vtl,'%2.1f'));     % Display vocal tract length.
```

```
function [dom,gd]=getfz(dom,gd,gdtext,h,sm);

% This function is used exclusively by the MATLAB function LPLV.M,
% in response to the user choosing one of the subglottal formant
% or zero "radio buttons", F1', Z1', F2', or Z2'.
%
% Harold Cheyne
% 1 February 2001

switch sm
case 1
  set(h(7:9),'Value',0);
case 2
  set(h([6 8 9]),'Value',0);
case 3
  set(h([6 7 9]),'Value',0);
case 4
  set(h(6:8),'Value',0);
end
if dom                                                  % If domain is frequency,
  gd = sm;                                              % set the get data button text to
```

```
end                                                     % for centering the time window.
set(h(29),'String',gdtext(gd,:));              % Set the text in the get data button.
```

```
function geth1a2(A,fname,fs,h,plotnum)

% Function GETH1A2.M - Get the acceleration spectrum's H1 and A2 amplitudes
%
% Usage:  geth1a2(A,fname,fs,h,plotnum)
% Input:  A, the spectrum of the acceleration
%        fname, the filename
%        fs, the sampling frequency (Hz)
%        h, the graphics handles from the ACCSPL3.M GUI
%        plotnum, a number corresponding to which set of plots to operate on (1-5)
% Output: Function GETH1A2.M circles the first harmonic (H1') and the harmonic peak
%          nearest to the second subglottal formant (A2') in the acceleration
%          spectrum, then prints their difference (H1'-A2') in dB in the GUI
%
% Harold Cheyne
% adapted from getaccspl.m - 30 May 2001
% revised 22 June 2001

hdl1 = msgbox('Use the cursor to select H1 from the spectrum.');
uiwait(hdl1);
[f0,h1]=ginput(1);                          % Get frequency and magnitude of cursor point (F0)
n = 2048;
f = 0:fs/n:fs/2-fs/n;                       % f = vector of frequencies in the FFT
f0i = round(f0.*n./fs);                     % Calculate index of nearest freq. to cursor
[h1,f0] = max(abs(A(f0i-3:f0i+3)));         % Find peak in pressure spectrum around cursor
f0 = f(f0 + f0i - 4);                       % Get frequency of h1 peak.
h1 = 20.*log10(h1);                         % Convert h1 to dB

hdl2 = msgbox('Use the cursor to select A2 from the spectrum.');
uiwait(hdl2);
[f2,a2]=ginput(1);                          % Get frequency and magnitude of cursor point (F0)
f2i = round(f2.*n./fs);                     % Calculate index of nearest freq. to cursor
[a2,f2] = max(abs(A(f2i-3:f2i+3)));         % Find peak in pressure spectrum around cursor
f2 = f(f2 + f2i - 4);                       % Get frequency of f2 peak.
a2 = 20.*log10(a2);                         % Convert a2 to dB
h1a2 = h1-a2;

switch plotnum
case 1, dBhdl = h(21);, spechdl = h(1);
case 2, dBhdl = h(27);, spechdl = h(2);
case 3, dBhdl = h(33);, spechdl = h(3);
case 4, dBhdl = h(39);, spechdl = h(4);
case 5, dBhdl = h(45);, spechdl = h(5);
end

set(dBhdl,'String',strcat('H1''-A2''=',num2str(h1a2,'%2.1f')));    % Display H1'-A2'.
subplot(spechdl)
plot(f,20.*log10(abs(A(1:n./2))),'k-',f0,h1,'ko',f2,a2,'ko')
axis([0 2000 min(20.*log10(abs(A(1:205)))) max(20.*log10(abs(A(1:205))))+5])
grid
```

```
if plotnum == 5
  xlabel('Frequency (Hz)');
end
ylabel('dB re 1 dyne/cm^2');
title(fname);
```

```
function Hvt=getHvt(fs,h)

% FUNCTION GETHVT.M - Get the vocal tract transfer function
%
% Usage:  Hvt=getHvt(fs,h)
% Input:  fs, the sampling frequency (Hz)
%         h, the graphic object handles created by ACCFLO.M
% Output: Hvt, the vocal tract transfer function magnitude
%
% adapted from INVFILT3.M
% Harold Cheyne
% modified from INVFILTF.M - 26 March 2001
% revised 14 May 2001 - last INVFILT3.M revision
% revised 30 May 2001

fp = str2double(get(h(49:2:53),'String'));     % Get frequencies of F1, F2, F3 from GUI
b =  str2double(get(h(50:2:54),'String'));% Get bandwidths of F1, F2, F3 from GUI
vtl = str2double(get(h(56),'String'));         % Get the vocal tract length from the GUI.
nf = round((fs/2)/(2*fp(1)));                       % Determine the number of upper formants
for m = 4:nf,                                          % Estimate the upper formants
  fp(m) = (2.*m-1).*35400./(4.*vtl);          % Assume upper formants depend only on
  b(m) = 250;                                             % vocal tract length and that their
end                                                       % bandwidths are 250 Hz.
[zp,zz] = s2z(fp,b,[],[],fs);
[b1,a1] = zp2tf([],zp,1);                               % Get transfer fxn with gain = 1
h1 = freqz(b1,a1,[0 1],fs);                       % Get the frequency response at DC
[b2,a2] = zp2tf([],zp,1./h1(1));                  % Get transfer fxn with DC gain = 1
n = str2double(get(h(20),'String'));
f = 0:fs/n:fs/2-fs/n;
Hvt = freqz(b2,a2,f,fs);
% Now display the results in the middle lower plot.
subplot(h(6));
plot(f,20.*log10(abs(Hvt)),'k-');
grid
xlim([0 3000])
xlabel('Frequency (Hz)');
ylabel('|V_{vm}/V_{vg}|, dB');
title('Model vocal tract transfer function spectrum');
```

```
function getmfdr3(b,dVvg,fs,h,t,wcn)

% FUNCTION GETMFDR3.M
%
% Usage:  getmfdr3(b,dVvg,fs,h,t,wcn)
% Input:  b, the button pressed that invoked this routine
%         (1="Mark MFDR" for accel. derived dVvg,
```

```
%         2="Mark MFDR" for actual dVvg
%       dVvg, the 1st derivative of the glottal volume velocity
%         either from the airflow or the acceleration
%       fs, the sampling frequency of the signal dVvg
%       h, the handles of the graphics objects created by ACCFLO.M
%       t, the time value for each sample (sec)
%       wcn, the center index of the analysis window
% Output: the mean MFDR (Maximum Flow Declination Rate) calculated by
%         averaging six consecutive minima found near the points
%         selected by the cursor is put into the GUI display created
%         by ACCFLO.M
%
% Harold Cheyne
% 4 April 2001
% revised 14 May 2001
% revised 28 August 2001 to produce plots for the thesis document, i.e., black & white

% mfdr = 3-column array; 1st column = mfdr values, 2nd column = mfdr indeces for
% dVvg vector, 3rd column = mfdr indeces for time vector
mfdr = zeros(6,3);
ld = fix(length(dVvg)./2);
ddl = str2double(get(h(62),'String'));    % Get the display length from the GUI
t1 = wcn - fix(ddl/2);                     % Calculate the start and end points for the
t2 = t1 + ddl - 1;                         % time axis of the dVvg/dt plot, and the
dvt1 = fix(length(dVvg)/2) - fix(ddl/2);  % corresponding start and end points
dvt2 = dvt1 + ddl - 1;                     % of the signal dVvg.
mindV = min(dVvg(dvt1:dvt2));                  % Get the minimum and maximum values of the
maxdV = max(dVvg(dvt1:dvt2));                  % dVvg/dt signal for adjusting the plot axes.
for m = 1:6,                                   % Take six MFDR samples and average them...
  again = 1;                                   % again = boolean deciding repeat or not
  while again
    [tn,an] = ginput(1);                   % Get one point from the cursor.
    if b == 1
      alim = get(h(5),'Ylim');
    else alim = get(h(6),'YLim');          % Get the amplitude limits of the dVvg/dt plot.
    end
    if tn>t(t2) | tn<t(t1) | an<alim(1) | an>alim(2) % If the cursor is clicked
      errordlg('Click inside the glottal flow derivative plot.');   % outside the
    else                                         % glottal flow derivative plot, show an error
      again = 0;                           % and do not change the value of again.
      d = wcn - round(tn.*fs);   % Otherwise, don't repeat, get the index of the
      [mfdr(m,1),mfdr(m,2)] = min(dVvg(ld-d-25:ld-d+25));   % point chosen, & find
      mfdr(m,3) = wcn - d + mfdr(m,2) - 24; % the mfdr value and index as the
      mfdr(m,2) = ld - d - 24 + mfdr(m,2);    % minimum within +/- 25 points of tn.
      mfdrtext = num2str(mean(mfdr(find(mfdr(:,1)),1)),'%4.0f');
      if b == 1
        set(h(47),'String',strcat('MFDR = ',mfdrtext,' liter/sec^2'));
      else set(h(64),'String',strcat('MFDR = ',mfdrtext,' liter/sec^2'));
      end
    end                                        % Show the average of the selected values
  end                                              % values in the GUI display and plot the selected
  if b == 1                                    % MFDR points as black circles.
    subplot(h(5))
  else subplot(h(6))
  end
```

```
  hold off
  if b == 1
    plot(t(t1:t2),dVvg(dvt1:dvt2),'k',t(mfdr(1:m,3)),mfdr(1:m,1),'ko')
%     plot(t(t1:t2),dVvg(dvt1:dvt2),'b',t(mfdr(1:m,3)),mfdr(1:m,1),'ko')
else
  plot(t(t1:t2),dVvg(dvt1:dvt2),'k',t(mfdr(1:m,3)),mfdr(1:m,1),'ko')
%  plot(t(t1:t2),dVvg(dvt1:dvt2),'r',t(mfdr(1:m,3)),mfdr(1:m,1),'ko')
  end
  hold on
  line([t(t1) t(t1)],[mindV maxdV],'Color',[0 0 0],'LineWidth',2);
  line([t(t2) t(t2)],[mindV maxdV],'Color',[0 0 0],'LineWidth',2);
  grid
  axis([t(t1-5) t(t2+5) mindV*1.1 maxdV*1.1])
  xlabel('Time(sec)')
  if b == 1
    ylabel('dVvg/dt from a(t), liter/sec^2');
  else ylabel('dVvg/dt, liter/sec^2');
  end
end
```

```
function getminflo(vglp,fs,h,t,wcn)

% FUNCTION GETMINFLO.M
%
% Usage:  getminflo(vglp,fs,h,t,wcn)
% Input:  vglp, the low-pass filtered estimated glottal volume
%           velocity
%         fs, the sampling frequency (Hz)
%         h, the graphics handles from the ACCFLO4.M GUI
%         t, the vector of time values for the original flow
%         wcn, the center index of the windowed portion of the
%           time signals
% Output: The MATLAB function GETMINFLO.M processes the signal
%           vglp to obtain an estimate of the minimum flow
%           averaged over 8 consecutive periods, and places
%           that estimate in the ACCFLO4.M GUI.
%
% This MATLAB function is based on the techniques discussed in
% Perkell JS, Holmberg EB, Hillman RE (1991).  A system for signal
% processing and data extraction from aerodynamic, acoustic, and
% electroglottographic signals in the study of voice production.
% J Acoust Soc Am 89(4) Pt.1, pp. 1777-1781.
% AND on the MATLAB functions HIBAND and PKPSD written by
% Yingyong Qi.
%
% Harold Cheyne
% 27 June 2001
% revised 28 June 2001

wn = str2double(get(h(21),'String'));    % Get the acceleration window size
n1 = wcn - fix(wn/2);                        % Define the starting point for window
n2 = n1 + wn - 1;                                % Define the ending point for window
n3 = n2 - length(vglp) + 1;              % Define the starting point for vglp
```

```
n4 = n2;                                        % Define the ending point for vglp
h1 = msgbox('Click on the 1st two maxima in the lower right plot.');
uiwait(h1);
[tmax,amax]=ginput(2);                          % Get two maxima from cursor clicks.
tmax = sort(tmax);                              % Make sure tmax entries are ascending.
n = round(tmax(1)*fs)-n3;                       % Convert tmax(1) to an index for vglp.
[amax,nmax] = max(vglp(n-5:n+5));               % Find maxima around first chosen point.
nmax = nmax + n - 6;                            % Adjust nmax to index w.r.t. vglp
tmax(1) = t(nmax + n3);                         % Adjust tmax(1) to time w.r.t. original flow
n = round(tmax(2)*fs)-n3;                       % Convert tmax(2) to an index for vglp.
[amax(2),nmax(2)] = max(vglp(n-5:n+5));% Find maxima around second chosen point.
nmax(2) = nmax(2) + n - 6;                      % Adjust nmax(2) to index w.r.t. vglp
tmax(2) = t(nmax(2) + n3);                      % Adjust tmax(2) to time w.r.t. original flow
T0 = tmax(2) - tmax(1);                         % Estimate the fundamental period T0.
for m = 3:9,                                    % Find the next six maxima based on T0.
  tmax(m) = tmax(m-1) + T0;                     % Estimate the time of the next maximum.
  n = round(tmax(m)*fs)-n3;                     % Convert that time to an index in samples.
  [amax(m),nmax(m)] = max(vglp(n-5:n+5));% Find the next maximum.
  nmax(m) = nmax(m) + n - 6;                    % Adjust nmax(m) to index w.r.t. vglp
  tmax(m) = t(nmax(m) + n3);                    % Adjust the time estimate based on nmax.
  T0 = (tmax(m) - tmax(1))./(m-1);              % Update the estimate of T0.
end
% Now the maxima are defined. Between those maxima, the threshold-based
% markers of the closed/open phase boundaries need to be placed. Use
% Yingyong Qi's PKPSD function, modified by me, with a 30% threshold.
[mk,tt] = pkpsd(vglp,nmax,0.3);

for m = 1:8,
  seglen = mk(m.*2)-mk(m.*2-1);                 % Define the closed phase length in samples.
  st = mk(m.*2-1) + fix(seglen./3); % st = start of min. flow portion
  ed = st + fix(seglen./3);                     % ed = end of min. flow portion
  MF(m) = median(vglp(st:ed));                  % MF = min. flow for period m is median of
end                                             % the center 1/3 of the closed phase.
minflo = mean(MF);                              % reported min. flow is average over 8 periods
set(h(39),'String',strcat('Min. flow=',num2str(minflo,'%3.1f')));
% Mark the peaks and thresholds on the glottal flow estimate plot.
subplot(h(6))
hold off
plot(t(n3:n4),vglp,'k-',tmax,amax,'ko',t(mk+n3),vglp(mk),'ko')
minvglp = min(vglp)-max(vglp).*0.1;
maxvglp = max(vglp).*1.1;
hold
grid
axis([t(n3) t(n4) minvglp maxvglp])
title('Inverse-filtered flow')
xlabel('Time (sec)')
ylabel('v_{vg}(t) estimate, cm^3/sec')
```

```
function [vg,vglp] = invfilt3(fbw,flow,flpf,fname,fs,h,in,t,wcn)

% FUNCTION INVFILT3.M
%
% Usage:  [vg,vglp] = invfilt3(fbw,flow,flpf,fs,h,wcn)
```

```
% Input:  fbw, the flow-estimated formant frequencies and
%           bandwidths (from function GETFBL.M)
%         flow, the airflow measured at the mouth (cm^3/sec)
%         flpf, the corner frequency for the flow low-pass
%           filter (4th order Butterworth)
%         fs, the sampling frequency (Hz)
%         h, the graphic object handles created by ACCFLO.M
%         wcn, the window center index
% Output: vg, the inverse-filtered airflow
%         vglp, a low-pass filtered version of vg
%
% FLOFILT.M is modeled after YingYong Qi's hiband.m function, but uses
% different computation techniques.  It takes the airflow signal from the
% Rothenberg mask, applies LPC analysis to estimate F1, F2 and F3, then
% inverse filters the vocal tract transfer function out of the airflow
% signal to produce vg.  Signal vglp is simply signal vg after having
% a 4th-order Butterworth LPF with Fc = flpf Hz applied.
%
% Harold Cheyne
% modified from INVFILTF.M - 26 March 2001
% revised 14 May 2001
% revised 28 August 2001 to produce plots for the thesis document, i.e., black & white
% revised 17 September 2001 to correct for window gain problem (see accflo3)

wn = str2double(get(h(50),'String'));            % Get analysis window length from GUI.
n1 = wcn - fix(wn/2);                              % n1 = start index of flow window
n2 = n1 + wn - 1;                                     % n2 = end index of flow window
w = hamming(wn)'.*1.8525;                             % Gain-corrected Hamming window
fbw(1:3,1) = str2double(get(h(23:2:27),'String')); % Use F1, F2, F3 & their BW's
fbw(1:3,2) = str2double(get(h(24:2:28),'String')); % from the GUI.
zp = vtractz(fbw(:,1),fbw(:,2),fs);              % Get z-domain poles from fp & b
[b1,a1] = zp2tf([],zp,1);                          % Get transfer fxn with gain = 1
H = freqz(b1,a1,[0 1],fs);                         % Get the frequency response at DC
[b2,a2] = zp2tf([],zp,1./H(1));                  % Get transfer fxn with DC gain = 1
vminv = filter(a2,sum(b2),flow(n1:n2));        % them in the inverse filtering procedure.
vg = vminv(length(a2):wn);                            % Time align the output with the input.
% Second, low-pass the flow signal vm with a 1.25kHz Butterworth low-pass filter.
[blpf,alpf] = butter(4,flpf/(fs/2));
vglp = filter(blpf,alpf,vg);
vglp = vglp(length(alpf):length(vg));            % Again, time align the output.
% Now display the results in the middle center plot, and mark the left and right
% display time limits on the center right plot.
subplot(h(7));
hold off
dn = str2double(get(h(59),'String'));          % Get Vvg display length from GUI.
t1 = wcn - fix(dn/2);
t2 = t1 + dn - 1;
vt1 = fix(length(vglp)/2) - fix(dn/2);
vt2 = vt1 + dn - 1;
plot(t(t1:t2),vglp(vt1:vt2),'k');
%plot(t(t1:t2),vglp(vt1:vt2));
minvglp = min(vglp(vt1:vt2))-max(vglp(vt1:vt2)).*0.1;
maxvglp = max(vglp(vt1:vt2)).*1.1;
hold
line([t(t1) t(t1)],[minvglp maxvglp],'Color',[0 0 0],'LineWidth',2);
```

```
line([t(t2) t(t2)],[minvglp maxvglp],'Color',[0 0 0],'LineWidth',2);
%line([t(t1) t(t1)],[minvglp maxvglp],'Color',[0 1 0],'LineWidth',2);
%line([t(t2) t(t2)],[minvglp maxvglp],'Color',[0 1 0],'LineWidth',2);
grid
axis([t(t1-5) t(t2+5) minvglp maxvglp])
ylabel('V_{vg}, cm^3/sec')
xlabel('Time (sec)')
title('Inverse-filtered flow signal')
subplot(h(4));
hold off
if in                                   % If the flow plot is already zoomed in, then
  t3 = wcn - fix(wn/2);            % Calculate the correct start & end points for the
  t4 = t3 + wn - 1;                     % plot, and show the flow signal in blue,
  plot(t(t3:t4),flow(t3:t4),'k');  % the Hamming window in red, the line on the
%  plot(t(t3:t4),flow(t3:t4),'b');% the Hamming window in red, the line on the
  hold on                               % center of the window in red, and the time limits
  minflow = min(flow(t3:t4));  % of the inverse-filtered flow display in green.
  maxflow = max(flow(t3:t4));
  plot(t(t3:t4),w.*maxflow./1.8525,'k');
%  plot(t(t3:t4),w.*maxflow,'r');
     line([wcn/fs wcn/fs],[minflow maxflow],'Color',[0 0 0],'LineWidth',2);
     line([t(t1) t(t1)],[minflow maxflow],'Color',[0 0 0],'LineWidth',2);
     line([t(t2) t(t2)],[minflow maxflow],'Color',[0 0 0],'LineWidth',2);
%    line([wcn/fs wcn/fs],[minflow maxflow],'Color',[1 0 0],'LineWidth',2);
%    line([t(t1) t(t1)],[minflow maxflow],'Color',[0 1 0],'LineWidth',2);
%    line([t(t2) t(t2)],[minflow maxflow],'Color',[0 1 0],'LineWidth',2);
     axis([t(t3) t(t4) minflow maxflow])
     grid
  xlabel('Time (sec)');
  ylabel('V_m, cm^3/sec');
else                                    % Otherwise if the entire flow signal is showing,
  plot(t,flow,'k-');               % then replot the entire flow signal, along with
%  plot(t,flow,'c-');                  % then replot the entire flow signal, along with
     hold on                            % the line on the center of the Hamming window in
  minflow = min(flow);                % red and the time limits of the inverse-filtered
  maxflow = max(flow);                % flow display in green.
     line([wcn/fs wcn/fs],[minflow maxflow],'Color',[0 0 0],'LineWidth',2);
     line([t(t1) t(t1)],[minflow maxflow],'Color',[0 0 0],'LineWidth',2);
     line([t(t2) t(t2)],[minflow maxflow],'Color',[0 0 0],'LineWidth',2);
%    line([wcn/fs wcn/fs],[minflow maxflow],'Color',[1 0 0],'LineWidth',2);
%    line([t(t1) t(t1)],[minflow maxflow],'Color',[0 1 0],'LineWidth',2);
%    line([t(t2) t(t2)],[minflow maxflow],'Color',[0 1 0],'LineWidth',2);
  grid
  xlabel('Time (sec)');
  ylabel('V_m, cm^3/sec');
  axis([0 t(length(t)) minflow maxflow])
end
```

```
function [vg,vglp] = invfilt4(flo,flpf,fs,h,t,wcn)

% FUNCTION INVFILT4.M
%
% Usage:  [vg,vglp] = invfilt4(flo,flpf,fs,h,t,wcn)
% Input:  flo, the airflow measured at the mouth (cm^3/sec)
```

```
%       flpf, the corner frequency for the flow low-pass
%         filter (4th order Butterworth)
%       fs, the sampling frequency (Hz)
%       h, the graphic object handles created by ACCFLO4.M
%       t, the vector of time values for each sample index
%       wcn, the window center index
% Output: vg, the inverse-filtered airflow
%       vglp, a low-pass filtered version of vg
%
% Harold Cheyne
% modified from INVFILT3.M, INVFILTF.M, and GETHVT.M - 26 June 2001
% revised 9 July 2001

fp = str2double(get(h(32:2:36),'String'));% Get frequencies of F1, F2, F3 from GUI
b =  str2double(get(h(33:2:37),'String'));% Get bandwidths of F1, F2, F3 from GUI
vtl = 5.*35400./(4.*fp(3));              % Compute the vocal tract length from F3.
nf = round((fs/2)/(2*fp(1)));              % Determine the number of upper formants
for m = 4:nf,                                 % Estimate the upper formants
  fp(m) = (2.*m-1).*35400./(4.*vtl);     % Assume upper formants depend only on
  b(m) = 250;                                    % vocal tract length and that their
end                                              % bandwidths are 250 Hz.
[zp,zz] = s2z(fp,b,[],[],fs);
[b1,a1] = zp2tf([],zp,1);                   % Get transfer fxn with gain = 1
h1 = freqz(b1,a1,[0 1],fs);              % Get the frequency response at DC
[b2,a2] = zp2tf([],zp,1./h1(1));          % Get transfer fxn with DC gain = 1
n = str2double(get(h(20),'String'));
f = 0:fs/n:fs/2-fs/n;
wn = str2double(get(h(23),'String'));      % Get flow window length from GUI.
n1 = wcn - fix(wn/2);                         % n1 = start index of flow window
n2 = n1 + wn - 1;                                 % n2 = end index of flow window
n1f = n1 - 64;                                       % n1f = start index for filtering
vminv = filter(a2,sum(b2),flo(n1f:n2)); % Inverse filter the flow signal, and
vg = vminv(length(a2):wn+64);                 % time align the output with the input.
% Second, low-pass the flow signal vm with a 1.25kHz Butterworth low-pass filter.
[blpf,alpf] = butter(4,flpf/(fs/2));
vglp = filter(blpf,alpf,vg);
vglp = vglp(length(alpf)+64:length(vg));     % Again, time align the output.
% Now display the results in the lower right plot.
subplot(h(6));
hold off
n3 = n2 - length(vglp) + 1;
n4 = n2;
plot(t(n3:n4),vglp,'k-');
minvglp = min(vglp)-max(vglp).*0.1;
maxvglp = max(vglp).*1.1;
hold
grid
axis([t(n3) t(n4) minvglp maxvglp])
ylabel('v_{vg}(t) estimate, cm^3/sec')
xlabel('Time (sec)')
title('Inverse-filtered flow')
```

```
function dvvga = invfiltat3(accel,dVvg,fs,h,t,wcn)
```

```
% FUNCTION INVFILTAT3.M - Inverse filter the acceleration in the time domain
%
% Usage:  dvvga = invfiltat3(accel,dVvg,fs,h,t,wcn)
% Input:  accel, the measured acceleration at the neck (cm/sec^2)
%         dVvg, the flow-derived glottal flow derivative
%         fs, the sampling frequency (Hz)
%         h, the graphic object handles created by ACCFLO.M
%         t, the time values for each sample of accel (sec)
%         wcn, the window center index
% Output: dvvga, the acceleration-derived glottal volume velocity derivative
%
% Harold Cheyne
% 4 April 2001
% revised 23 May 2001 - added high-pass filter to tracheal wall model
% revised 25 May 2001 - changed HPF in tracheal wall model to 4th-order Butterworth
% revised 14 June 2001 - removed revision of 25 May 2001 due to phase response!
% revised 28 August 2001 to produce plots for the thesis document, i.e., black & white

Spf = str2double(get(h([37 41]),'String'));     % Get F1' and F2' frequencies
Spbw = str2double(get(h([38 42]),'String'));  % Get F1' and F2' bandwidths
Szf = str2double(get(h([39 43]),'String'));     % Get Z1' and Z2' frequencies
Szbw = str2double(get(h([40 44]),'String'));  % Get Z1' and Z2' bandwidths
wn = str2double(get(h(57),'String'));              % Get FFT length from GUI.
f = 0:1e4;

% ============== Tracheal wall model ==================
% The tracheal wall model includes a resistance and a mass.  It ignores the
% compliance of the wall because past reports of tracheal wall input impedance
% and vocal tract wall input impedance show the wall's resonance to be below
% 80 Hz.  It converts the acceleration to pressure in the tracheal deep to the
% accelerometer.
Mm = str2double(get(h(55),'String'));              % Get model mass value from GUI.
Rm = str2double(get(h(53),'String'));              % Get model resistance from GUI.
A = 0.8.*0.56;                                                  % A = area under accelerometer
% The following if/then is from 14 June 2001.
if Rm == 0                                                        % If the resistance is zero, then
  pa = accel(wcn-wn/2:wcn+wn/2-1).*Mm./A;   % convert from acceleration to pressure
else                                                                % without creating this digital filter:
  zp1 = 1;                                                        % z-domain pole at z = 1, and zero at
  zz1 = cos(2.*pi.*Rm./(Mm.*fs)) + j.*sin(2.*pi.*Rm./(Mm.*fs)); % e^(j*2*pi*Rm/(Mm*fs))
  [b1,a1] = zp2tf(zz1,zp1,1);                        % 1st estimate of digital filter
  H1 = freqz(b1,a1,[1 3000],fs);                   % Get filter response at 1, 3000 Hz
  G1 = Mm./(A.*abs(H1(2)));                              % Set gain at 3000 Hz to Mm/A
  [b2,a2] = zp2tf(zz1,zp1,G1);                      % re-calculate filter with gain G1
  pa = filter(b2,a2,accel(wcn-wn/2:wcn+wn/2-1));     % Filter acceleration signal.
end
% The following is the filter from 23 May 2001.
pa2 = filter([1 -1],[1.0031 -0.9969],pa);       % High-pass pa at 20 Hz.
% The following is the HP Butterworth filter - DO NOT USE BECAUSE IT HAS BAD PHASE!
%[b3,a3] = butter(4,75/1e4,'high');                % 4th-order Butterworth HPF with Fc=75Hz
%pa2 = filter(b3,a3,pa);                                    % HPF the estimated pressure signal
% ========== End of tracheal wall model ================

[pz,zz] = s2z(Szf,Szbw,Spf,Spbw,fs);          % For inverse filter, swap poles & zeros
```

```
[b4,a4] = zp2tf(zz,pz,1);                    % Get transfer fxn with gain = 1
% Get the filter frequency response at 0 Hz, and set gain so that 0 Hz is at 50dB.
H4 = freqz(b4,a4,[0 1],fs);
G4 = 10^(50/20)./abs(H4(1));
[b5,a5] = zp2tf(zz,pz,G4);
% Create the "inverse" filter for the subglottal transfer impedance.
dvvga = filter(b5,a5,pa2);
dvvga = -dvvga./1e3;                         % Convert from cm^3/sec^2 to litre/sec^2
dvvga = LPb125(dvvga,fs);                    % Low-pass filter the result @ 1.25kHz.

% Plot results.
l2 = fix(length(dvvga)./2);
ddl = str2double(get(h(62),'String'));      % Get dVvg display length from the GUI
dt1 = wcn - fix(ddl/2);
dt2 = dt1 + ddl - 1;
dvt1 = fix(length(dVvg)/2) - fix(ddl/2);
dvt2 = dvt1 + ddl - 1;
subplot(h(5))
hold off
plot(t(dt1:dt2),dvvga(dvt1:dvt2),'k')
%plot(t(dt1:dt2),dvvga(dvt1:dvt2),'b')
grid
maxdvvga = max(dvvga(dvt1:dvt2)).*1.1;
mindvvga = min(dvvga(dvt1:dvt2)) - max(dvvga(dvt1:dvt2)).*0.1;
axis([t(dt1) t(dt2) mindvvga maxdvvga])
xlabel('Time (sec)')
ylabel('dV_{vg}/dt from a(t), litre/sec^2')
% Now plot the spectrum of the acceleration-derived glottal flow derivative.
dVspec3(dvvga,dVvg,fs,h)
```

```
function [fout,yout] = linespec(f,fmax,fs,h,n,yin)

% Function LINESPEC.M - Create line spectrum from FFT of harmonic signal.
%                 For example, the spectrum for voice.
%
% Usage:  [fout,yout] = linespec(f,fmax,fs,h,n,yin)
% Input:  f, the vector containing the frequencies in the spectrum (Hz)
%         fmax, the maximum frequency for which to determine the line spectrum;
%           for example, 3000 Hz
%         fs, the sampling frequency (Hz)
%         h, the graphics handles from the accflo3.m GUI
%         n, the FFT length used to produce the spectrum
%         yin, a vector of spectral magnitudes for the frequencies in fin
% Output: fout, a vector of the frequency vector indeces for the line spectrum (Hz)
%         yout, a vector containing the spectral magnitudes for the frequencies
%           in fout
%
% Harold Cheyne
% 8 May 2001
% revised 16 May 2001

F0 = str2double(get(h(68),'String'));       % Get F0 from the GUI.
f0i = round(F0.*n./fs);                              % Calculate index of nearest freq. to F0.
```

```
[a1,f0] = max(yin(f0i-3:f0i+3));                % Get peak in spectrum around F0
f0 = f0 + f0i - 4;                              % Adjust index of f0
numh = fix(fmax./f(f0));                        % numh = # of harmonics in line spectrum
yout = zeros(numh-1,1);                             % Initialize output vectors.
fout = zeros(numh-1,1);
yout(1) = a1;
fout(1) = f0;
for m = 2:numh-1,                                       % For each harmonic,
  fm = m.*f0;                                           % find the frequency index of the harmonic,
  [yout(m),fout(m)] = max(yin(fm-2:fm+2));      % find the maximum around that index,
  fout(m) = fout(m) + fm - 3;                   % and adjust the frequency index for it.
end
```

```
function y = LPb125(x,fs);

% Low-pass the input signal x with a 1.25kHz Butterworth low-pass filter.
% revised 8 May 2001
% Harold Cheyne
[blpf,alpf] = butter(4,1250/(fs/2));
x2 = filter(blpf,alpf,x);
y = x2(length(alpf):length(x2));                % time align the output.
```

```
% M-File LPLV.M - Larynx Postition and Lung Volume data analysis
% adapted from m-file SPAN.M
%
% Harold Cheyne
% 21 December 2000
% revised 5 February 2001

[fname,fpath] = uigetfile('*.mat','Enter the filename for the analysis.');
if fname ~= 0
  load(strcat(fpath,fname));
  lacc = length(acc);                           % lacc = length of data (samples)
  wnch = inputdlg('Enter the analysis window length.');
  wn = str2num(char(wnch));                     % wn = length of window for fft
  w = zeros(1,wn);                                  % Initialize the window vector.
  w(2:wn-1) = hanning(wn-2);                    % Use a Hanning window for the spectra
  wcn = round(lacc./2);                         % Initialize wcn = window center index
  n1 = wcn - wn/2 + 1;                              % n1 = window start point
     n2 = n1 + wn - 1;                              % n2 = window end point
  x = acc - mean(acc);                          % operate on the AC acceleration only
  y = egg;                                              % rename egg signal for functions below
  z = aco;                                              % rename acoustic signal for functions below
  afn = zeros(4,2);                                 % Initialize amp. & freq. output vector.
  h = zeros(31,1);                              % Initialize handle vector for all handles.
  % h(1) is entire acceleration signal    h(16) is for text box F1'amp.
  % h(2) is acceleration signal zoom         h(17) is for text box Z1'amp.
  % h(3) is acoustic signal zoom             h(18) is for text box F2'amp.
  % h(4) is acceleration spectrum            h(19) is for text box Z2'amp.
  % h(5) is acoustic spectrum                h(20) is for text box mean EGG voltage
  % h(6) is for button F1'                   h(21) is for button -> (move window right)
  % h(7) is for button Z1'                   h(22) is for button <- (move window left)
```

```
% h(8) is for button F2'                    h(23) is for text box Move Analysis Window
% h(9) is for button Z2'                    h(24) is for text box Vocal tract...
% h(10) is for title "Freq. (Hz)"         h(25) is for text box F1=...
% h(11) is for title "Amp. (dB)"            h(26) is for text box F2=...
% h(12) is for text box F1' freq.       h(27) is for text box F3=...
% h(13) is for text box Z1' freq.     h(28) is for button Click to analyze...
% h(14) is for text box F2' freq.   h(29) is for button Get...
% h(15) is for text box Z2' freq.      h(30) is for button Print
%                             h(31) is for button Done
t = 0:1/fs:(lacc-1)/fs;                        % Create a time index for the signal.
n = 2048;                                          % n = length of fft
S = ones(n,1);                                     % Initialize spectrum vector.
f = 0:fs/n:fs/2-fs/n;                       % Create the frequency vector for the spectra
fmax = round(3000.*n./fs);                        % fmax = highest freq. index to display

% sm is a scalar corresponding to the spectral measure being estimated, with
% 1 = f1', 2 = z1', 3 = f2', 4 = z2'.
sm = 1;

% gd is a scalar corresponding to the state of the push button for getting data,
% and gdtext holds the text for the push button for each state.
% 1 = Get F1', 2 = Get Z1', 3 = Get F2', 4 = Get Z2', 5 = Get Twc (window center)
gd = 5;
gdtext = ['Get F1''';'Get Z1''';'Get F2''';'Get Z2''';'Get Twc'];

% analysischoice holds the text to go in a button that allows the user to
% choose between using the mouse input to locate the time window or get
% data from the acceleration spectrum.
analysischoice = ['Click to analyze frequency domain'];
analysischoice(2,:) = [' Click to analyze time domain   '];

dom = 0;                                                 % 0 = time, 1 = frequency

% Create the figure for the GUI.
figure(1)
clf
set(1,'Renderer','OpenGL','Position',[1 29 1152 768],'Resize','off','Color',[1 1 1])
h(1) = subplot(3,1,1);                           % h(1) = entire acceleration time signal,
plot(t,x,'k-');                                    % used for reference, to see where spectra
   hold on
   line([wcn/fs wcn/fs],[min(x) max(x)],'Color',[0 0 0],'LineWidth',2) % and window
grid                                                     % are located relative to entire signal.
xlabel('Time (sec)');
ylabel('cm/s^2');                                    % The acceleration signal is calibrated.
title(strcat(fname,',acceleration signal'));
axis([0 t(length(t)) min(x) max(x)])

h(2) = subplot(3,2,3);                           % h(2) = close-up of the few periods
t1 = n1-wn*2;
t2 = n2+wn*2;
plot(t(t1:t2),x(t1:t2),'k',t(n1:n2),w.*max(x(t1:t2)),'k');
   axis([t(t1) t(t2) min(x(t1:t2)) max(x(t1:t2))]);
   grid
   xlabel('Time (sec)');
   ylabel('cm/s^2');
```

```
   title(strcat('acceleration signal,','analysis window length=',num2str(wn)));

h(3) = subplot(3,2,4);                           % h(3) = close-up of the few periods
   plot(t(t1:t2),z(t1:t2),'k'); % Re-plot acoustic time signal,
   hold on
   line([wcn/fs wcn/fs],[min(z(t1:t2)) max(z(t1:t2))],'Color',[0 0 0],'LineWidth',2);
   axis([t(n1-wn*2) t(n2+wn*2) min(z(t1:t2)) max(z(t1:t2))])
   grid
   xlabel('Time (sec)');
   ylabel('dyne/cm^2');
   title('acoustic signal');

h(4) = subplot(3,3,7);                           % h(4) = acceleration spectrum
   X = 2.*fft(x(n1:n2).*w,n)./n;                 % Take the FFT of the new windowed portion.
   S = abs(X(1:n/2)).*2.*pi.*f;                  % Calculate the spectrum magnitude,
   plot(f(2:fmax),20.*log10(S(2:fmax)),'k');     % and plot it.
   ylim([min(20.*log10(S(2:fmax)))-5 max(20.*log10(S(2:fmax)))+5]);
   grid
   ylabel('dB');
   xlabel('Frequency (Hz)');
   title(strcat('|fft(V_{acc}-mean(V_{acc}))|*2*pi*f, subject:',fname(1:2)));

h(5) = subplot(3,3,9);                           % h(5) = acoustic LPC spectrum
a = autolpc(z(wcn-n/2+1:wcn+n/2),20);      % Perform LPC on a n-point window around wcn
   az = ones(21,1);                                 % Create a vector for holding coefficients
az(2:21) = -a;                                      % and put the LPC coefficients in it for
LPC = freqz(1,az,f,fs);                          % use with the function FREQZ.M.
plot(f(2:fmax),20.*log10(abs(LPC(2:fmax))),'k');   % Plot the LPC spectrum.
ylim([min(20.*log10(LPC(2:fmax)))-5 max(20.*log10(LPC(2:fmax)))+5]);
grid
ylabel('dB');
xlabel('Frequency (Hz)');
title('LPC spectrum of acoustic signal');
flpc = sort(angle(roots(az)).*fs./(2.*pi));   % Get the pole frequencies in Hz.
posflpc = find(flpc>0);                                  % Then find the positive ones.

% Create the spectral measurement interactive display.

% First create the buttons to choose which spectral measure to make.
buttonpos = [501 141 40 20;...
     501 121 40 20;...
     501 101 40 20;...
     501 81 40 20];
buttontext = ['F1 ''';'Z1 ''';'F2 ''';'Z2 '''];
for m = 1:4,
  h(m+5) = uicontrol('Style','radiobutton',...
     'String',buttontext(m,:),...
     'Position',buttonpos(m,:),...
     'SelectionHighlight','off',...
     'BackgroundColor',[1 1 1]);
end
set(h(6),'Value',1,'Callback','sm=1;,[dom,gd]=getfz(dom,gd,gdtext,h,sm);');
set(h(7),'Callback','sm=2;,[dom,gd]=getfz(dom,gd,gdtext,h,sm);');
set(h(8),'Callback','sm=3;,[dom,gd]=getfz(dom,gd,gdtext,h,sm);');
set(h(9),'Callback','sm=4;,[dom,gd]=getfz(dom,gd,gdtext,h,sm);');
```

```
% Second, create the labels above the frequency and amplitude values for
% the measures.
h(10) = uicontrol('Style','text',...
  'String','Freq. (Hz)',...
  'Position',[541 161 50 20],...
  'BackgroundColor',[1 1 1]);
h(11) = uicontrol('Style','text',...
  'String','Amp. (dB)',...
  'Position',[591 161 50 20],...
  'BackgroundColor',[1 1 1]);

% Third, create the text boxes to hold the frequency and amplitude values measured.
freqpos = [541 141 50 20;...
    541 121 50 20;...
    541 101 50 20;...
    541 81 50 20];
freqtext = [' ';' ';' ';' '];
amppos = [591 141 50 20;...
    591 121 50 20;...
    591 101 50 20;...
    591 81 50 20];
amptext = [' ';' ';' ';' '];
for m = 1:4,
  h(m+11) = uicontrol('Style','text',...
    'String',freqtext(m,:),...
    'Position',freqpos(m,:),...
    'BackgroundColor',[1 1 1]);
  h(m+15) = uicontrol('Style','text',...
    'String',amptext(m,:),...
    'Position',amppos(m,:),...
    'BackgroundColor',[1 1 1]);
end

% Fourth, create the text box to display the mean EGG voltage for the
% time window selected.
h(20) = uicontrol('Style','text',...
  'String',strcat('Mean(EGG)=',num2str(mean(y(n1:n2)),'%2.3f')),...
  'Position',[421 171 110 20],...
  'BackgroundColor',[1 1 1]);

% Fifth, create the buttons for moving the time analysis window one point
% at a time, and the text above them.

h(21) = uicontrol('Style','pushbutton',...
  'String','->',...
  'Position',[591 261 50 20],...
  'BackgroundColor',[1 1 1],...
  'Callback','wcn=wcn+1;,S=movespec(f,fmax,fname,fs,h,n,0,t,w,wcn,x,y);');

h(22) = uicontrol('Style','pushbutton',...
  'String','<-',...
  'Position',[421 261 50 20],...
  'BackgroundColor',[1 1 1],...
  'Callback','wcn=wcn-1;,S=movespec(f,fmax,fname,fs,h,n,0,t,w,wcn,x,y);');
```

```
h(23) = uicontrol('Style','text',...
   'String','Move analysis window',...
   'Position',[471 261 120 20],...
   'BackgroundColor',[1 1 1]);

% Sixth, create the text boxes for displaying the vocal tract formants derived
% from LPC analysis of the acoustic signal.

h(24) = uicontrol('Style','text',...
   'String','Vocal tract formants (LPC)',...
   'Position',[661 151 70 30],...
   'BackgroundColor',[1 1 1]);

h(25) = uicontrol('Style','text',...
   'String',strcat('F1=',num2str(flpc(posflpc(1)),'%3.0f')),...
   'Position',[661 121 70 20],...
   'BackgroundColor',[1 1 1]);

h(26) = uicontrol('Style','text',...
   'String',strcat('F2=',num2str(flpc(posflpc(2)),'%4.0f')),...
   'Position',[661 101 70 20],...
   'BackgroundColor',[1 1 1]);

h(27) = uicontrol('Style','text',...
   'String',strcat('F3=',num2str(flpc(posflpc(3)),'%4.0f')),...
   'Position',[661 81 70 20],...
   'BackgroundColor',[1 1 1]);

% Lastly, create the "Click to analyze...", "Get...", "Print", and "Done",
% "Done" pushbuttons to determine which set of axes is being analyzed, whether
% an additional point needs to be selected in that domain, and to signal the
% end of the analysis.

h(28) = uicontrol('Style','pushbutton',...
   'String',analysischoice(dom+1,:),...
   'Position',[421 201 220 20],...
   'BackgroundColor',[1 1 1],...
   'Callback','[dom,gd]=domain(analysischoice,dom,gd,gdtext,h,sm);');

h(29) = uicontrol('Style','pushbutton',...
   'String',gdtext(gd,:),...
   'Position',[421 81 60 80],...
   'BackgroundColor',[1 1 1],...
   'Callback','[afn,S,wcn]=gdp(afn,dom,f,fmax,fname,fs,h,n,S,sm,t,w,wn,wcn,x,y);');

h(30) = uicontrol('Style','pushbutton',...
   'String','Print',...
   'Position',[661 241 70 40],...
   'BackgroundColor',[1 1 1],...
   'Callback','print(1,''-dmeta'',''-zbuffer'');');

h(31) = uicontrol('Style','pushbutton',...
   'String','Done',...
   'Position',[661 191 70 40],...
```

```
    'BackgroundColor',[1 1 1],...
    'Callback','clear;, close(1)');

  h(32) = uicontrol('Style','pushbutton',...
    'String','Update acoustic signal',...
    'Position',[421 231 220 20],...
    'BackgroundColor',[1 1 1],...
    'Callback','update(f,fmax,fs,h,n,t,wn,wcn,z);');
end
```

```
% M-File LPLVLONG.M - Larynx Postition and Lung Volume data analysis
% using a LONG time window (e.g., 512 or 1024 points for Fs = 20kHz)
% adapted from m-file SPAN.M
%
% Harold Cheyne
% 21 December 2000
% revised 27 March 2001

[fname,fpath] = uigetfile('*.mat','Enter the filename for the analysis.');
if fname ~= 0
  load(strcat(fpath,fname));
  lacc = length(acc);                          % lacc = length of data (samples)
  wnch = inputdlg('Enter the analysis window length.');
  wn = str2num(char(wnch));                    % wn = length of window for fft
  w = zeros(1,wn);                                 % Initialize the window vector.
  w(2:wn-1) = hanning(wn-2);                   % Use a Hanning window for the spectra
  wcn = round(lacc./2);                        % Initialize wcn = window center index
  n1 = wcn - wn/2 + 1;                             % n1 = window start point
  n2 = n1 + wn - 1;                                % n2 = window end point
  nsch = inputdlg('Enter an odd number of points for the spectral smoothing function.');
  ns = str2num(char(nsch));
  x = acc - mean(acc);                         % operate on the AC acceleration only
  y = egg;                                             % rename egg signal for functions below
  z = aco;                                             % rename acoustic signal for functions below
  afn = zeros(4,2);                                % Initialize amp. & freq. output vector.
  h = zeros(31,1);                             % Initialize handle vector for all handles.
  % h(1) is entire acceleration signal     h(16) is for text box F1'amp.
  % h(2) is acceleration signal zoom           h(17) is for text box Z1'amp.
  % h(3) is acoustic signal zoom               h(18) is for text box F2'amp.
  % h(4) is acceleration spectrum              h(19) is for text box Z2'amp.
  % h(5) is acoustic spectrum                  h(20) is for text box mean EGG voltage
  % h(6) is for button F1'                     h(21) is for button -> (move window right)
  % h(7) is for button Z1'                     h(22) is for button <- (move window left)
  % h(8) is for button F2'                     h(23) is for text box Move Analysis Window
  % h(9) is for button Z2'                     h(24) is for text box Vocal tract...
  % h(10) is for title "Freq. (Hz)"        h(25) is for text box F1=...
  % h(11) is for title "Amp. (dB)"             h(26) is for text box F2=...
  % h(12) is for text box F1'freq.         h(27) is for text box F3=...
  % h(13) is for text box Z1'freq.      h(28) is for button Click to analyze...
  % h(14) is for text box F2'freq.    h(29) is for button Get...
  % h(15) is for text box Z2'freq.         h(30) is for button Print
  %                                h(31) is for button Done
```

```
t = 0:1/fs:(lacc-1)/fs;                     % Create a time index for the signal.
n = 2048;                                      % n = length of fft
S = ones(n,1);                                 % Initialize spectrum vector.
f = 0:fs/n:fs/2-fs/n;                   % Create the frequency vector for the spectra
fmax = round(3000.*n./fs);                  % fmax = highest freq. index to display

% sm is a scalar corresponding to the spectral measure being estimated, with
% 1 = f1', 2 = z1', 3 = f2', 4 = z2'.
sm = 1;

% gd is a scalar corresponding to the state of the push button for getting data,
% and gdtext holds the text for the push button for each state.
% 1 = Get F1', 2 = Get Z1', 3 = Get F2', 4 = Get Z2', 5 = Get Twc (window center)
gd = 5;
gdtext = ['Get F1 ''';'Get Z1 ''';'Get F2 ''';'Get Z2 ''';'Get Twc'];

% analysischoice holds the text to go in a button that allows the user to
% choose between using the mouse input to locate the time window or get
% data from the acceleration spectrum.
analysischoice = ['Click to analyze frequency domain'];
analysischoice(2,:) = ['  Click to analyze time domain    '];

dom = 0;                                                 % 0 = time, 1 = frequency

% Create the figure for the GUI.
figure(1)
clf
set(1,'Renderer','OpenGL','Position',[1 29 1152 768],'Resize','off','Color',[1 1 1])
h(1) = subplot(3,1,1);                        % h(1) = entire acceleration time signal,
plot(t,x,'k-');                               % used for reference, to see where spectra
   hold on
   line([wcn/fs wcn/fs],[min(x) max(x)],'Color',[0 0 0],'LineWidth',2) % and window
grid                                                     % are located relative to entire signal.
xlabel('Time (sec)');
ylabel('cm/s^2');                                   % The acceleration signal is calibrated.
title(strcat(fname,',acceleration signal'));
axis([0 t(length(t)) min(x) max(x)])

h(2) = subplot(3,2,3);                        % h(2) = close-up of the few periods
t1 = n1-wn;
t2 = n2+wn;
plot(t(t1:t2),x(t1:t2),'k',t(n1:n2),w.*max(x(t1:t2)),'k');
   axis([t(t1) t(t2) min(x(t1:t2)) max(x(t1:t2))]);
   grid
   xlabel('Time (sec)');
   ylabel('cm/s^2');
   title(strcat('acceleration signal,','analysis window length=',num2str(wn)));

h(3) = subplot(3,2,4);                        % h(3) = close-up of the few periods
   plot(t(t1:t2),z(t1:t2),'k'); % Re-plot acoustic time signal,
   hold on
   line([wcn/fs wcn/fs],[min(z(t1:t2)) max(z(t1:t2))],'Color',[0 0 0],'LineWidth',2);
   axis([t(n1-wn*2) t(n2+wn*2) min(z(t1:t2)) max(z(t1:t2))])
   grid
   xlabel('Time (sec)');
```

```
   ylabel('dyne/cm^2');
   title('acoustic signal');

h(4) = subplot(3,3,7);                      % h(4) = acceleration spectrum
X = 2.*fft(x(n1:n2).*w,n)./n;               % Take the FFT of the new windowed portion.
S = abs(X(1:n/2)).*2.*pi.*f;                % Calculate the spectrum magnitude, and a
Ss = smooth([S S(n/2:-1:1)],ns);            % smoothed version of it, and plot them.
plot(f(2:fmax),20.*log10(S(2:fmax)),'k');
hold on
plot(f(round(ns/2):fmax-fix(ns/2)),20.*log10(Ss(1:fmax-(ns-1)))+20,'k');
   ylim([min(20.*log10(S(2:fmax)))-5 max(20.*log10(Ss(1:fmax-(ns-1))))+25]);
   grid
   ylabel('dB');
   xlabel('Frequency (Hz)');
   title(strcat('|fft(V_{acc}-mean(V_{acc}))|*2*pi*f, subject:',fname(1:2)));

h(5) = subplot(3,3,9);                      % h(5) = acoustic LPC spectrum
a = autolpc(z(wcn-n/2+1:wcn+n/2),20);       % Perform LPC on a n-point window around wcn
   az = ones(21,1);                         % Create a vector for holding coefficients
az(2:21) = -a;                              % and put the LPC coefficients in it for
LPC = freqz(1,az,f,fs);                     % use with the function FREQZ.M.
plot(f(2:fmax),20.*log10(abs(LPC(2:fmax))),'k');   % Plot the LPC spectrum.
ylim([min(20.*log10(LPC(2:fmax)))-5 max(20.*log10(LPC(2:fmax)))+5]);
grid
ylabel('dB');
xlabel('Frequency (Hz)');
title('LPC spectrum of acoustic signal');
flpc = sort(angle(roots(az)).*fs./(2.*pi));   % Get the pole frequencies in Hz.
posflpc = find(flpc>0);                       % Then find the positive ones.

% Create the spectral measurement interactive display.

% First create the buttons to choose which spectral measure to make.
buttonpos = [501 141 40 20;...
    501 121 40 20;...
    501 101 40 20;...
    501 81 40 20];
buttontext = ['F1 ';'Z1 ';'F2 ';'Z2 '];
for m = 1:4,
  h(m+5) = uicontrol('Style','radiobutton',...
    'String',buttontext(m,:),...
    'Position',buttonpos(m,:),...
    'BackgroundColor',[1 1 1],...
    'SelectionHighlight','off');
end
set(h(6),'Value',1,'Callback','sm=1;,[dom,gd]=getfz(dom,gd,gdtext,h,sm);');
set(h(7),'Callback','sm=2;,[dom,gd]=getfz(dom,gd,gdtext,h,sm);');
set(h(8),'Callback','sm=3;,[dom,gd]=getfz(dom,gd,gdtext,h,sm);');
set(h(9),'Callback','sm=4;,[dom,gd]=getfz(dom,gd,gdtext,h,sm);');

% Second, create the labels above the frequency and amplitude values for
% the measures.
h(10) = uicontrol('Style','text',...
  'String','Freq. (Hz)',...
  'BackgroundColor',[1 1 1],...
```

```
  'Position',[541 161 50 20]);
h(11) = uicontrol('Style','text',...
  'String','Amp. (dB)',...
  'BackgroundColor',[1 1 1],...
  'Position',[591 161 50 20]);

% Third, create the text boxes to hold the frequency and amplitude values measured.
freqpos = [541 141 50 20;...
    541 121 50 20;...
    541 101 50 20;...
    541 81 50 20];
freqtext = [' ';' ';' ';' '];
amppos = [591 141 50 20;...
    591 121 50 20;...
    591 101 50 20;...
    591 81 50 20];
amptext = [' ';' ';' ';' '];
for m = 1:4,
  h(m+11) = uicontrol('Style','text',...
    'String',freqtext(m,:),...
    'BackgroundColor',[1 1 1],...
    'Position',freqpos(m,:));
  h(m+15) = uicontrol('Style','text',...
    'String',amptext(m,:),...
    'BackgroundColor',[1 1 1],...
    'Position',amppos(m,:));
end

% Fourth, create the text box to display the mean EGG voltage for the
% time window selected.
h(20) = uicontrol('Style','text',...
  'String',strcat('Mean(EGG)=',num2str(mean(y(n1:n2)),'%2.3f')),...
  'BackgroundColor',[1 1 1],...
  'Position',[421 171 110 20]);

% Fifth, create the buttons for moving the time analysis window one point
% at a time, and the text above them.

h(21) = uicontrol('Style','pushbutton',...
  'String','->',...
  'Position',[591 261 50 20],...
  'BackgroundColor',[1 1 1],...
  'Callback','wcn=wcn+1;,S=movespec(f,fmax,fname,fs,h,n,ns,t,w,wcn,x,y);');

h(22) = uicontrol('Style','pushbutton',...
  'String','<-',...
  'Position',[421 261 50 20],...
  'BackgroundColor',[1 1 1],...
  'Callback','wcn=wcn-1;,S=movespec(f,fmax,fname,fs,h,n,ns,t,w,wcn,x,y);');

h(23) = uicontrol('Style','text',...
  'String','Move analysis window',...
  'BackgroundColor',[1 1 1],...
  'Position',[471 261 120 20]);
```

```
% Sixth, create the text boxes for displaying the vocal tract formants derived
% from LPC analysis of the acoustic signal.

h(24) = uicontrol('Style','text',...
   'String','Vocal tract formants (LPC)',...
   'BackgroundColor',[1 1 1],...
   'Position',[661 151 70 30]);

h(25) = uicontrol('Style','text',...
   'String',strcat('F1=',num2str(flpc(posflpc(1)),'%3.0f')),...
   'BackgroundColor',[1 1 1],...
   'Position',[661 121 70 20]);

h(26) = uicontrol('Style','text',...
   'String',strcat('F2=',num2str(flpc(posflpc(2)),'%4.0f')),...
   'BackgroundColor',[1 1 1],...
   'Position',[661 101 70 20]);

h(27) = uicontrol('Style','text',...
   'String',strcat('F3=',num2str(flpc(posflpc(3)),'%4.0f')),...
   'BackgroundColor',[1 1 1],...
   'Position',[661 81 70 20]);

% Lastly, create the "Click to analyze...", "Get...", "Print", and "Done",
% "Done" pushbuttons to determine which set of axes is being analyzed, whether
% an additional point needs to be selected in that domain, and to signal the
% end of the analysis.

h(28) = uicontrol('Style','pushbutton',...
   'String',analysischoice(dom+1,:),...
   'Position',[421 201 220 20],...
   'BackgroundColor',[1 1 1],...
   'Callback','[dom,gd]=domain(analysischoice,dom,gd,gdtext,h,sm);');

h(29) = uicontrol('Style','pushbutton',...
   'String',gdtext(gd,:),...
   'Position',[421 81 60 80],...
   'BackgroundColor',[1 1 1],...
   'Callback','[afn,S,wcn]=gdplong(afn,dom,f,fmax,fname,fs,h,n,ns,S,sm,t,w,wn,wcn,x,y);');

h(30) = uicontrol('Style','pushbutton',...
   'String','Print',...
   'Position',[661 241 70 40],...
   'BackgroundColor',[1 1 1],...
   'Callback','print(1,''-dmeta'',''-zbuffer'');');

h(31) = uicontrol('Style','pushbutton',...
   'String','Done',...
   'Position',[661 191 70 40],...
   'BackgroundColor',[1 1 1],...
   'Callback','clear;, close(1)');

h(32) = uicontrol('Style','pushbutton',...
   'String','Update acoustic signal',...
   'Position',[421 231 220 20],...
```

```
    'BackgroundColor',[1 1 1],...
    'Callback','update(f,fmax,fs,h,n,t,wn,wcn,z);');
end
```

```
function S = movespec(f,fmax,fname,fs,h,n,ns,t,w,wcn,x,y)

% This function is used exclusively by the MATLAB functions LPLV.M
% and LPLVLONG.M, in response to the user activating the "Get Twc"
% button, or by clicking on the -> or <- (move window right and
% left) button.
%
% Harold Cheyne
% 26 January 2001
% revised 30 March 2001

wn = length(w);                          % wn = window length
n1 = wcn - wn/2 + 1;                     % n1 = window start point
n2 = n1 + wn - 1;                        % n2 = window end point

subplot(h(1));                           % Select entire acceleration time plot
hold off
plot(t,x,'k-')
hold on
line([wcn/fs wcn/fs],[min(x) max(x)],'Color',[0 0 0],'LineWidth',2) % and window
axis([t(1) t(length(x)) min(x) max(x)])                             % marker.
grid
xlabel('Time (sec)');
ylabel('cm/s^2');
title(strcat(fname,',acceleration signal'));

subplot(h(2));                           % Select acceleration time signal plot
t1 = n1-wn;
t2 = n2+wn;
plot(t(t1:t2),x(t1:t2),'k',t(n1:n2),w.*max(x(t1:t2)),'k');
axis([t(t1) t(t2) min(x(t1:t2)) max(x(t1:t2))]);
grid
xlabel('Time (sec)');
ylabel('cm/s^2');
title(strcat('acceleration signal,','analysis window length=',num2str(wn)));

subplot(h(4));                           % Select the acceleration spectrum plot.
X = 2.*fft(x(n1:n2).*w,n)./n;            % Take the FFT of the new windowed portion.
S = abs(X(1:n/2)).*2.*pi.*f;             % Calculate the spectrum magnitude,
if wn >= 512,                            % If a long time window is being used,
  Ss = smooth([S S(n/2:-1:1)],ns);       % then also smooth the spectrum, and
  hold off
  plot(f(2:fmax),20.*log10(S(2:fmax)),'k')
  hold on
  plot(f(round(ns/2):fmax-fix(ns/2)),20.*log10(Ss(1:fmax-(ns-1)))+20,'k');
  ylim([min(20.*log10(S(2:fmax)))-5 max(20.*log10(Ss(1:fmax-(ns+1))))+25]);
else                                     % plot both the smoothed and regular
  plot(f(2:fmax),20.*log10(S(2:fmax)),'k');  % spectra, otherwise just plot the regular.
  ylim([min(20.*log10(S(2:fmax)))-5 max(20.*log10(S(2:fmax)))+5]);
```

```
end
grid
ylabel('dB');
xlabel('Frequency (Hz)');
title(strcat('|fft(V_{acc}-mean(V_{acc}))|*2*pi*f, subject:',fname(1:2)));
set(h(20),'String',strcat('Mean(EGG)=',num2str(mean(y(n1:n2)),'%2.3f')));
```

```
function [A,acc,aco,fname,fs,P,t,wa,wcn,wp]=newdata(h)

% Function NEWDATA.M
%
% Usage:  [A,acc,aco,fname,fs,P,t,wa,wcn,wp]=newdata(h)
% Input:  h, the graphics handles from the active GUI
% Output: A, the FFT of the acceleration
%         acc, the acceleration signal in the file
%         aco, the acoustic signal in the file
%         fname, the file name
%         fs, the sampling frequency of the signals
%         P, the FFT of the acoustic signal
%         t, a vector holding the time values for each signal index
%         wa, the Hanning window for the acceleration signal
%         wcn, the center index of the window on the signals
%         wp, the Hamming window for the microphone pressure signal
%
% Summary: This MATLAB function is called from the m-file "ACCSPL1.M",
% and it prompts the user for a new file name, after which it opens
% that file and places the appropriate acceleration and acoustic
% time signals and spectra into the active GUI.
%
% Harold Cheyne
% 30 May 2001
% revised 1 June 2001

[fname,fpath] = uigetfile('*.mat','Enter the filename for the analysis.');
load(strcat(fpath,fname));
acc = acc - mean(acc);                    % Remove the DC from the acceleration signal
lacc = length(acc);                       % lacc = length of data (samples)
t = 0:1/fs:(lacc-1)/fs;                   % Create a time index for the signal.
wcn = round(lacc./2);                     % Initialize wcn = window center index
wla = str2double(get(h(16),'String'));    % wla = acceleration window length
wa = zeros(1,wla);                        % Initialize the acceleration window vector.
wa(1:wla-1) = hanning(wla-1,'periodic').*2; % Use a Hanning window for the accel.
n1a = wcn - fix(wla./2);                  % Calculate the start and end points for the
n2a = n1a + wla - 1;                      % acceleration FFT.
wlp = str2double(get(h(18),'String'));    % wlp = microphone pressure window length
wp = hamming(wlp)'.*1.8525;               % Initialize the microphone pressure window
n1p = wcn - fix(wlp./2);                  % Calculate the start and end points for the
n2p = n1p + wlp - 1;                      % acoustic FFT.
n = str2double(get(h(20),'String'));  % Get the FFT length from the GUI.
f = 0:fs/n:fs/2-fs/n;                     % f = frequencies in FFT
if length(get(h(11),'String')) == 4       % If the ZOOM button in the GUI says "Zoom",
  subplot(h(3));                          % then plot the entire acceleration and acoustic
  hold off
```

```
    plot(t,acc,'k-');                          % signals with the window centers marked by
    hold on                                       % vertical lines.
    line([wcn/fs wcn/fs],[min(acc) max(acc)],'Color',[0 0 0],'LineWidth',2)
  axis([0 t(length(t)) min(acc) max(acc)])
  grid
    xlabel('Time (sec)');
  ylabel('acc(t), cm/sec^2');
  subplot(h(4));
  hold off
  plot(t,aco,'k-');
    hold on
    line([wcn/fs wcn/fs],[min(aco) max(aco)],'Color',[0 0 0],'LineWidth',2)
    axis([0 t(length(t)) min(aco) max(aco)])
    grid
    xlabel('Time (sec)');
  ylabel('p_{mic}, dyne/cm^2');
else                                              % Otherwise, the time plots should show the
  subplot(h(3));                             % windowed portions of the acceleration and
  hold off
    plot(t(n1a:n2a),acc(n1a:n2a),'k-');   % acoustic signals with the window centers
    hold on                                       % marked by vertical lines.
  line([wcn/fs wcn/fs],[min(acc(n1a:n2a)) max(acc(n1a:n2a))],'Color',[0 0 0],'LineWidth',2)
  plot(t(n1a:n2a),wa.*max(acc(n1a:n2a))./2,'k-')
  axis([t(n1a) t(n2a) min(acc(n1a:n2a)) max(acc(n1a:n2a))])
  grid
    xlabel('Time (sec)');
  ylabel('acc(t), cm/sec^2');
  subplot(h(4));
  hold off
  plot(t(n1p:n2p),aco(n1p:n2p),'k-');
    hold on
  line([wcn/fs wcn/fs],[min(aco(n1p:n2p)) max(aco(n1p:n2p))],'Color',[0 0 0],'LineWidth',2)
  plot(t(n1p:n2p),wp.*max(aco(n1p:n2p))./1.8525,'k-')
    axis([t(n1p) t(n2p) min(aco(n1p:n2p)) max(aco(n1p:n2p))])
    grid
    xlabel('Time (sec)');
  ylabel('p_{mic}, dyne/cm^2');
end
subplot(h(1));                               % And plot the acceleration and acoustic
A = 2.*fft(acc(n1a:n2a).*wa,n)./n;     % spectra using the windows created above.
plot(f,20.*log10(abs(A(1:n./2))),'k-');
grid
axis([0 3000 min(20.*log10(abs(A(1:308)))) max(20.*log10(abs(A(1:308))))+5])
xlabel('Frequency (Hz)')
ylabel('|ACC|, dB re 1 cm/sec^2')
title(strcat(fname,',AC acceleration signal'));
subplot(h(2));
P = 2.*fft(aco(n1p:n2p).*wp,n)./n;
plot(f,20.*log10(abs(P(1:n./2))),'k-');
grid
axis([0 3000 min(20.*log10(abs(P(1:308)))) max(20.*log10(abs(P(1:308))))+5])
xlabel('Frequency (Hz)')
ylabel('|P_{mic}|, dB re 1 dyne/cm^2')
title(strcat(fname,',microphone pressure signal'));
dBSPL = 20.*log10(sqrt(mean(aco(n1p:n2p).^2))./2e-4);
```

```
set(h(61),'String',num2str(dBSPL,'%3.1f'));
```

```
function [A,acc,aco,fname,fs,t,wcn]=newdata2(h,plotnum)

% Function NEWDATA2.M
%
% Usage:  [A,acc,aco,fname,fs,P,t,wa,wcn,wp]=newdata2(h)
% Input:  h, the graphics handles from the active GUI
%        plotnum, a number corresponding to which set of plots to operate on (1-5)
% Output: A, the FFT of the acceleration
%        acc, the acceleration signal in the file
%        aco, the acoustic signal in the file
%        fname, the file name
%        fs, the sampling frequency of the signals
%        t, a vector holding the time values for each signal index
%        wcn, the center index of the window on the signals
%
% Summary: This MATLAB function is called from the m-file "ACCSPL3.M",
% and it prompts the user for a new file name, after which it opens
% that file and places the appropriate acceleration and acoustic
% time signals and spectra into the active GUI.
%
% Harold Cheyne
% 21 June 2001 - adapted from newdata.m
% revised 22 June 2001

switch plotnum
case 1, h1 = h(1);,h2 = h(6);,h3 = h(7);,h4 = h(17);,h5 = h(20);
case 2, h1 = h(2);,h2 = h(8);,h3 = h(9);,h4 = h(23);,h5 = h(26);
case 3, h1 = h(3);,h2 = h(10);,h3 = h(11);,h4 = h(29);,h5 = h(32);
case 4, h1 = h(4);,h2 = h(12);,h3 = h(13);,h4 = h(35);,h5 = h(38);
case 5, h1 = h(5);,h2 = h(14);,h3 = h(15);,h4 = h(41);,h5 = h(44);
end
[fname,fpath] = uigetfile('*.mat','Enter the filename for the analysis.');
load(strcat(fpath,fname));
acc = acc - mean(acc);                    % Remove the DC from the acceleration signal
lacc = length(acc);                       % lacc = length of data (samples)
t = 0:1/fs:(lacc-1)/fs;                   % Create a time index for the signal.
wcn = round(lacc./2);                     % Initialize wcn = window center index
wla = 2048;                                   % wla = acceleration window length
wa = zeros(1,wla);                        % Initialize the acceleration window vector.
wa(1:wla-1) = hanning(wla-1,'periodic').*2;% Use a Hanning window for the acceleration
n1a = wcn - fix(wla./2);                  % Calculate the start and end points for the
n2a = n1a + wla - 1;                          % acceleration FFT.
wlp = 2048;                                   % wlp = microphone pressure window length
wp = hamming(wlp)'.*1.8525;               % Initialize the microphone pressure window
n1p = wcn - fix(wlp./2);                  % Calculate the start and end points for the
n2p = n1p + wlp - 1;                          % acoustic FFT.
n = 2048;                                     % Get the FFT length from the GUI.
f = 0:fs/n:fs/2-fs/n;                     % f = frequencies in FFT
if length(get(h4,'String')) == 4          % If the ZOOM button in the GUI says "Zoom",
  subplot(h2);                                % then plot the entire acceleration and acoustic
  hold off
```

```
    plot(t,acc,'k-');                           % signals with the window centers marked by
    hold on                                     % vertical lines.
    line([wcn/fs wcn/fs],[min(acc) max(acc)],'Color',[0 0 0],'LineWidth',2)
  axis([0 t(length(t)) min(acc) max(acc)])
  grid
  ylabel('cm/sec^2');
  subplot(h3);
  hold off
  plot(t,aco,'k-');
    hold on
    line([wcn/fs wcn/fs],[min(aco) max(aco)],'Color',[0 0 0],'LineWidth',2)
    axis([0 t(length(t)) min(aco) max(aco)])
  grid
  if plotnum == 5
   xlabel('Time (sec)');
  end
  ylabel('dyne/cm^2');
else                                            % Otherwise, the time plots should show the
  subplot(h2);                                  % windowed portions of the acceleration and
  hold off
    plot(t(n1a:n2a),acc(n1a:n2a),'k-');   % acoustic signals with the window centers
    hold on                                     % marked by vertical lines.
  line([wcn/fs wcn/fs],[min(acc(n1a:n2a)) max(acc(n1a:n2a))],'Color',[0 0 0],'LineWidth',2)
  plot(t(n1a:n2a),wa.*max(acc(n1a:n2a))./2,'k-')
  axis([t(n1a) t(n2a) min(acc(n1a:n2a)) max(acc(n1a:n2a))])
  grid
  ylabel('cm/sec^2');
  subplot(h3);
  hold off
  plot(t(n1p:n2p),aco(n1p:n2p),'k-');
    hold on
  line([wcn/fs wcn/fs],[min(aco(n1p:n2p)) max(aco(n1p:n2p))],'Color',[0 0 0],'LineWidth',2)
  plot(t(n1p:n2p),wp.*max(aco(n1p:n2p))./1.8525,'k-')
    axis([t(n1p) t(n2p) min(aco(n1p:n2p)) max(aco(n1p:n2p))])
  grid
  if plotnum == 5
   xlabel('Time (sec)');
  end
  ylabel('dyne/cm^2');
end
subplot(h1);                                    % And plot the acceleration and acoustic
A = 2.*fft(acc(n1a:n2a).*wa,n)./n;  % spectra using the windows created above.
plot(f,20.*log10(abs(A(1:n./2))),'k-');
grid
axis([0 2000 min(20.*log10(abs(A(1:205)))) max(20.*log10(abs(A(1:205))))+5])
if plotnum == 5
  xlabel('Frequency (Hz)')
end
ylabel('dB re 1 cm/sec^2')
title(fname);
dBSPL = 20.*log10(sqrt(mean(aco(n1p:n2p).^2))./2e-4);
set(h5,'String',strcat('dB SPL=',num2str(dBSPL,'%3.1f')));
```

```
function [A,x,y,z,aco,fname,fs,F,t,wa,wcn,wf]=newdata3(h)

% Function NEWDATA3.M
%
% Usage:  [A,x,y,z,aco,fname,fs,P,t,wa,wcn,wf]=newdata3(h)
% Input:  h, the graphics handles from the active GUI
% Output: A, the FFT of the acceleration
%         x, the AC acceleration signal in the file
%         y, the EGG signal from the file
%         z, the airflow signal from the file
%         aco, the acoustic signal in the file
%         fname, the file name
%         fs, the sampling frequency of the signals
%         F, the FFT of the airflow signal
%         t, a vector holding the time values for each signal index
%         wa, the Hanning window for the acceleration signal
%         wcn, the center index of the window on the signals
%         wf, the Hamming window for the microphone pressure signal
%
% Summary: This MATLAB function is called from the m-file "ACCFLO3.M",
% and it prompts the user for a new file name, after which it opens
% that file and places the appropriate acceleration and airflow
% time signals and spectra into the active GUI.
%
% Harold Cheyne
% 30 May 2001
% revised 1 June 2001
% adapted from NEWDATA.M - 5 June 2001
% revised 17 September 2001 for thesis document (i.e., black & white plots), and
% to correct for the window gain problem (see accflo3)

[fname,fpath] = uigetfile('*.mat','Enter the filename for the analysis.');
load(strcat(fpath,fname));
x = acc - mean(acc);                        % Remove the DC from the acceleration signal
y = egg;
z = flo;
lacc = length(acc);                         % lacc = length of data (samples)
t = 0:1/fs:(lacc-1)/fs;                     % Create a time index for the signal.
wcn = round(lacc./2);                       % Initialize wcn = window center index
wla = str2double(get(h(14),'String'));      % wla = acceleration window length
wa = zeros(1,wla);                          % Initialize the acceleration window vector.
wa(1:wla-1) = hanning(wla-1,'periodic')./2; % Use a Hanning window for the accel.
n1a = wcn - fix(wla./2);                    % Calculate the start and end points for the
n2a = n1a + wla - 1;                        % acceleration FFT.
wlf = str2double(get(h(50),'String'));      % wlf = airflow window length
wf = hamming(wlf)'.*1.8525;                 % Initialize the microphone pressure window
n1f = wcn - fix(wlf./2);                    % Calculate the start and end points for the
n2f = n1f + wlf - 1;                        % acoustic FFT.
n = str2double(get(h(57),'String'));  % Get the FFT length from the GUI.
f = 0:fs/n:fs/2-fs/n;                       % f = frequencies in FFT
if length(get(h(11),'String')) == 4         % If the ZOOM button in the GUI says "Zoom",
  subplot(h(3));                            % then plot the entire acceleration and acoustic
  hold off
    plot(t,x,'k-');                         % signals with the window centers marked by
    hold on                                 % vertical lines.
```

```
    line([wcn/fs wcn/fs],[min(x) max(x)],'Color',[0 0 0],'LineWidth',2)
  axis([0 t(length(t)) min(x) max(x)])
  grid
    xlabel('Time (sec)');
  ylabel('acc(t), cm/sec^2');
  subplot(h(4));
  hold off
  plot(t,z,'k-');
    hold on
    line([wcn/fs wcn/fs],[min(z) max(z)],'Color',[0 0 0],'LineWidth',2)
    axis([0 t(length(t)) min(z) max(z)])
    grid
    xlabel('Time (sec)');
  ylabel('v_{vm}, cm^3/sec');
else                                          % Otherwise, the time plots should show the
  subplot(h(3));                              % windowed portions of the acceleration and
  hold off
    plot(t(n1a:n2a),x(n1a:n2a),'k-');         % acoustic signals with the window centers
    hold on                                   % marked by vertical lines.
  line([wcn/fs wcn/fs],[min(x(n1a:n2a)) max(x(n1a:n2a))],'Color',[0 0 0],'LineWidth',2)
  plot(t(n1a:n2a),wa.*max(x(n1a:n2a))./2,'k-')
  axis([t(n1a) t(n2a) min(x(n1a:n2a)) max(x(n1a:n2a))])
  grid
    xlabel('Time (sec)');
  ylabel('acc(t), cm/sec^2');
  subplot(h(4));
  hold off
  plot(t(n1f:n2f),z(n1f:n2f),'k-');
    hold on
  line([wcn/fs wcn/fs],[min(z(n1f:n2f)) max(z(n1f:n2f))],'Color',[0 0 0],'LineWidth',2)
  plot(t(n1f:n2f),wf.*max(z(n1f:n2f))./1.8525,'k-')
    axis([t(n1f) t(n2f) min(z(n1f:n2f)) max(z(n1f:n2f))])
    grid
    xlabel('Time (sec)');
  ylabel('v_{vm}, cm^3/sec');
end
subplot(h(1));                                % And plot the acceleration and acoustic
A = 2.*fft(x(n1a:n2a).*wa,n)./n;              % spectra using the windows created above.
f2 = round(3000.*n./fs);
plot(f,20.*log10(abs(A(1:n./2))),'k');
grid
axis([0 3000 min(20.*log10(abs(A(1:308)))) max(20.*log10(abs(A(1:308))))+5])
ytickmin = round(min(20.*log10(abs(A(1:f2))))./10).*10;
ytickmax = round(max(20.*log10(abs(A(1:f2))))./10).*10;
set(gca,'YTick',ytickmin:10:ytickmax);
xlabel('Frequency (Hz)')
ylabel('|ACC|, dB re 1 cm/sec^2')
title(strcat(fname,',AC acceleration signal'));
subplot(h(2));
F = 2.*fft(z(n1f:n2f).*wf,n)./n;
plot(f,20.*log10(abs(F(1:n./2))),'k');
grid
axis([0 3000 min(20.*log10(abs(F(1:308)))) max(20.*log10(abs(F(1:308))))+5])
ytickmin = round(min(20.*log10(abs(F(1:f2))))./10).*10;
ytickmax = round(max(20.*log10(abs(F(1:f2))))./10).*10;
```

```
set(gca,'YTick',ytickmin:10:ytickmax);
xlabel('Frequency (Hz)')
ylabel('|V_{vm}|, dB re 1 cm^3/sec')
title(strcat(fname,',airflow signal'));
```

```
function [A,acc,F,flo,fname,fs,t,wa,wcn,wf]=newdata4(h)

% Function NEWDATA4.M
%
% Usage:  [A,acc,F,flo,fname,fs,t,wa,wcn,wf]=newdata4(h)
% Input:  h, the graphics handles from the active GUI
% Output: A, the FFT of the acceleration
%         acc, the acceleration signal in the file
%         F, the FFT of the flow signal
%         flo, the airflow signal in the file
%         fname, the file name
%         fs, the sampling frequency of the signals
%         t, a vector holding the time values for each signal index
%         wa, the Hanning window for the acceleration signal
%         wcn, the center index of the window on the signals
%         wf, the Hamming window for the airflow signal
%
% Summary: This MATLAB function is called from the m-file "ACCFLO4.M",
% and it prompts the user for a new file name, after which it opens
% that file and places the appropriate acceleration and airflow
% time signals and spectra into the active GUI.
%
% Harold Cheyne
% 26 June 2001
% revised 13 July 2001

[fname,fpath] = uigetfile('*.mat','Enter the filename for the analysis.');
if fname ~= 0                                   % As long as a valid filename is chosen,
  load(strcat(fpath,fname));                    % Load that file into the workspace.
  acc = acc - mean(acc);                        % Remove the DC from the acceleration signal
  lacc = length(acc);                           % lacc = length of data (samples)
  t = 0:1/fs:(lacc-1)/fs;                       % Create a time index for the signal.
  wcn = round(lacc./2);                         % Initialize wcn = window center index
  wla = str2double(get(h(21),'String'));  % wla = acceleration window length
  wa = zeros(1,wla);                            % Initialize the acceleration window vector.
  wa(1:wla-1) = hanning(wla-1,'periodic').*2;     % Use a Hanning window for the accel.
  n1a = wcn - fix(wla./2);                 % Calculate the start and end points for the
  n2a = n1a + wla - 1;                          % acceleration FFT.
  wlf = str2double(get(h(23),'String'));  % wlf = flow window length
  wf = hamming(wlf)'.*1.8525;                   % Define the flow window
  n1f = wcn - fix(wlf./2);                 % Calculate the start and end points for the
  n2f = n1f + wlf - 1;                          % acoustic FFT.
  n = 2048;                                          % FFT length = 2048.
  f = 0:fs/n:fs/2-fs/n;                    % f = frequencies in FFT
  if length(get(h(19),'String')) == 4 % If the ZOOM button in the GUI says "Zoom",
    subplot(h(3));                              % then plot the entire acoustic and inverted
    hold off                                         % acceleration signals with the window centers
    plot(t,-acc,'k-');                     % marked by vertical lines.
```

```
  hold on
  line([wcn/fs wcn/fs],[min(-acc) max(-acc)],'Color',[0 0 0],'LineWidth',2)
  axis([0 t(length(t)) min(-acc) max(-acc)])
  grid
  xlabel('Time (sec)');
  ylabel('-acc(t), cm/sec^2');
  subplot(h(4));
  hold off
  plot(t,flo,'k-');
  hold on
  line([wcn/fs wcn/fs],[min(flo) max(flo)],'Color',[0 0 0],'LineWidth',2)
  axis([0 t(length(t)) min(flo) max(flo)])
  grid
  xlabel('Time (sec)');
  ylabel('v_{vm}(t), cm^3/sec');
else                                        % Otherwise, the time plots should show the
  subplot(h(3));                            % windowed portions of the acoustic and
  hold off                                  % inverted acceleration signals with the
  plot(t(n1a:n2a),-acc(n1a:n2a),'k-');% window centers marked by vertical lines.
  hold on
  accmin = min(-acc(n1a:n2a));
  accmax = max(-acc(n1a:n2a));
  line([wcn/fs wcn/fs],[accmin accmax],'Color',[0 0 0],'LineWidth',2)
  plot(t(n1a:n2a),wa.*accmax./2,'k-')
  axis([t(n1a) t(n2a) accmin accmax])
  grid
  xlabel('Time (sec)');
  ylabel('-acc(t), cm/sec^2');
  subplot(h(4));
  hold off
  plot(t(n1f:n2f),flo(n1f:n2f),'k-');
  hold on
  flomin = min(flo(n1f:n2f));
  flomax = max(flo(n1f:n2f));
  line([wcn/fs wcn/fs],[flomin flomax],'Color',[0 0 0],'LineWidth',2)
  plot(t(n1f:n2f),wf.*flomax./1,8525,'k-')
  axis([t(n1f) t(n2f) flomin flomax])
  grid
  xlabel('Time (sec)');
  ylabel('v_{vm}(t), cm^3/sec');
end
subplot(h(1));                              % And plot the acceleration and flow
A = 2.*fft(acc(n1a:n2a).*wa,n)./n;          % spectra using the windows created above.
plot(f,20.*log10(abs(A(1:n./2))),'k-');
grid
axis([0 3000 min(20.*log10(abs(A(1:308)))) max(20.*log10(abs(A(1:308))))+5])
xlabel('Frequency (Hz)')
ylabel('|ACC|, dB re 1 cm/sec^2')
title(strcat(fname,',AC acceleration signal'));
subplot(h(2));
F = 2.*fft(flo(n1f:n2f).*wf,n)./n;
plot(f,20.*log10(abs(F(1:n./2))),'k-');
grid
axis([0 3000 min(20.*log10(abs(F(1:308)))) max(20.*log10(abs(F(1:308))))+5])
xlabel('Frequency (Hz)')
```

```
  ylabel('|V_{vm}|, dB re 1 cm^3/sec')
  title(strcat(fname,',airflow signal'));
end
```

```
function wcn = newtime3(acc,flow,fname,fs,h,t)

% Harold Cheyne
% 5 April 2001
% revised 14 May 2001
% revised 17 September 2001 to correct for Hanning window gain problem

subplot(h(3));                              % Make acceleration time plot active.
tlima = xlim;                           % Get the time limits on the acceleration plot.
alima = ylim;                           % Get the amplitude limits on the acceleration plot.
subplot(h(4));                              % Make flow time plot active.
tlimf = xlim;                           % Get the time limits on the flow plot.
alimf = ylim;                           % Get the amplitude limits on the flow plot.
again = 1;                                  % again = boolean deciding repeat or not
while again                                 % As long as user chooses new time within the
  [x,y] = ginput(1);                  % acceleration or flow time signal plots, then
  if ((x>tlima(1)) & x<tlima(2) & y>alima(1) & y<alima(2))  % do not repeat
    again = 0;                          % getting a point from mouse input.
  elseif ((x>tlimf(1)) & x<tlimf(2) & y>alimf(1) & y<alimf(2))
    again = 0;
  else
    error('Choose a point within the acceleration or flow time signal plots.');
  end
end
wcn = round(x*fs);                          % new window center = time*fs
wn = str2double(get(h(14),'String'));   % get the window length from the GUI,
w = zeros(1,wn);                            % initialize the acceleration window vector,
w(1:wn-1) = hanning(wn-1,'periodic').*2;   % and define a Hanning window for the spectra.
wnf = str2double(get(h(50),'String'));  % Then get the flow window length from the
wf = hamming(wnf)'.*1.8525;            % GUI and make a Hamming window for the LPC.
% Note that the 1.8525 factor corrects for the Hamming window gain.
t1 = wcn - fix(wn/2);                  % t1 & t2 are the start & end points of the
t2 = t1 + wn - 1;                          % "zoomed" view of the acceleration.
t3 = wcn - fix(wnf/2);                 % t3 & t4 are the start & end points of the
t4 = t3 + wnf - 1;                     % "zoomed" view of the airflow.
zinout = get(h(12),'String');          % Are the time plots zoomed in or out?
if length(zinout) == 4                 % If the zoom button says "Zoom in",
  subplot(h(3))                            % then the entire time plots are shown.
  hold off
  plot(t,acc,'k');
  hold on
  line([wcn/fs wcn/fs],[min(acc) max(acc)],'Color',[0 0 0],'LineWidth',2)
  ylabel('cm/sec^2')
  xlabel('Time (sec)')
  axis([0 t(length(t)) min(acc) max(acc)])
  grid
  subplot(h(4))
  hold off
  plot(t,flow,'k');
```

```
  hold on
  line([wcn/fs wcn/fs],[min(flow) max(flow)],'Color',[0 0 0],'LineWidth',2)
  ylabel('cm^3/sec')
  xlabel('Time (sec)')
  axis([0 t(length(t)) min(flow) max(flow)])
  grid
else
  subplot(h(3));                          % then replot a section of each signal that is
  hold off                                    % twice as long as the window, with the window
  plot(t(t1:t2),acc(t1:t2),'k');          % centered in the view.
    hold on
  plot(t(t1:t2),w.*max(acc(t1:t2))./2,'k');
  line([wcn/fs wcn/fs],[min(acc(t1:t2)) max(acc(t1:t2))],'Color',[0 0 0],'LineWidth',2);
  axis([t(t1) t(t2) min(acc(t1:t2)) max(acc(t1:t2))])
    grid
  xlabel('Time (sec)');
  ylabel('a(t) (cm/sec^2)');
  subplot(h(4));
  hold off
  plot(t(t3:t4),flow(t3:t4),'k');
  hold on
  plot(t(t3:t4),wf.*max(flow(t3:t4))./1.8525,'k')
    line([wcn/fs wcn/fs],[min(flow(t3:t4)) max(flow(t3:t4))],'Color',[0 0 0],'LineWidth',2);
    axis([t(t3) t(t4) min(flow(t3:t4)) max(flow(t3:t4))])
    grid
  xlabel('Time (sec)');
  ylabel('cm^3/sec');
end
n = str2double(get(h(57),'String')); % Get the FFT length from the GUI.
f = 0:fs./n:fs./2-fs./n;
f2 = round(3000.*n./fs);
subplot(h(1));                          % h(1) = acceleration spectrum
A = 2.*fft(acc(t1:t2).*w,n)./n;         % A = spectrum of acceleration
plot(f,20.*log10(abs(A(1:n./2))),'k');
grid
xlim([0 3000])
ytickmin = round(min(20.*log10(abs(A(1:f2))))./10).*10;
ytickmax = round(max(20.*log10(abs(A(1:f2))))./10).*10;
set(gca,'YTick',ytickmin:10:ytickmax);
xlabel('Frequency (Hz)')
ylabel('dB re 1 cm/sec^2')
title(strcat(fname,',AC acceleration signal'));
subplot(h(2));                          % h(2) = airflow spectrum
F = 2.*fft(flow(t3:t4).*wf,n)./n;               % F = spectrum of airflow
plot(f,20.*log10(abs(F(1:n./2))),'k');
grid
xlim([0 3000])
ytickmin = round(min(20.*log10(abs(F(1:f2))))./10).*10;
ytickmax = round(max(20.*log10(abs(F(1:f2))))./10).*10;
set(gca,'YTick',ytickmin:10:ytickmax);
xlabel('Frequency (Hz)')
ylabel('dB re 1 cm^3/sec')
title(strcat(fname,',airflow signal'));
```

```
function wcn = newtime4(acc,aco,fname,fs,h,t)

% Harold Cheyne
% 5 April 2001
% revised 14 May 2001
% adapted for use with ACCSPL1.M - 31 May 2001
% revised 1 June 2001

subplot(h(3));                          % Make acceleration time plot active.
tlima = xlim;                         % Get the time limits on the acceleration plot.
alima = ylim;                         % Get the amplitude limits on the acceleration plot.
subplot(h(4));                          % Make flow time plot active.
tlimf = xlim;                         % Get the time limits on the flow plot.
alimf = ylim;                         % Get the amplitude limits on the flow plot.
again = 1;                              % again = boolean deciding repeat or not
while again                             % As long as user chooses new time within the
  [x,y] = ginput(1);                 % acceleration or flow time signal plots, then
  if ((x>tlima(1)) & x<tlima(2) & y>alima(1) & y<alima(2))   % do not repeat
    again = 0;                        % getting a point from mouse input.
  elseif ((x>tlimf(1)) & x<tlimf(2) & y>alimf(1) & y<alimf(2))
    again = 0;
  else
    h1 = errordlg('Choose a point in the acceleration or flow time signal plots.');
    uiwait(h1);
  end
end
wcn = round(x*fs);                      % new window center = time*fs
wla = str2double(get(h(16),'String'));   % wla = acceleration window length
wa = zeros(1,wla);                      % Initialize the acceleration window vector.
wa(1:wla-1) = hanning(wla-1,'periodic').*2; % Use a Hanning window for the accel.
n1a = wcn - fix(wla./2);              % Calculate the start and end points for the
n2a = n1a + wla - 1;                    % acceleration FFT.
wlp = str2double(get(h(18),'String'));   % wlp = microphone pressure window length
wp = hamming(wlp)'.*1.8525;           % Initialize the microphone pressure window
n1p = wcn - fix(wlp./2);              % Calculate the start and end points for the
n2p = n1p + wlp - 1;                    % acoustic FFT.
n = str2double(get(h(20),'String')); % Get the FFT length from the GUI.
f = 0:fs/n:fs/2-fs/n;                 % f = frequencies in FFT
wla = str2double(get(h(16),'String'));   % wla = acceleration window length
wa = zeros(1,wla);                      % Initialize the acceleration window vector.
wa(2:wla-1) = hanning(wla-2);         % Use a Hanning window for the acceleration
n1a = wcn - fix(wla./2);              % Calculate the start and end points for the
n2a = n1a + wla - 1;                    % acceleration FFT.
wlp = str2double(get(h(18),'String'));   % wlp = microphone pressure window length
wp = hamming(wlp)';                     % Initialize the microphone pressure window
n1p = wcn - fix(wlp./2);              % Calculate the start and end points for the
n2p = n1p + wlp - 1;                    % acoustic FFT.
n = str2double(get(h(20),'String')); % Get the FFT length from the GUI.
f = 0:fs/n:fs/2-fs/n;                 % f = frequencies in FFT
if length(get(h(11),'String')) == 4    % If the ZOOM button in the GUI says "Zoom",
  subplot(h(3));                        % then plot the entire acceleration and acoustic
  hold off
    plot(t,acc,'k-');                   % signals with the window centers marked by
    hold on                               % vertical lines.
```

```
    line([wcn/fs wcn/fs],[min(acc) max(acc)],'Color',[0 0 0],'LineWidth',2)
  axis([0 t(length(t)) min(acc) max(acc)])
  grid
    xlabel('Time (sec)');
  ylabel('acc(t), cm/sec^2');
  subplot(h(4));
  hold off
  plot(t,aco,'k-');
    hold on
    line([wcn/fs wcn/fs],[min(aco) max(aco)],'Color',[0 0 0],'LineWidth',2)
    axis([0 t(length(t)) min(aco) max(aco)])
    grid
    xlabel('Time (sec)');
  ylabel('p_{mic}, dyne/cm^2');
else                                            % Otherwise, the time plots should show the
  subplot(h(3));                                % windowed portions of the acceleration and
  hold off
    plot(t(n1a:n2a),acc(n1a:n2a),'k-');   % acoustic signals with the window centers
    hold on                                     % marked by vertical lines.
  line([wcn/fs wcn/fs],[min(acc(n1a:n2a)) max(acc(n1a:n2a))],'Color',[0 0 0],'LineWidth',2)
  plot(t(n1a:n2a),wa.*max(acc(n1a:n2a)),'k-')
  axis([t(n1a) t(n2a) min(acc(n1a:n2a)) max(acc(n1a:n2a))])
  grid
    xlabel('Time (sec)');
  ylabel('acc(t), cm/sec^2');
  subplot(h(4));
  hold off
  plot(t(n1p:n2p),aco(n1p:n2p),'k-');
    hold on
  line([wcn/fs wcn/fs],[min(aco(n1p:n2p)) max(aco(n1p:n2p))],'Color',[0 0 0],'LineWidth',2)
  plot(t(n1p:n2p),wp.*max(aco(n1p:n2p)),'k-')
    axis([t(n1p) t(n2p) min(aco(n1p:n2p)) max(aco(n1p:n2p))])
    grid
    xlabel('Time (sec)');
  ylabel('p_{mic}, dyne/cm^2');
end
subplot(h(1));                                  % And plot the acceleration and acoustic
A = 2.*fft(acc(n1a:n2a).*wa,n)./n;       % spectra using the windows created above.
plot(f,20.*log10(abs(A(1:n./2))),'k-');
grid
axis([0 3000 min(20.*log10(abs(A(1:308)))) max(20.*log10(abs(A(1:308))))+5])
xlabel('Frequency (Hz)')
ylabel('|ACC|, dB re 1 cm/sec^2')
title(strcat(fname,',AC acceleration signal'));
subplot(h(2));
P = 2.*fft(aco(n1p:n2p).*wp,n)./n;
plot(f,20.*log10(abs(P(1:n./2))),'k-');
grid
axis([0 3000 min(20.*log10(abs(P(1:308)))) max(20.*log10(abs(P(1:308))))+5])
xlabel('Frequency (Hz)')
ylabel('|P_{mic}|, dB re 1 dyne/cm^2')
title(strcat(fname,',microphone pressure signal'));
dBSPL = 20.*log10(sqrt(mean(aco(n1p:n2p).^2))./2e-4);
set(h(61),'String',num2str(dBSPL,'%3.1f'));
```

```
function [A,wcn] = newtime5(acc,aco,fname,fs,h,plotnum,t)

% Function NEWTIME5.M
%
% Usage:  [A,wcn] = newtime5(acc,aco,fname,fs,h,plotnum,t)
% Input:  acc, the entire acceleration time signal
%         aco, the entire acoustic time signal
%         fname, the file name
%         fs, the sampling frequency (Hz)
%         h, the handles for the ACCSPL3.M GUI
%         plotnum, a number corresponding to which set of plots to operate on (1-5)
%         t, the vector holding the time values for acc and aco
% Output: A, the new spectrum of the acceleration
%         wcn, a new window center index value
%
% Harold Cheyne
% 5 April 2001
% revised 14 May 2001
% adapted for use with ACCSPL3.M - 21 June 2001
% revised 22 June 2001

switch plotnum
case 1, h1 = h(1);,h2 = h(6);,h3 = h(7);,h4 = h(17);,h5 = h(20);
case 2, h1 = h(2);,h2 = h(8);,h3 = h(9);,h4 = h(23);,h5 = h(26);
case 3, h1 = h(3);,h2 = h(10);,h3 = h(11);,h4 = h(29);,h5 = h(32);
case 4, h1 = h(4);,h2 = h(12);,h3 = h(13);,h4 = h(35);,h5 = h(38);
case 5, h1 = h(5);,h2 = h(14);,h3 = h(15);,h4 = h(41);,h5 = h(44);
end
subplot(h2);                       % Make acceleration time plot active.
tlima = xlim;                      % Get the time limits on the acceleration plot.
alima = ylim;                      % Get the amplitude limits on the acceleration plot.
subplot(h3);                       % Make flow time plot active.
tlimf = xlim;                      % Get the time limits on the flow plot.
alimf = ylim;                      % Get the amplitude limits on the flow plot.
again = 1;                            % again = boolean deciding repeat or not
while again                           % As long as user chooses new time within the
  [x,y] = ginput(1);             % acceleration or flow time signal plots, then
  if ((x>tlima(1)) & x<tlima(2) & y>alima(1) & y<alima(2))   % do not repeat
    again = 0;                     % getting a point from mouse input.
  elseif ((x>tlimf(1)) & x<tlimf(2) & y>alimf(1) & y<alimf(2))
    again = 0;
  else
    h1 = errordlg('Choose a point in the acceleration or flow time signal plots.');
    uiwait(h1);
  end
end
wcn = round(x*fs);                    % new window center = time*fs
wla = 2048;                              % wla = acceleration window length
wa = zeros(1,wla);                    % Initialize the acceleration window vector.
wa(2:wla-1) = hanning(wla-2);      % Use a Hanning window for the acceleration
n1a = wcn - fix(wla./2);           % Calculate the start and end points for the
n2a = n1a + wla - 1;                  % acceleration FFT.
```

```
wlp = 2048;                              % wlp = microphone pressure window length
wp = hamming(wlp)';                      % Initialize the microphone pressure window
n1p = wcn - fix(wlp./2);                 % Calculate the start and end points for the
n2p = n1p + wlp - 1;                     % acoustic FFT.
n = 2048;                                % Get the FFT length from the GUI.
f = 0:fs/n:fs/2-fs/n;                    % f = frequencies in FFT
if length(get(h4,'String')) == 4         % If the ZOOM button in the GUI says "Zoom",
  subplot(h2);                           % then plot the entire acceleration and acoustic
  hold off
    plot(t,acc,'c-');                    % signals with the window centers marked by
    hold on                              % vertical lines.
    line([wcn/fs wcn/fs],[min(acc) max(acc)],'Color',[1 0 0],'LineWidth',2)
  axis([0 t(length(t)) min(acc) max(acc)])
  grid
  ylabel('cm/sec^2');
  subplot(h3);
  hold off
  plot(t,aco,'c-');
    hold on
    line([wcn/fs wcn/fs],[min(aco) max(aco)],'Color',[1 0 0],'LineWidth',2)
    axis([0 t(length(t)) min(aco) max(aco)])
  grid
  if plotnum == 5
    xlabel('Time (sec)');
  end
  ylabel('dyne/cm^2');
else                                     % Otherwise, the time plots should show the
  subplot(h2);                           % windowed portions of the acceleration and
  hold off
    plot(t(n1a:n2a),acc(n1a:n2a),'b-');  % acoustic signals with the window centers
    hold on                              % marked by vertical lines.
  line([wcn/fs wcn/fs],[min(acc(n1a:n2a)) max(acc(n1a:n2a))],'Color',[1 0 0],'LineWidth',2)
  plot(t(n1a:n2a),wa.*max(acc(n1a:n2a)),'r-')
  axis([t(n1a) t(n2a) min(acc(n1a:n2a)) max(acc(n1a:n2a))])
  grid
  ylabel('cm/sec^2');
  subplot(h3);
  hold off
  plot(t(n1p:n2p),aco(n1p:n2p),'c-');
    hold on
  line([wcn/fs wcn/fs],[min(aco(n1p:n2p)) max(aco(n1p:n2p))],'Color',[1 0 0],'LineWidth',2)
  plot(t(n1p:n2p),wp.*max(aco(n1p:n2p)),'r-')
    axis([t(n1p) t(n2p) min(aco(n1p:n2p)) max(aco(n1p:n2p))])
  grid
  if plotnum == 5
    xlabel('Time (sec)');
  end
  ylabel('dyne/cm^2');
end
subplot(h1);                             % And plot the acceleration spectrum
A = 2.*fft(acc(n1a:n2a).*wa,n)./n;       % using the windows created above.
plot(f,20.*log10(abs(A(1:n./2))));
grid
axis([0 2000 min(20.*log10(abs(A(1:205)))) max(20.*log10(abs(A(1:205))))+5])
if plotnum == 5
```

```
  xlabel('Frequency (Hz)')
end
ylabel('dB re 1 cm/sec^2')
title(fname);
dBSPL = 20.*log10(sqrt(mean(aco(n1p:n2p).^2))./2e-4);
set(h5,'String',strcat('dB SPL=',num2str(dBSPL,'%3.1f')));
```

```
function wcn = newtime6(acc,flo,fname,fs,h,t)

% Harold Cheyne
% 26 June 2001
% revised 11 July 2001

subplot(h(3));                          % Make acceleration time plot active.
tlima = xlim;                        % Get the time limits on the acceleration plot.
alima = ylim;                        % Get the amplitude limits on the acceleration plot.
subplot(h(4));                          % Make flow time plot active.
tlimf = xlim;                        % Get the time limits on the flow plot.
alimf = ylim;                        % Get the amplitude limits on the flow plot.
again = 1;                              % again = boolean deciding repeat or not
while again                             % As long as user chooses new time within the
  [x,y] = ginput(1);                 % acceleration or flow time signal plots, then
  if ((x>tlima(1)) & x<tlima(2) & y>alima(1) & y<alima(2))   % do not repeat
    again = 0;                       % getting a point from mouse input.
  elseif ((x>tlimf(1)) & x<tlimf(2) & y>alimf(1) & y<alimf(2))
    again = 0;
  else
    error('Choose a point within the acceleration or flow time signal plots.');
  end
end
wcn = round(x*fs);                   % new window center = time*fs
wla = str2double(get(h(21),'String'));   % wla = acceleration window length
wa = zeros(1,wla);                   % Initialize the acceleration window vector.
wa(1:wla-1) = hanning(wla-1,'periodic').*2;  % Use a Hanning window for the accel.
n1a = wcn - fix(wla./2);             % Calculate the start and end points for the
n2a = n1a + wla - 1;                    % acceleration FFT.
wlf = str2double(get(h(23),'String'));   % wlf = flow window length
wf = hamming(wlf)'.*1.8525;          % Define the flow window
n1f = wcn - fix(wlf./2);             % Calculate the start and end points for the
n2f = n1f + wlf - 1;                    % acoustic FFT.
n = 2048;                                  % FFT length = 2048.
f = 0:fs/n:fs/2-fs/n;                % f = frequencies in FFT
zinout = get(h(19),'String');        % Are the time plots zoomed in or out?
if length(zinout) == 4               % If the zoom button says "Zoom",
  subplot(h(3));                        % then plot the entire acoustic and inverted
  hold off                                 % acceleration signals with the window centers
  plot(t,-acc,'k-');                 % marked by vertical lines.
  hold on
  line([wcn/fs wcn/fs],[min(-acc) max(-acc)],'Color',[0 0 0],'LineWidth',2)
  axis([0 t(length(t)) min(-acc) max(-acc)])
  grid
  xlabel('Time (sec)');
  ylabel('-acc(t), cm/sec^2');
```

```
  subplot(h(4));
  hold off
  plot(t,flo,'k-');
  hold on
  line([wcn/fs wcn/fs],[min(flo) max(flo)],'Color',[0 0 0],'LineWidth',2)
  axis([0 t(length(t)) min(flo) max(flo)])
  grid
  xlabel('Time (sec)');
  ylabel('v_{vm}(t), cm^3/sec');
else                                        % If the ZOOM button says "UnZoom", then plot the
  subplot(h(3));                            % windowed portions of the acoustic and inverted
  hold off                                  % acceleration signals with the wondow centers
  plot(t(n1a:n2a),-acc(n1a:n2a),'k-');      % marked by vertical lines.
  hold on
  accmin = min(-acc(n1a:n2a));
  accmax = max(-acc(n1a:n2a));
  line([wcn/fs wcn/fs],[accmin accmax],'Color',[0 0 0],'LineWidth',2)
  plot(t(n1a:n2a),wa.*accmax./2,'k-')
  axis([t(n1a) t(n2a) accmin accmax])
  grid
  xlabel('Time (sec)');
  ylabel('-acc(t), cm/sec^2');
  subplot(h(4));
  hold off
  plot(t(n1f:n2f),flo(n1f:n2f),'k-');
  hold on
  flomin = min(flo(n1f:n2f));
  flomax = max(flo(n1f:n2f));
  line([wcn/fs wcn/fs],[flomin flomax],'Color',[0 0 0],'LineWidth',2)
  plot(t(n1f:n2f),wf.*flomax./1.8525,'k-')
  axis([t(n1f) t(n2f) flomin flomax])
  grid
  xlabel('Time (sec)');
  ylabel('v_{vm}(t), cm^3/sec');
end
subplot(h(1));                              % And plot the acceleration and flow
A = 2.*fft(acc(n1a:n2a).*wa,n)./n;  % spectra using the windows created above.
plot(f,20.*log10(abs(A(1:n./2))),'k-');
grid
axis([0 3000 min(20.*log10(abs(A(1:308)))) max(20.*log10(abs(A(1:308))))+5])
xlabel('Frequency (Hz)')
ylabel('|ACC|, dB re 1 cm/sec^2')
title(strcat(fname,',AC acceleration signal'));
subplot(h(2));
F = 2.*fft(flo(n1f:n2f).*wf,n)./n;
plot(f,20.*log10(abs(F(1:n./2))),'k-');
grid
axis([0 3000 min(20.*log10(abs(F(1:308)))) max(20.*log10(abs(F(1:308))))+5])
xlabel('Frequency (Hz)')
ylabel('|V_{vm}|, dB re 1 cm^3/sec')
title(strcat(fname,',airflow signal'));
```

```
function newwindow(acc,flo,fname,fs,h,t,wcn)
```

```
% Harold Cheyne
% 26 June 2001
% revised 12 July 2001

wla = str2double(get(h(21),'String'));      % wla = acceleration window length
wa = zeros(1,wla);                               % Initialize the acceleration window vector.
wa(1:wla-1) = hanning(wla-1,'periodic').*2;  % Use a Hanning window for the accel.
n1a = wcn - fix(wla./2);                    % Calculate the start and end points for the
n2a = n1a + wla - 1;                             % acceleration FFT.
wlf = str2double(get(h(23),'String'));      % wlf = flow window length
wf = hamming(wlf)'.*1.8525;                 % Define the flow window
n1f = wcn - fix(wlf./2);                    % Calculate the start and end points for the
n2f = n1f + wlf - 1;                             % acoustic FFT.
n = 2048;                                             % FFT length = 2048.
f = 0:fs/n:fs/2-fs/n;                       % f = frequencies in FFT
zinout = get(h(19),'String');               % Are the time plots zoomed in or out?
if length(zinout) == 4                      % If the zoom button says "Zoom",
  subplot(h(3));                                 % then plot the entire acoustic and inverted
  hold off                                            % acceleration signals with the window centers
  plot(t,-acc,'k-');                       % marked by vertical lines.
  hold on
  line([wcn/fs wcn/fs],[min(-acc) max(-acc)],'Color',[0 0 0],'LineWidth',2)
  axis([0 t(length(t)) min(-acc) max(-acc)])
  grid
  xlabel('Time (sec)');
  ylabel('-acc(t), cm/sec^2');
  subplot(h(4));
  hold off
  plot(t,flo,'k-');
  hold on
  line([wcn/fs wcn/fs],[min(flo) max(flo)],'Color',[0 0 0],'LineWidth',2)
  axis([0 t(length(t)) min(flo) max(flo)])
  grid
  xlabel('Time (sec)');
  ylabel('v_{vm}(t), cm^3/sec');
else                                                  % If the ZOOM button says "UnZoom", then plot the
  subplot(h(3));                                 % windowed portions of the acoustic and inverted
  hold off                                            % acceleration signals with the window centers
  plot(t(n1a:n2a),-acc(n1a:n2a),'k-');      % marked by vertical lines.
  hold on
  accmin = min(-acc(n1a:n2a));
  accmax = max(-acc(n1a:n2a));
  line([wcn/fs wcn/fs],[accmin accmax],'Color',[0 0 0],'LineWidth',2)
  plot(t(n1a:n2a),wa.*accmax./2,'k-')
  axis([t(n1a) t(n2a) accmin accmax])
  grid
  xlabel('Time (sec)');
  ylabel('-acc(t), cm/sec^2');
  subplot(h(4));
  hold off
  plot(t(n1f:n2f),flo(n1f:n2f),'k-');
  hold on
  flomin = min(flo(n1f:n2f));
  flomax = max(flo(n1f:n2f));
```

```
  line([wcn/fs wcn/fs],[flomin flomax],'Color',[0 0 0],'LineWidth',2)
  plot(t(n1f:n2f),wf.*flomax./1.8525,'k-')
  axis([t(n1f) t(n2f) flomin flomax])
  grid
  xlabel('Time (sec)');
  ylabel('v_{vm}(t), cm^3/sec');
end
subplot(h(1));                                    % And plot the acceleration and flow
A = 2.*fft(acc(n1a:n2a).*wa,n)./n;  % spectra using the windows created above.
plot(f,20.*log10(abs(A(1:n./2))),'k-');
grid
axis([0 3000 min(20.*log10(abs(A(1:308)))) max(20.*log10(abs(A(1:308))))+5])
xlabel('Frequency (Hz)')
ylabel('|ACC|, dB re 1 cm/sec^2')
title(strcat(fname,',AC acceleration signal'));
subplot(h(2));
F = 2.*fft(flo(n1f:n2f).*wf,n)./n;
plot(f,20.*log10(abs(F(1:n./2))),'k-');
grid
axis([0 3000 min(20.*log10(abs(F(1:308)))) max(20.*log10(abs(F(1:308))))+5])
xlabel('Frequency (Hz)')
ylabel('|V_{vm}|, dB re 1 cm^3/sec')
title(strcat(fname,',airflow signal'));
```

```
function [mk,tt] = pkpsd(xx,ppk,thrd)
% positive-peak based threshold detection algorithm
% usage [mk,tt] = pkpsd(xx,ppk,thrd)
%    mk   --- threshold marks
%    tt   --- mark times
%    xx   --- input signal
%    ppk  --- positive peak locations
%    thrd --- thresholds (0-1)
% modified as noted below by Harold Cheyne, 28 June 2001.

indx = 1;                                   % Initialize the index counter.
P = length(ppk);                        % P = # of positive peaks
for k=1:P                                   % Loop through the positive peaks
  if(k==1)                                  % For the first peak location, set
    lbd=1;                                % the lower bound to one, and the
    ubd = ppk(2);                       % upper bound = peak2.
  elseif(k==P)                            % For the last peak location, set
    lbd = ppk(P-1);                     % the lower bound to the previous peak,
    ubd = length(xx);                  % and the upper bound to the input length.
  else                                        % For all of the other peak locations, set
    lbd = ppk(k-1);                     % the lower bound to the previous peak, and
    ubd = ppk(k+1);                    % the upper bound to the following peak.
  end;
  % mn is the minimum of the input signal between the lower and upper bounds
  mn = min(xx(lbd:ubd));

  % Search for a local minimum on the left side, for all peaks except the first.
  if k ~= 1                                % modification by Harold Cheyne
    i = 1;                                  % Initialize the counter
```

```
    while ((ppk(k)-i)>0) & ...   % While (peak location - counter) is positive and
        (((xx(ppk(k)-i)<=xx(ppk(k)-i+1))|(xx(ppk(k)-i)>(xx(ppk(k))-mn)/3+mn)))
      i = i+1; % [the input signal has + slope at the peak minus the counter, or
    end;            % the input signal at the peak minus the counter is larger than 1/3
    locnt = ppk(k)-i;   % of the AC excursion plus the minimum], increment the counter.
  end    % Once the loop is exited, save the location of the left minima in locnt.

  % Search for a local minimum on the right side, for all peaks except the last.
  if k ~= P          % modification by Harold Cheyne
    i = 1;           % Initialize the counter, and while the peak location plus the
    while ((ppk(k)+i)<=length(xx)) & ... % counter is less than the input length, and
        (((xx(ppk(k)+i)<=xx(ppk(k)+i-1))|(xx(ppk(k)+i)>(xx(ppk(k))-mn)/3+mn)))
      i=i+1;  % [the input signal has - slope at the peak plus the counter, or
    end;            % the input at the peak plus the counter is greater than 1/3 the
    hicnt = ppk(k)+i;  % AC excursion plus the minimum], increment the counter.
  end    % Once the loop is exited, save the location of the right minima in hicnt.

  % Set the lower limit for marks and the reference
  if k == 1
    lw = xx(hicnt);
  elseif k == P
    lw = xx(locnt);
  else lw = max([xx(locnt) xx(hicnt)]);  % lw = greater of left or right minimum
  end

  % ref = input at current peak minus lw, times the threshold, plus lw.
  ref = (xx(ppk(k))-lw)*thrd+lw;              % ref = (AC excursion)*threshold+lw

  % Find threshold on the left side, for all peaks except the first.
  if k ~= 1                                              % modification by Harold Cheyne
    for i=locnt:ppk(k)                        % From the left minimum to the peak,
      if (prod(xx(i:i+1)-[ref ref])<=0)% if the current and the next input value
        mk(indx) = i;                         % are around the value ref, then set the
        tt(indx) = i+(ref-xx(i))/(xx(i+1)-xx(i));    % threshold mark to the current
        indx = indx + 1;                      % index, set the threshold time, and
      end;                                               % increment the index counter.
    end;
  end

  % Find threshold on the right side, for all peaks except the last.
  if k ~= P                                              % modification by Harold Cheyne
    for i=ppk(k):hicnt                        % From the peak to the right minimum,
      if (prod(xx(i:i+1)-[ref ref])<=0)% if the current and the next input value
        mk(indx) = i+1;                       % are around the value ref, then set the
        tt(indx) = i+(ref-xx(i))/(xx(i+1)-xx(i));    % threshold mark to the next
        indx = indx + 1;                      % index, set the threshold time, and
      end;                                               % increment the index counter.
    end;
  end
end;
```

```
function x = rad(f,r)
```

```
% This MATLAB function computes the magnitude of the radiation
% characteristic for the input frequency f (in Hz).  It
% uses the simple source model for the calculation, or
% R = (f*rho)/(2*r), where f is the frequency, rho is the density
% of air, and r is the distance from the source.
%
% USAGE:   x = rad(f,r)
% Input:   f = frequency or frequencies for calculation (Hz)
%          r = distance from source (cm)
%
% Harold Cheyne, 14 October 1999

rho = 0.00118;                          % rho = density of air at 22°C, in g/cm^3
y = f.*rho./(2.*r);
x = y';
```

```
function [pz,zz] = s2z(p,pbw,z,zbw,fs)

% Function S2Z.M
%
% USAGE:   [pz,zz] = s2z(p,pbw,z,zbw,fs)
%
% Input:   p = column vector of the system pole frequencies (Hz)
%          pbw = column vector of the system pole bandwidths (Hz)
%          z = column vector of the system zero frequencies (Hz)
%          Zbw = column vector of the system zero bandwidths (Hz)
%          fs = sampling frequency (Hz)
%
% Output:  pz = corresponding z-domain poles
%          zz = corresponding z-domain zeros
%
% This MATLAB function converts s-domain system poles and zeros (in
% Hertz not radians/sec) to z-domain poles and zeros.
%
% Harold Cheyne
% adapted from vtractz.m, 7 December 2000
% revised 18 April 2001

% The z-domain poles are of the form a+jb, where a and b are between 0 and 1.
% And each pole/zero needs to appear in a pair, a+jb and a-jb.
np = length(p);                          % Get the number of poles.
nz = length(z);                          % Get the number of zeros.
pz = zeros(2.*np,1);                     % Initialize the poles output vector.
zz = zeros(2.*nz,1);                     % Initialize the zeros output vector.
Bp = pi.*pbw./fs;                        % Express the pole bandwidths in radians.
Bz = pi.*zbw./fs;                        % Express the zero bandwidths in radians.
Fp = 2.*pi.*p./fs;                       % Express the pole frequencies in radians.
Fz = 2.*pi.*z./fs;                       % Express the zero frequencies in radians.
rp = -cos(Bp)+2-sqrt((cos(Bp)-1).*(cos(Bp)-3));  % Calculate z-plane pole magnitudes.
rz = -cos(Bz)+2-sqrt((cos(Bz)-1).*(cos(Bz)-3));  % Calculate z-plane zero magnitudes.
for m = 1:np,                            % For each input pole frequency, calculate
  if p(m) < fs/4                         % the real part, which is positive for poles
    a = rp(m)./sqrt(1+tan(Fp(m)).^2);    % in the 1st quadrant and negative for poles
```

```
  else a = -rp(m)./sqrt(1+tan(Fp(m)).^2);% in the 2nd quadrant.
  end
  b = a.*tan(Fp(m));                          % Calculate the imaginary part.
  pz(2*m-1) = a + j.*b;
  pz(2*m) = a - j.*b;
end
for m = 1:nz,                                 % For each input zero frequency, calculate
  if z(m) < fs/4                              % the real part, which is positive for zeros
    a = rz(m)./sqrt(1+tan(Fz(m)).^2);    % in the 1st quadrant and negative for zeros
  else a = -rz(m)./sqrt(1+tan(Fz(m)).^2);% in the 2nd quadrant.
  end
  b = a.*tan(Fz(m));                          % Calculate the imaginary part.
  zz(2*m-1) = a + j.*b;
  zz(2*m) = a - j.*b;
end
```

```
function setF1F2F3(h)

% Function SETF1F2F3.M
%
% Usage:  setF1F2F3(h)
% Input:  h, the graphics handles of the accflo3.m GUI
% Output: The update of the F1, F2, and F3 frequencies
%         displayed in the GUI based on the input.
%
% Harold Cheyne
% 15 May 2001
% adapted from SetF3orL.M 31 May 2001

L = str2double(get(h(56),'String'));
set(h(49),'String',num2str(35400./(4.*L),'%4.0f'));
set(h(51),'String',num2str(3.*35400./(4.*L),'%4.0f'));
set(h(53),'String',num2str(5.*35400./(4.*L),'%4.0f'));
```

```
function setF3orL(F3,L,h)

% Function SETF3ORL.M
%
% Usage:  setF3orL(F3,L,h)
% Input:  F3, the frequency of the third formant (Hz)
%         L, the length of the vocal tract (cm)
%         h, the graphics handles of the accflo3.m GUI
% Output: The update of the F3 frequency or vocal tract length
%         display in the accflo3.m GUI based on the input.
%         The purpose of this function is to lock the values
%         of F3 and L together, so that changing one in the
%         GUI results in the other also changing accordingly.
%
% Harold Cheyne
% 15 May 2001

if isempty(F3)
```

```
  set(h(27),'String',num2str(5.*35400./(4.*L),'%4.0f'));
elseif isempty(L)
  set(h(29),'String',num2str(5.*35400./(4.*F3),'%2.1f'));
end
```

```
function update(f,fmax,fs,h,n,t,wn,wcn,z)

% This function is used exclusively by the MATLAB function LPLV.M,
% in response to the user activating the "Update acoustic signal"
% button.
%
% Harold Cheyne
% 5 February 2001

n1 = wcn - wn/2 + 1;                         % n1 = window start point
n2 = n1 + wn - 1;                            % n2 = window end point
t1 = n1-wn*2;
t2 = n2+wn*2;

subplot(h(3));                                   % Select acoustic time signal plot
hold off
plot(t(t1:t2),z(t1:t2),'b'); % Re-plot acoustic time signal,
hold on
line([wcn/fs wcn/fs],[min(z(t1:t2)) max(z(t1:t2))],'Color',[1 0 0],'LineWidth',2);
axis([t(t1) t(t2) min(z(t1:t2)) max(z(t1:t2))]);
grid
xlabel('Time (sec)');
ylabel('dyne/cm^2');
title('acoustic signal');

subplot(h(5));                                   % Select the LPC spectrum plot.
a = autolpc(z(wcn-n/2+1:wcn+n/2),20);      % Perform LPC on a n-point window around wcn
az = ones(21,1);                             % Create a vector for holding coefficients
az(2:21) = -a;                                   % and put the LPC coefficients in it for
LPC = freqz(1,az,f,fs);                      % use with the function FREQZ.M.
plot(f(2:fmax),20.*log10(abs(LPC(2:fmax))));   % Plot the LPC spectrum.
axis([f(2) f(fmax) min(20.*log10(abs(LPC(2:fmax))))-5 max(20.*log10(abs(LPC(2:fmax))))+5]);
grid
ylabel('dB');
xlabel('Frequency (Hz)');
title('LPC spectrum of acoustic signal');
flpc = sort(angle(roots(az)).*fs./(2.*pi));% Get the pole frequencies in Hz.
posflpc = find(flpc>0);                          % Then find the positive ones, and
set(h(25),'String',strcat('F1=',num2str(flpc(posflpc(1)),'%3.0f')));   % print them in
set(h(26),'String',strcat('F2=',num2str(flpc(posflpc(2)),'%4.0f')));   % the F1, F2, & F3
set(h(27),'String',strcat('F3=',num2str(flpc(posflpc(3)),'%4.0f')));   % text boxes.
```

```
function zp = vtractz(p,bw,fs)

% This MATLAB function produces a vector of z-domain poles corresponding
% to the input pole frequencies and bandwidths, assuming an all-pole
% model of the vocal tract.  It takes vector input of pole frequencies
```

```
% and bandwidths (in Hz), and a scalar sampling frequency fs, and
% outputs a vector containing the z-domain poles for the vocal tract.
%
% USAGE:   zp = vtractz(p,bw,fs)
%
% Input:   p = column vector of the system pole frequencies (Hz)
%          bw = column vector of the system pole bandwidths (Hz)
%          fs = sampling frequency (Hz)
%
% Output:  zp = vocal tract model z-domain poles
%
% Harold Cheyne
% 7 December 2000
% revised 14 December 2000

% The z-domain poles are of the form a+jb, where a and b are between 0 and 1.
% And each pole needs to appear in a pair, a+jb and a-jb.
n = length(p);                                  % get the number of poles
zp = zeros(2.*n,1);                             % initialize the output vector
B = pi.*bw./fs;                                 % Express the bandwidths in radians
P = 2.*pi.*p./fs;                               % Express the pole frequencies in radians
r = -cos(B)+2-sqrt((cos(B)-1).*(cos(B)-3));  % Calculate the z-plane pole magnitudes
for m = 1:n,                                    % For each input pole frequency, calculate
  if p(m) < fs/4                                % the real part, which is positive for poles
    a = r(m)./sqrt(1+tan(P(m)).^2);          % in the 1st quadrant and negative for poles
  else a = -r(m)./sqrt(1+tan(P(m)).^2);      % in the 2nd quadrant.
  end
  b = a.*tan(P(m));                             % Calculate the imaginary part.
  zp(2*m-1) = a + j.*b;
  zp(2*m) = a - j.*b;
end
```

```
function P=vvg2mic(fs,h,Hvt,vvg)

% Function VVG2MIC.M - Convert glottal volume velocity to pressure at a microphone
%                      at some distance from the mouth
%
% Usage:  p=vvg2mic(fs,h,Hvt,vvg);
% Input:  fs, the sampling frequency of the signals
%         h, the graphics handles of the active GUI
%         Hvt, the transfer function of the vocal tract
%         vvg, the estimated glottal volume velocity spectrum
% Output: P, the spectral magnitude of the microphone pressure
%           calculated to exist at a distance away from the mouth
%
% Harold Cheyne
% 30 May 2001

Vvm = abs(Hvt).*abs(vvg);                       % Get the mouth volume velocity magnitude
n = str2double(get(h(20),'String'));
f = 0:fs/n:fs/2-fs/n;
P = rad(f,15)'.*Vvm;
subplot(h(7))
```

```
plot(f,20.*log10(P),'k-')
y = ylim(h(2));
axis([0 3000 y(1) y(2)])
grid
xlabel('Frequency (Hz)');
ylabel('dB re 1 dyne/cm^2');
title('Acceleration-derived microphone pressure')
```

```
function in=zoominout3(fname,fs,h,in,t,wcn,x,z)

% Function ZOOMINOUT3.M
%
% Usage:  in=zoominout(fname,fs,h,in,t,wcn,w,x,z)
% Input:  fname, the filename being processed by ACCFLO.M
%         fs, the sampling frequency (Hz) of the signals being processed by ACCFLO.M
%         h, the handles of all of the graphics objects created by ACCFLO.M
%         in, the state of the ZOOM button (0=zoom out, 1=zoom in)
%         t, the time indeces for the signals being processed by ACCFLO.M
%         wcn, the center index of the window to be used on the acceleration signal
%         x, the acceleration signal
%         z, the flow signal
% Output: in (see above)
%
% Harold Cheyne
% 3 April 2001
% revised 14 May 2001
% revised 28 August 2001 to produce plots for thesis document (i.e., black & white)
% revised 17 September 2001 to correct for the window gain problem (see accflo3)

if in                                   % If the signals are already zoomed in, then
  subplot(h(3));                        % replot the entire acceleration time signal
  hold off                              % along with the window center time marker.
  plot(t,x,'k-');
%  plot(t,x,'c-');
  hold on
    line([wcn/fs wcn/fs],[min(x) max(x)],'Color',[0 0 0],'LineWidth',2);
%   line([wcn/fs wcn/fs],[min(x) max(x)],'Color',[1 0 0],'LineWidth',2);
  grid
  xlabel('Time (sec)');
  ylabel('a(t)=acc(t)-mean(acc(t)) (cm/sec^2)');
  axis([0 t(length(t)) min(x) max(x)])
  subplot(h(4));                        % And replot the entire flow time signal along
  hold off                              % with the window center time marker.
  plot(t,z,'k-');
%  plot(t,z,'c-');
    hold on
    line([wcn/fs wcn/fs],[min(z) max(z)],'Color',[0 0 0],'LineWidth',2);
%   line([wcn/fs wcn/fs],[min(z) max(z)],'Color',[1 0 0],'LineWidth',2);
  grid
  xlabel('Time (sec)');
  ylabel('cm^3/sec');
  axis([0 t(length(t)) min(z) max(z)])
  set(h(12),'String','Zoom');           % And change the zoom button to "Zoom in"
```

```
else                                            % Otherwise if the entire signals are showing,
  wn = str2double(get(h(14),'String'));  % get the window length from the GUI,
  w = zeros(1,wn);                             % initialize the acceleration window vector,
  w(1:wn-1) = hanning(wn-1,'periodic').*2;% & define a Hanning window for the spectra.
  wnf = str2double(get(h(50),'String'));  % Then get the flow window length from the
  wf = hamming(wnf)'.*1.8525;              % GUI and make a Hamming window for the LPC.
  subplot(h(3));                                % then replot a section of each signal that is
  hold off                                          % twice as long as the window, with the window
  t1 = wcn - fix(wn/2);                       % t1 & t2 are the start & end points of the
  t2 = t1 + wn - 1;                              % "zoomed" view of the acceleration.
  plot(t(t1:t2),x(t1:t2),'k');             % centered in the view.
%  plot(t(t1:t2),x(t1:t2),'b');                % centered in the view.
    hold on
  plot(t(t1:t2),w.*max(x(t1:t2))./2,'k');
%  plot(t(t1:t2),w.*max(x(t1:t2)),'r');
  line([wcn/fs wcn/fs],[min(x(t1:t2)) max(x(t1:t2))],'Color',[0 0 0],'LineWidth',2);
%  line([wcn/fs wcn/fs],[min(x(t1:t2)) max(x(t1:t2))],'Color',[1 0 0],'LineWidth',2);
  axis([t(t1) t(t2) min(x(t1:t2)) max(x(t1:t2))])
    grid
  xlabel('Time (sec)');
  ylabel('a(t) (cm/sec^2)');
  subplot(h(4));
  hold off
  t3 = wcn - fix(wnf/2);
  t4 = t3 + wnf - 1;
  plot(t(t3:t4),z(t3:t4),'k');
%  plot(t(t3:t4),z(t3:t4),'b');
  hold on
  plot(t(t3:t4),wf.*max(z(t3:t4))./1.8525,'k')
%  plot(t(t3:t4),wf.*max(z(t3:t4)),'r')
    line([wcn/fs wcn/fs],[min(z(t3:t4)) max(z(t3:t4))],'Color',[0 0 0],'LineWidth',2);
%   line([wcn/fs wcn/fs],[min(z(t3:t4)) max(z(t3:t4))],'Color',[1 0 0],'LineWidth',2);
    axis([t(t3) t(t4) min(z(t3:t4)) max(z(t3:t4))])
    grid
  xlabel('Time (sec)');
  ylabel('cm^3/sec');
  set(h(12),'String','unZoom');
end
in = ~in;                                       % toggle the state of in
```

```
function zoominout4(acc,aco,fname,fs,h,t,wcn)

% Function ZOOMINOUT4.M
%
% Usage:  zoominout4(fname,fs,h,t,wcn,w,x,z)
% Input:  acc, the acceleration time signal,
%         aco, the acoustic time signal,
%         fname, the filename being processed by ACCFLO.M
%         fs, the sampling frequency (Hz) of the signals being processed by ACCFLO.M
%         h, the handles of all of the graphics objects created by ACCFLO.M
%         t, the time indeces for the signals being processed by ACCFLO.M
%         wcn, the center index of the window to be used on the acceleration signal
% Output: zoominout4.m alters the GUI to show either the entire acceleration and
```

```
%         acoustic time signals, or just the windowed portions of them.
%
% Harold Cheyne
% 3 April 2001
% revised 14 May 2001
% adapted for use with ACCSPL1.M 31 May 2001
% revised 1 June 2001

wla = str2double(get(h(16),'String'));    % wla = acceleration window length
wa = zeros(1,wla);                            % Initialize the acceleration window vector.
wa(1:wla-1) = hanning(wla-1,'periodic').*2; % Use a Hanning window for the accel.
n1a = wcn - fix(wla./2);                    % Calculate the start and end points for the
n2a = n1a + wla - 1;                          % acceleration FFT.
wlp = str2double(get(h(18),'String'));    % wlp = microphone pressure window length
wp = hamming(wlp)'.*1.8525;                 % Initialize the microphone pressure window
n1p = wcn - fix(wlp./2);                    % Calculate the start and end points for the
n2p = n1p + wlp - 1;                          % acoustic FFT.
n = str2double(get(h(20),'String')); % Get the FFT length from the GUI.
f = 0:fs/n:fs/2-fs/n;                        % f = frequencies in FFT
if length(get(h(11),'String')) == 6       % If the ZOOM button in the GUI says "unZoom",
  subplot(h(3));                              % then plot the entire acceleration and acoustic
  hold off
    plot(t,acc,'k-');                         % signals with the window centers marked by
    hold on                                     % vertical lines.
    line([wcn/fs wcn/fs],[min(acc) max(acc)],'Color',[0 0 0],'LineWidth',2)
  axis([0 t(length(t)) min(acc) max(acc)])
  grid
    xlabel('Time (sec)');
  ylabel('acc(t), cm/sec^2');
  subplot(h(4));
  hold off
  plot(t,aco,'k-');
    hold on
    line([wcn/fs wcn/fs],[min(aco) max(aco)],'Color',[0 0 0],'LineWidth',2)
    axis([0 t(length(t)) min(aco) max(aco)])
    grid
    xlabel('Time (sec)');
  ylabel('p_{mic}, dyne/cm^2');
  set(h(11),'String','Zoom');
else                                            % Otherwise, the time plots should show the
  subplot(h(3));                              % windowed portions of the acceleration and
  hold off
  plot(t(n1a:n2a),acc(n1a:n2a),'k-'); % acoustic signals with the window centers
  grid
    hold on                                     % marked by vertical lines.
  line([wcn/fs wcn/fs],[min(acc(n1a:n2a)) max(acc(n1a:n2a))],'Color',[0 0 0],'LineWidth',2)
  plot(t(n1a:n2a),wa.*max(acc(n1a:n2a))./2,'k-')
  axis([t(n1a) t(n2a) min(acc(n1a:n2a)) max(acc(n1a:n2a))])
  xlabel('Time (sec)');
  ylabel('acc(t), cm/sec^2');
  subplot(h(4));
  hold off
  plot(t(n1p:n2p),aco(n1p:n2p),'k-');
  grid
  hold on
```

```
  line([wcn/fs wcn/fs],[min(aco(n1p:n2p)) max(aco(n1p:n2p))],'Color',[0 0 0],'LineWidth',2)
  plot(t(n1p:n2p),wp.*max(aco(n1p:n2p))./1.8525,'k-')
     axis([t(n1p) t(n2p) min(aco(n1p:n2p)) max(aco(n1p:n2p))])
     xlabel('Time (sec)');
  ylabel('p_{mic}, dyne/cm^2');
  set(h(11),'String','unZoom');
end
```

```
function zoominout5(acc,aco,fname,fs,h,plotnum,t,wcn)

% Function ZOOMINOUT5.M
%
% Usage:  zoominout5(acc,aco,fname,fs,h,plotnum,t,wcn)
% Input:  acc, the acceleration time signal,
%         aco, the acoustic time signal,
%         fname, the filename being processed by ACCSPL3.M
%         fs, the sampling frequency (Hz) of the signals being processed by ACCSPL3.M
%         h, the handles of the ACCSPL3.M GUI
%         plotnum, a number corresponding to which set of plots to operate on (1-5)
%         t, the time indeces for the signals being processed by ACCSPL3.M
%         wcn, the center index of the window to be used on the acceleration signal
% Output: zoominout5.m alters the GUI to show either the entire acceleration and
%           acoustic time signals, or just the windowed portions of them.
%
% Harold Cheyne
% 3 April 2001
% revised 14 May 2001
% adapted for use with ACCSPL3.M 21 June 2001
% revised

switch plotnum
case 1, h2 = h(6);,h3 = h(7);,h4 = h(17);
case 2, h2 = h(8);,h3 = h(9);,h4 = h(23);
case 3, h2 = h(10);,h3 = h(11);,h4 = h(29);
case 4, h2 = h(12);,h3 = h(13);,h4 = h(35);
case 5, h2 = h(14);,h3 = h(15);,h4 = h(41);
end
wla = 2048;                                  % wla = acceleration window length
wa = zeros(1,wla);                         % Initialize the acceleration window vector.
wa(2:wla-1) = hanning(wla-2);          % Use a Hanning window for the acceleration
n1a = wcn - fix(wla./2);                 % Calculate the start and end points for the
n2a = n1a + wla - 1;                        % acceleration FFT.
wlp = 2048;                                  % wlp = microphone pressure window length
wp = hamming(wlp)';                        % Initialize the microphone pressure window
n1p = wcn - fix(wlp./2);                 % Calculate the start and end points for the
n2p = n1p + wlp - 1;                        % acoustic FFT.
n = 2048;                                        % Get the FFT length from the GUI.
f = 0:fs/n:fs/2-fs/n;                      % f = frequencies in FFT
if length(get(h4,'String')) == 6      % If the ZOOM button in the GUI says "unZoom",
  subplot(h2);                                % then plot the entire acceleration and acoustic
  hold off
     plot(t,acc,'c-');                          % signals with the window centers marked by
     hold on                                        % vertical lines.
```

```
    line([wcn/fs wcn/fs],[min(acc) max(acc)],'Color',[1 0 0],'LineWidth',2)
  axis([0 t(length(t)) min(acc) max(acc)])
  grid
  ylabel('cm/sec^2');
  subplot(h3);
  hold off
  plot(t,aco,'c-');
    hold on
    line([wcn/fs wcn/fs],[min(aco) max(aco)],'Color',[1 0 0],'LineWidth',2)
    axis([0 t(length(t)) min(aco) max(aco)])
  grid
  if plotnum == 5
   xlabel('Time (sec)');
  end
  ylabel('dyne/cm^2');
  set(h4,'String','Zoom');
else                                        % Otherwise, the time plots should show the
  subplot(h2);                              % windowed portions of the acceleration and
  hold off
  plot(t(n1a:n2a),acc(n1a:n2a),'b-'); % acoustic signals with the window centers
  grid
    hold on                                 % marked by vertical lines.
  line([wcn/fs wcn/fs],[min(acc(n1a:n2a)) max(acc(n1a:n2a))],'Color',[1 0 0],'LineWidth',2)
  plot(t(n1a:n2a),wa.*max(acc(n1a:n2a)),'r-')
  axis([t(n1a) t(n2a) min(acc(n1a:n2a)) max(acc(n1a:n2a))])
  ylabel('cm/sec^2');
  subplot(h3);
  hold off
  plot(t(n1p:n2p),aco(n1p:n2p),'c-');
  grid
  hold on
  line([wcn/fs wcn/fs],[min(aco(n1p:n2p)) max(aco(n1p:n2p))],'Color',[1 0 0],'LineWidth',2)
  plot(t(n1p:n2p),wp.*max(aco(n1p:n2p)),'r-')
  axis([t(n1p) t(n2p) min(aco(n1p:n2p)) max(aco(n1p:n2p))])
  if plotnum == 5
   xlabel('Time (sec)');
  end
  ylabel('dyne/cm^2');
  set(h4,'String','unZoom');
end
```

```
function zoominout6(acc,flo,fname,fs,h,t,wcn)

% Function ZOOMINOUT6.M
%
% Usage:  zoominout6(acc,flo,fname,fs,h,t,wcn)
% Input:  acc, the acceleration time signal,
%         flo, the airflow time signal,
%         fname, the filename being processed by ACCFLO4.M
%         fs, the sampling frequency (Hz) of the signals being processed by ACCFLO4.M
%         h, the handles of all of the graphics objects created by ACCFLO4.M
%         t, the time indeces for the signals being processed by ACCFLO4.M
%         wcn, the center index of the window to be used on the acceleration signal
```

```
% Output: zoominout6.m alters the GUI to show either the entire acceleration and
%           airflow time signals, or just the windowed portions of them.
%
% Harold Cheyne
% 26 June 2001
% revised 12 July 2001

wla = str2double(get(h(21),'String'));     % wla = acceleration window length
wa = zeros(1,wla);                              % Initialize the acceleration window vector.
wa(1:wla-1) = hanning(wla-1,'periodic').*2; % Use a Hanning window for the accel.
n1a = wcn - fix(wla./2);                    % Calculate the start and end points for the
n2a = n1a + wla - 1;                            % acceleration FFT.
wlf = str2double(get(h(23),'String'));     % wlf = flow window length
wf = hamming(wlf)'.*1.8525;                 % Define the flow window
n1f = wcn - fix(wlf./2);                    % Calculate the start and end points for the
n2f = n1f + wlf - 1;                            % acoustic FFT.
n = 2048;                                           % FFT length = 2048.
f = 0:fs/n:fs/2-fs/n;                       % f = frequencies in FFT
if length(get(h(19),'String')) == 6    % If the ZOOM button in the GUI says "unZoom",
  subplot(h(3));                                % then plot the entire acoustic and inverted
  hold off                                          % acceleration signals with the window centers
  plot(t,-acc,'k-');                        % marked by vertical lines.
  hold on
  line([wcn/fs wcn/fs],[min(-acc) max(-acc)],'Color',[0 0 0],'LineWidth',2)
  axis([0 t(length(t)) min(-acc) max(-acc)])
  grid
  xlabel('Time (sec)');
  ylabel('-acc(t), cm/sec^2');
  subplot(h(4));
  hold off
  plot(t,flo,'k-');
  hold on
  line([wcn/fs wcn/fs],[min(flo) max(flo)],'Color',[0 0 0],'LineWidth',2)
  axis([0 t(length(t)) min(flo) max(flo)])
  grid
  xlabel('Time (sec)');
  ylabel('v_{vm}(t), cm^3/sec');
  set(h(19),'String','Zoom');
else                                                    % Otherwise, the time plots should show the
  subplot(h(3));                                % windowed portions of the acoustic and inverted
  hold off                                          % acceleration signals with the window centers
  plot(t(n1a:n2a),-acc(n1a:n2a),'k-');      % marked by vertical lines.
  hold on
  accmin = min(-acc(n1a:n2a));
  accmax = max(-acc(n1a:n2a));
  line([wcn/fs wcn/fs],[accmin accmax],'Color',[0 0 0],'LineWidth',2)
  plot(t(n1a:n2a),wa.*accmax./2,'k-')
  axis([t(n1a) t(n2a) accmin accmax])
  grid
  xlabel('Time (sec)');
  ylabel('-acc(t), cm/sec^2');
  subplot(h(4));
  hold off
  plot(t(n1f:n2f),flo(n1f:n2f),'k-');
  hold on
```

```
  flomin = min(flo(n1f:n2f));
  flomax = max(flo(n1f:n2f));
  line([wcn/fs wcn/fs],[flomin flomax],'Color',[0 0 0],'LineWidth',2)
  plot(t(n1f:n2f),wf.*flomax./1.8525,'k-')
  axis([t(n1f) t(n2f) flomin flomax])
  grid
  xlabel('Time (sec)');
  ylabel('v_{vm}(t), cm^3/sec');
  set(h(19),'String','unZoom');
end
```

12 Bibliography

Buekers R, Bierens E, Kingma H, Marres EHMA (1995). Vocal load as measured by the voice accumulator. *Folia Phoniatrica et Logopaedica* 47:252-261.

Childers, DG, Hicks DM, Moore GP, Alsaka YA (1986). A model for vocal fold vibratory motion, contact area, and the electroglottogram. *Journal of the Acoustical Society of America* 80(5):1309-1320.

Cranen B, Boves L (1987). On subglottal formant analysis. *Journal of the Acoustical Society of America* 81(3): 734-746.

Elliot N, Sundberg J, Gramming P (1992). Physiological Aspects of a Vocal Exercise. *Speech Transmission Laboratory Quarterly Progress and Status Report 2(3)*, Royal Institute of Technology, Stockholm, 79-87.

Ewan WG, Krones R (1974). Measuring larynx movement using the thyroumbrometer. *Journal of Phonetics* 2: 327-335.

Fant G (1960). *Acoustic Theory of Speech Production.* The Hague, The Netherlands: Mouton & Co.

Fant G (1962). Formant bandwidth data. *Speech Transmission Laboratory Quarterly Progress and Status Report 1*, Royal Institute of Technology, Stockholm, 1-3.

Fant G, Ishizaka K, Lindqvist J, Sundberg J (1972). Subglottal Formants. *Speech Transmission Laboratory Quarterly Progress and Status Report 1*, Royal Institute of Technology, Stockholm, 1-12.

Fant G, Liljencrants J, Lin Q (1985). A Four-Parameter Model of Glottal Flow. *Speech Transmission Laboratory Quarterly Progress and Status Report 4*, Royal Institute of Technology, Stockholm, 1-13.

Fant G, Lin Q (1988). Frequency Domain Interpretation And Derivation Of Glottal Flow Parameters. *Speech Transmission Laboratory Quarterly Progress and Status Report 2-3*, Royal Institute of Technology, Stockholm, 1-21.

Fredberg JJ, Wohl MEB, Glass GM, Dorkin HL (1980). Airway area by acoustic reflections measured at the mouth. *Journal of Applied Physiology: Respiratory, Environmental and Exercise Physiology* 48(5): 749-758.

Fredberg JJ, Hoenig A (1978). Mechanical Response of the Lungs at High Frequencies. *Journal of Biomedical Engineering* 100: 57-66.

Gold B, Rabiner LR (1968). Analysis of Digital and Analog Formant Synthesizers. *IEEE Transactions on Audio and Electroacoustics*, Vol. AU-16: 81-94.

Habib RH, Chalker RB, Suki B, Jackson AC (1994). Airway geometry and wall mechanical properties estimated from subglottal input impedance in humans. *Journal of Applied Physiology* 77:441-451.

Hanson HM, Chuang ES (1999). Glottal characteristics of male speakers: Acoustic correlates and comparison with female data. *Journal of the Acoustical Society of America* 106(2): 1064-1077.

Hanson HM (1997). Glottal characteristics of female speakers: Acoustic correlates. *Journal of the Acoustical Society of America* 101(1): 466-481.

Hillman RE, Montgomery WW, Zeitels SM (1997). Appropriate use of objective measures of vocal function in the multidisciplinary management of voice disorders. *Current Opinion in Otolaryngology and Head & Neck Surgery* 5:172-175.

Hillman RE, Holmberg EB, Perkell JS, Walsh M, Vaughan C (1989). Objective assessment of vocal hyperfunction: an experimental framework and initial results. *Journal of Speech and Hearing Research* 32(3):373-392.

Hixon TJ (1973). Respiratory function in speech. In F. Minifie, T. Hixon, & F. Williams (Eds.), *Normal aspects of speech, hearing, and language.* Englewood Cliffs, New Jersey: Prentice-Hall, Inc.

Holmberg EB, Hillman RE, Perkell JS, Guiod PC, Goldman SL (1995). Comparisons Among Aerodynamic, Electroglottographic, and Acoustic Spectral Measures of Female Voice. *Journal of Speech and Hearing Research* 38:1212-1223.

Holmberg EB, Hillman RE, Perkell JS, Gress C (1994). Relationships Between Intra-Speaker Variation in Aerodynamic Measures of Voice Production and Variation in SPL Across Repeated Recordings. *Journal of Speech and Hearing Research* 37: 484-495.

Holmberg EB, Hillman RE, Perkell JS (1988). Glottal airflow and transglottal air pressure measurements for male and female speakers in soft, normal, and loud voice. *Journal of the Acoustical Society of America* 84(2): 511-529.

Horii Y (1980). An Accelerometric Approach to Nasality Measurement: A Preliminary Report. *Cleft Palate Journal* 17(3):254-261.

Horii Y (1983). An Accelerometric Measure as a Physical Correlate of Perceived Hypernasality in Speech. *Journal of Speech and Hearing Research* 26(3):476-480.

Horii Y, Monroe N (1983). Auditory and Visual Feedback of Nasalization Using a Modified Accelerometric Method. *Journal of Speech and Hearing Research* 26(3):472-475.

Horii Y, Fuller B (1990). Selected Acoustic Characteristics of Voices Before Intubation And After Extubation. *Journal of Speech and Hearing Research* 33(3):505-510.

Ishizaka K, Matsudaira M, Kaneko T (1976). Input acoustic-impedance measurement of the subglottal system. *Journal of the Acoustical Society of America* 60(1):190-197.

Ishizaka K, French JC, Flanagan JL (1975). Direct Determination of Vocal Tract Wall Impedance. *IEEE Transactions on Acoustics, Speech, and Signal Processing* ASSP-23(4):370-373.

Ishizaka K, Flanagan JL (1972). Synthesis of voiced sounds from a two mass model of the vocal cords. *Bell Systems Technical Journal* 51:1233-1268.

Klatt DH, Klatt LC (1990). Analysis, synthesis, and perception of voice quality variations among female and male talkers. *Journal of the Acoustical Society of America* 87(2):820-857.

Lippmann RP (1981). Detecting Nasalization Using a Low-Cost Miniature Accelerometer. *Journal of Speech and Hearing Research* 24:314-317.

Masuda T, Ikeda Y, Manako H, Komiyama S (1993). Analysis of vocal abuse: fluctuations in phonation time and intensity in 4 groups of speakers. *Acta Oto-Laryngologica (Stockh)* 113:547-552.

Milstein CF (1999). Laryngeal Function Associated With Changes In Lung Volume During Voice And Speech Production In Normal Speaking Women. Doctoral Dissertation: University of Arizona.

Ohlsson A-C, Brink O, Löfqvist A (1989). A voice accumulator – validation and application. *Journal of Speech and Hearing Research* 32:451-457.

Pasterkamp H, Kramen SS, DeFrain PD, Wodicka GR (1993). Measurement of Respiratory Acoustical Signals. Comparison of Sensors. *Chest* 104:1518-1525.

Pasterkamp H, Schafer J, Wodicka GR (1996). Posture-dependent changes of Tracheal Sounds at Standardized Flows in Patients with Obstructive Sleep Apnea. *Chest* 110:1493-1498.

Perkell JS, Holmberg EB, Hillman RE (1991). A system for signal processing and data extraction from aerodynamic, acoustic, and electroglottographic signals in the study of voice production. *Journal of the Acoustical Society of America* 89(4), Pt. 1: 1777-1781.

Peterson GE, Barney H (1952). Control methods used in a study of the vowels. *Journal of the Acoustical Society of America* 24: 175-184.

Qi YY (1996a). *derivf.m*, MATLAB® program adapted with permission of the author.

Qi YY (1996b). *hiband.m*, MATLAB® program adapted with permission of the author.

Rabiner LR, Juang BH (1993). *Fundamentals of speech recognition.* New Jersey: Prentice Hall, pp. 97-107.

Ramig LO, Verdolini K (1998). Treatment Efficacy: Voice Disorders. *Journal of Speech Language and Hearing Research* 41:S101-S116.

Rothenberg, M (1992). A Multichannel Electroglottograph. *Journal of Voice* 6(1): 36-43.

Rothenberg M (1973). A new inverse-filtering technique for deriving the glottal air flow waveform during voicing. *Journal of the Acoustical Society of America* 53(6): 1632-1645.

Ryu S, Komiyama S, Kannae S, Watanabe H (1983). A Newly Devised Speech Accumulator. *ORL Journal of Otorhinolarygology and Related Specialties* 45(2):108-114.

Sapienza CM, Stathopoulos ET (1994). Comparison of Maximum Flow Declination Rate: Children Versus Adults. *Journal of Voice* 8(3):240-247.

Shipp T (1975). Vertical Laryngeal Position During Continuous And Discrete Vocal Frequency Change. *Journal of Speech and Hearing Research* 18: 707-718.

Stemple JC, Weiler E, Whitehead W, Komray R (1980). Electromyographic biofeedback training with patients exhibiting a hyperfunctional voice disorder. *Laryngoscope* 90:471-476.

Stevens KN (1998). *Acoustic Phonetics.* Cambridge, MA: MIT Press.

Stevens KN (1977). Physics of laryngeal behavior and larynx modes. *Phonetica* 34: 264-279.

Stevens KN, Kalikow DN, Willemain TR (1975). A miniature accelerometer for detecting glottal waveforms and nasalization. *Journal of Speech and Hearing Research* 18: 594-599.

Stevens KN, Nickerson RS, Boothroyd A, Rollins AM (1976). Assessment of Nasalization in the Speech of Deaf Children. *Journal of Speech and Hearing Research* 19: 393-416.

Titze IR (1984). Parameterization of the glottal area, glottal flow, and vocal fold contact area. *Journal of the Acoustical Society of America* 75(2):570-580.

Weibel ER (1963). *Morphometry of the Human Lung.* Berlin: Springer.

Zenker W (1964). Questions regarding the function of external laryngeal muscles. In D. Brewer (Ed.), *Research Potentials in Voice Physiology* (pp. 20-40). Syracuse, NY: State University of New York.