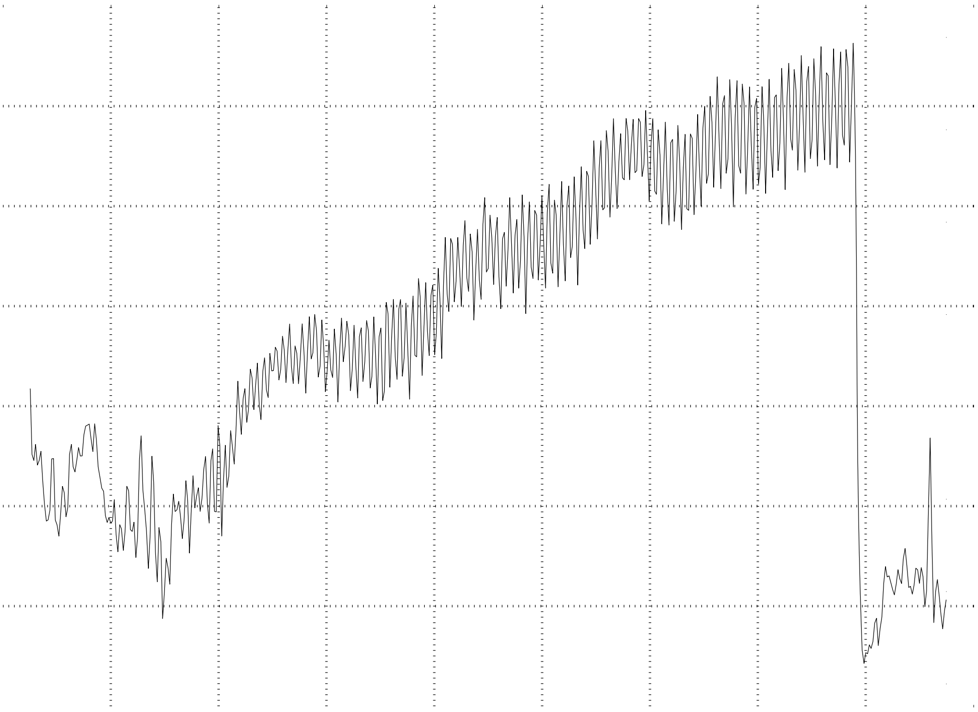
CHALMERS



Tones Rising from Noise

A Study of the Detectability
of Ramped Tones Masked by Noise

Thesis for the Master's Programme in Sound and Vibration

MICHAEL P. MAGILL

Department of Civil and Environmental Engineering
Division of Applied Acoustics
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2005
Master's Thesis 2005:118

Master's Thesis 2005:118

Tones rising from noise

A study of the detectability of ramped tones masked by noise

Michael P. Magill

Department of Civil and Environmental Engineering
Division of Applied Acoustics
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2005

Tones rising from noise
A study of the detectability of ramped tones masked by noise

Master's Thesis 2005:118

Department of Civil and Environmental Engineering
Division of Applied Acoustics
Chalmers University of Technology
SE-41296 Göteborg
Sweden

Tel. +46-(0)31 772 1000

Printed by:
Reproservice / Department of Civil and Environmental Engineering
Göteborg, Sweden 2005

Cover: A Fourier transform of a sample from the Ramped Tone Detection test.

Tones rising from noise
A study of the detectability of ramped tones masked by noise
Thesis for the Master's programme in Sound and Vibration
Michael P. Magill
Department of Civil and Environmental Engineering
Division of Applied Acoustics
Chalmers University of Technology

ABSTRACT

Fan noise from personal computers is, for many, a constant presence. It would be a beneficial development if they were less obtrusive and hardly noticeable. With this in mind, a psychoacoustic listening test was conducted with the goal of determining the limen[1] of detection for a tone ramping like a computer fan increasing its speed to full power. The results of this test indicate that, in the tested octave, a change rate of $\Delta L_p/\Delta t \approx 0.17$ dB/s and $\Delta f/\Delta t \approx 2.9$ Hz/s and lower are detected at approximately the same point in the ramp.

An additional test was conducted as an investigation into the threshold of audibility for hearing the tonal component of a fan in its normal running mode where the only background noise is the computer. This threshold was found to be $L_p \approx 1.5$ dB louder than the background noise.

KEYWORDS: ramped, tone, detection, limen, just-noticeable, threshold.

---

[1] På svenska är "limen" medvetandetröskel.

ACKNOWLEDGMENTS

I greatly appreciate the support given from the following people

- My adviser along this ride, Anders Genell,
- Mendel Kleiner, Daniel Västfjäll and Anders Sköld for answering question after question after question,
- Marco Beltman of the Intel Corporation for supplying the tools needed to complete this project, as well as
- The Intel Corporation as a whole for the grant which funded this Sound Quality of Computing Devices project, and
- Rasmus Elofsson Bernstedt for the many mathematical tips and advice pertaining to cover appearances.

Göteborg, January 9, 2006

MICHAEL P. MAGILL

CONTENTS

Acknowledgments vii

1 Introduction **1**

1.1 Experimental Methods 1

1.2 Difference Limina 2

1.3 Personal Computer Sound Quality Research 3

2 Ramped Tone Detection Test **5**

2.1 Participants 5

2.2 Measures 5

2.3 Stimuli and Presentation 5

2.4 Procedure 7

3 Tone Audibility Test **9**

3.1 Participants 9

3.2 Measures 9

3.3 Stimuli and Presentation 9

3.4 Procedure 10

4 Results **13**

4.1 Ramped Tone Detection Listening Test 13

4.2 Tone Audibility Listening Test 13

5 Conclusions **17**

5.1 Discussion of Ramped Tone Detection Test 17

5.2 Discussion of Tone Audibility Listening Test 18

5.3 Future Work 18

Bibliography **19**

Appendices **21**

A Additional Data and Graphs **21**

A.1 Box-and-whisker Plot, Audibility Test 21

A.2 Box-and-whisker Plots, Ramped Tone Detection Test 21

B Matlab Code **25**

B.1 Listening Test Program 25

B.2 Other Code 35

1 INTRODUCTION

The noise from the fan of a personal computer has become amongst the most frequently heard sounds in office environments. As personal computers become more powerful, they generate more heat and require more cooling. Thus, the fans are becoming more powerful and thereby louder. Additionally, current computers turn their fans on and off as needed; they do not simply fade into the background. The cycle during which these fans power up is a new source of annoyance and the motivation for this study. Perhaps if they ramped up to full power more slowly, they would not be detectable at all and would casually introduce themselves as part of the background noise.

1.1 EXPERIMENTAL METHODS

For this investigation, two listening tests were conducted. Chronologically, the first was to determine the audibility limen of standard personal computing fans running at full power in the presence of a background noise supplied by Intel. It will be referred to as the Tone Audibility test. The second test attempted to determine the detection limen of these same fans during their ramping stage. As such, it is called the Ramped Tone Detection test. The next few paragraphs discuss the historical basis for the methodology used.

1.1.1 ADAPTIVE LEVEL DETERMINATION

Listening tests involving some comparison or judgment are extremely common in psychoacoustics, especially in liminal studies. For example, if the audibility threshold of a tone in small band noise is desired one might have a listener compare a series of paired sounds where one contains the tone at a number of levels and the other has no tone. A fixed method would use every member of this series. But, a more efficient method by which the series of comparisons can be made was developed by Taylor and Creelman (1967) named the Parameter Estimation by Sequential Testing (PEST) method.

The important development introduced by the PEST adaptive method was that the step size of stimuli from one sample to the next is determined by the test-taker. That is, after the listener answers, the test giver supplies another stimulus and the fixed method would have the steps between stimuli be the smallest increment of the stimulus series. With PEST, as long as the taker answers such that s/he orbits his/her limen, the step sizes get smaller. Again take the above example: if s/he is asked "Was the tone in the second or in the first spot?"[1] and answers correctly, the

---

[1] Kaplan (1999)

step size decreases. An incorrect answer yields the previous sound. Two incorrect answers yield an increase in step size. Eventually, using this algorithm, the smallest step size is reached and so finally, the threshold being measured is determined.

1.1.2 BÉKÉSY AUDIOMETER

In 1946, Georg von Békésy developed an audiometer at the Karolinska Institute in Stockholm. It operated on a slightly different principle than contemporary audiometers: rather than using pulses, it ramped through the frequency spectrum. The test-taker would listen as a tone ramped from 100 Hz to 10,000 Hz over fifteen minutes, or $\Delta f/\Delta t \approx 11$ Hz/s. The tone would also ramp up and down in amplitude depending on whether or not the listener could hear it. The amplitude changed at a rate of $\Delta L_p/\Delta t \approx 0.67$ dB/s.[2] Because the design was simple and affordably produced, this audiometer was used globally for decades.

The ramping aspects of this test equipment are obviously similar to the Ramped Tone Listening test and served as a model. In fact, the ramping rate of most personal computing fans is practically equal to the change rates Békésy chose. One important difference between the audiometer and the Ramped Tone listening test is that while the audiometer only measured intensity, the listening test requested that the listener pay attention to changes in both frequency and amplitude simultaneously.

1.2 DIFFERENCE LIMINA

The two limina that are points of concern for this study are those of frequency and amplitude. These two thresholds are amongst the most investigated points in psychoacoustics. As is suggested in Zwicker and Fastl (1999), there are two distinct versions of just-noticeable changes. One is where the analyzed trait changes continuously, called "Variation," and the other is where the change is abrupt, called "Difference." The Ramped Tone Detection test is an exercise in the "Variation" version of threshold. Likewise, the Tone Audibility test works with the idea of the "Difference" version.

Moore (1997) and Jesteadt *et al.* (1977) found that the amplitude limen follows Weber's Law.[3]

$$\frac{\Delta L_p}{L_p} = K \tag{1.1}$$

For the range of levels for this project, they found the threshold of detection to be $\Delta L_p \approx 0.4$ dB. In Wier *et al.* (1977), the frequency limen is also shown to follow Weber's Law and in the range this study investigated, their limen range is reported

---

[2] von Békésy (1947)

[3] E.H. Weber, a German physician, developed this law and laid the groundwork for the field of psychophysics in about 1860.

as being $\Delta f = 0.8 - 1.5$ Hz. However, nearly all the studies[4] done on these subjects have been conducted using pulses, *i.e.* Zwicker's "Difference" form. The results of the "Variation" form are significantly less documented but Zwicker and Shower and Biddulph (1931) modulated a tone and found a limen of $\Delta f \approx 4$ Hz in this range.

1.3 Personal Computer Sound Quality Research

As one might expect, most of the acoustic research done on personal computers deals with noise control and specifically meeting the standards set in, among others, ECMA-109 (1996) and ISO-9296 (1988). For example, Robert D. Hellweg *et al.* have been publishing since the mid-1990s on the subject of reducing the sound power emitted from portable personal computers.[5] Some research focuses on meeting both acoustic and thermal requirements (Shaw and Cruz 2003). Also, there is research done which seeks to improve the existing standards (Man 2004).

As for the sound quality of personal computers, much less has been done. The two cited standards have an optional appendix concerned with the character of the sounds emitted by office equipment, but the characterization is limited to the existence of either transient noises or a distinct tone. Some of the research that has been done on sound quality include Nelson and Balant (2000) who investigated how prominent the tonal component from computers is. Additionally, Parker (2001) conducted a study on the sound quality of personal computers and how it affects customer acceptance and DeMoss (2001) administered a listening test comparing the preferences of personal computer sound between the genders.

A Note on Statistical Methodology

For a normally distributed set of data X, the standard deviation (SD) is a good formula to use to adequately describe how concentrated or diffuse this data is. It is calculated as follows for total number N of entries in the set and for the mean $\bar{x}$:

$$\sigma = SD = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2} \tag{1.2}$$

This method is so successful that the symbol σ is used synonymously with the SD. However, for sets of data that are not so beautifully Gaussian, there are other ways of calculating σ. One of these will be described here.

Firstly, the concept of a quartile is defined. An abnormally distributed data set Y can be ordered by magnitude and divided into four subsets (assuming the contents are real numbers). Each of these subsets contains one quarter of the data such

---

[4] *e.g.* Harris (1952), Rosenblith and Stevens (1953), Nordmark (1968) and Moore (1973)

[5] Hellweg (1996), Hellweg *et al.* (2003), Hellweg *et al.* (2005).

that the first subset contains the lowest items by magnitude and the fourth subset contains the highest.

The numbers which divide the subsets are called quartiles. They can be calculated, as the software Matlab does, by taking Y with n components such that $\{y_1 \leq y_2 \leq \ldots \leq y_n\}$, a quartile index j, and k rounded up,

$$Q_j = \frac{3y_k + y_{(k+1)}}{4} \quad \text{where} \quad k = \left\lceil \frac{n \cdot j}{4} \right\rceil \tag{1.3}$$

In particular, the second quartile is more frequently referred to as the median. The difference $Q_3 - Q_1$ is called the interquartile range (IQR) and a normalized version of this quantity is used in this study to describe the distribution. The process of normalization is discussed in the following.

The ideal and normally distributed set X can be divided by quartile too. Using the values of these quartiles, the normalizing constant is determined:

$$C_{normalizing} = \frac{SD_x}{Q_{3,x} - Q_{1,x}} = 0.7413 \tag{1.4}$$

Multiplying this proportion with the IQR gives the normalized interquartile range ($NIQR$) which serves as an alternative definition of σ if a few criteria are met.

$$\sigma = NIQR = (Q_{3,y} - Q_{1,y}) \times C_{normalizing} \tag{1.5}$$

These criteria are a high concentration of points around the median and the presence of a few points very far from the median (termed: outliers). Finally, there must be outliers on both sides of the median. These outliers are the primary reason for using the $NIQR$ because they inflate the SD while offering little information about the trend the data would show. The IQR by definition excludes extreme cases and focuses the attention on the middle fifty percent of the data. But, if the distribution is skewed so that a significant percentage of the data lies in the first or fourth quartile then both these formulae will fail to describe the data. That is, the IQR method requires the median be close to the center of the entire range and that the outliers represent the extremes above and below this median.

2 RAMPED TONE DETECTION TEST

2.1 PARTICIPANTS

Thirty-one undergraduate and graduate students of Göteborg University and Chalmers University volunteered and were compensated with a coupon worth one cinema ticket (US $10). Nine of the participants were female, and the average age was 25.5 years (SD 2.4). Ethnographically, twenty-four of the test-takers were Swedish and the rest were from Europe and North America. All listeners passed an audiometric test for normal hearing using the departmental audiometer.

2.2 MEASURES

The measure of detection was the time from when the user pressed a button to when s/he heard a change in the sounds they heard. The reaction time of each participant was also measured, in a manner to be discussed later, for the purpose of subtracting from the measured detection time.

2.3 STIMULI AND PRESENTATION

The stimuli consisted of ten files of sine wave tones constructed using Adobe Audition software that all ramp from 350 Hz to 700 Hz and from 44 dB SPL to 64 dB SPL. The stimuli differed in how slowly they changed ranging from 15 seconds to 540 seconds as is described in Table 2.1.

Sample Length [s]	$\Delta L_p/\Delta t$ **[dB/s]**	$\Delta f/\Delta t$ **[Hz/s]**
15	1.3	23.3
30	0.67	11.7
60	0.33	5.8
90	0.22	3.9
120	0.17	2.9
180	0.11	1.9
240	0.083	1.5
300	0.067	1.2
420	0.048	0.83
540	0.037	0.65

Table 2.1. Differences in Ramped Tone Stimuli

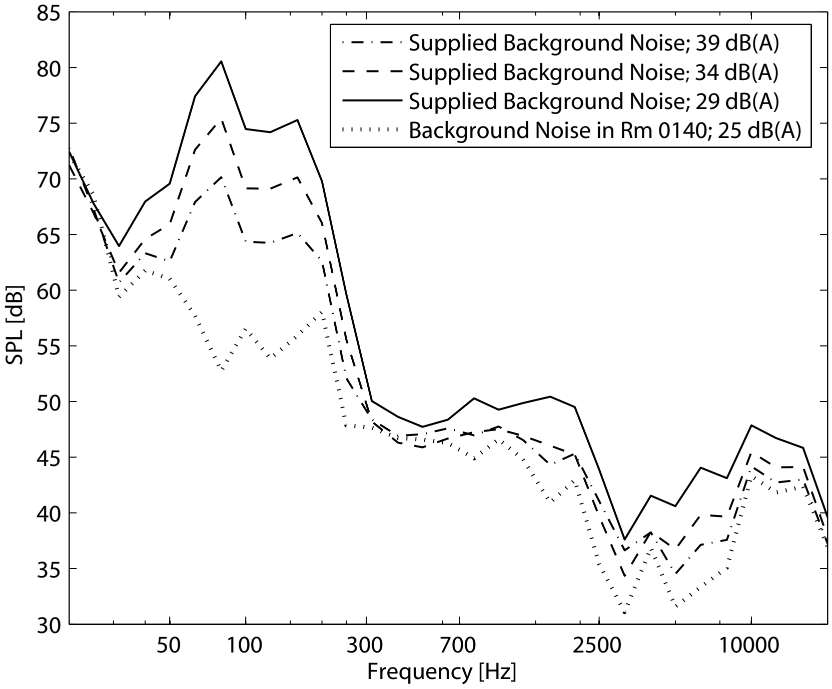


Figure 2.1. Background Noises analyzed in Third-Octave Bands, Room 0140 is in the Applied Acoustics building.

Additionally, there were dummy sounds made in the same manner and range as the stimuli but of square waves. These sounds were of the lengths 15, 30, 60 and 90 seconds. The intention for presenting these sounds was to refresh the listener and to keep him/her from learning to listen only for sine waves.

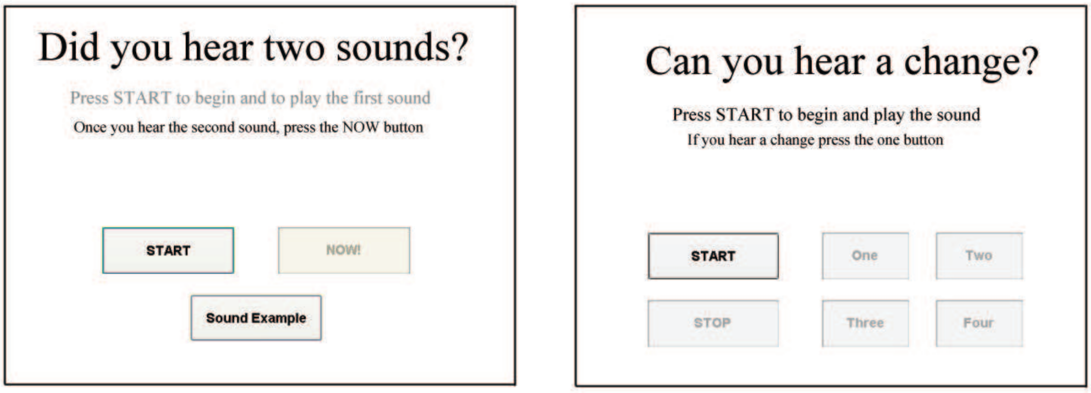
Along with the tones, there was a background noise presented that had the same spectrum as the binaurally recorded noise supplied by Intel but whose phase had been scrambled - to eliminate directionality.[1]

Each test taker was exposed to two of the three different noise levels: 29, 34 and 39 dB(A). The third-octave band spectra of the different noises are shown in Figure 2.1 including the natural background noise in the room with no inserted sounds. While the natural background noise appears to be slightly louder than the supplied noise at some frequencies, it is more probable that the supplied noise was the same level as the natural noise at these frequencies. The apparent discrepancy stems from measurement noise.

The sounds were presented using AKG K1000 headphones, which do not come into physical contact with the ears. They were powered by a NAD Model No. 3020 stereo amplifier.

---

[1] Code shown in B.2.1 on page 35.



Did you hear two sounds?

Press START to begin and to play the first sound

Once you hear the second sound, press the NOW button

Can you hear a change?

Press START to begin and play the sound

If you hear a change press the one button

Figure 2.2. The graphical interfaces encountered during the test. The left is of the Reaction Time test and the right is of the Ramped Tone test.

2.4 PROCEDURE

To take the test, an individual came to the testing room and was given the audiometric test. Following this, the listener was verbally instructed for the Ramped Tone test and was left alone in the room. The test was administered on a personal computer.

The first step for the listener was to complete a Reaction Time test.[2] The Reaction Time test required the listener press a button to start a timer and a pop of noise. After a random amount of time had passed another pop sounded at which point the subject was to press the button again.

After this, the listener took a one-sound tutorial for the Ramped Tone test. This consisted of the 15 second square wave dummy sound and ran exactly as the actual test but the results were not recorded. The graphical interfaces used are pictured in Figure 2.2.

Finally, the portion of the test wherein data were collected began. The program randomized the order of the 10 test stimuli and the dummy sounds and also randomly chose a background noise level.[3] The listener pressed the "Start" button when s/he was ready to begin and the first sound file was then played. The user was to concentrate and press the "One" button when s/he detected any change in the sounds s/he was hearing.

Pressing the "One" button caused the program to play the file in reverse; that is, from 700 Hz/64 dB SPL downward to 350 Hz/44 dB SPL. Here, the user was asked to press the "Two" button when a change in the sound was detected. Pressing the

---

[2] Code for the whole program shown in B.1 on page 25.

[3] Some early test-takers had a total of 22 sounds per background noise level to listen to. This was reduced to 19 when it was observed that many users were taking an excessively long time to complete the test.

Figure 2.3. The test taking environment. The left shows the audiometer portion of the testing procedure and the right shows the ramping portion.

"Two" flipped the file again and the process repeated. The user listened for a change in the upward direction and eventually pressed "Three" and finally in the downward direction again and pressed "Four." Pressing this button advanced the program to the next sound.

The user listened to and detected all the sounds and upon completion began again with a different background noise level and a different sound order. After completing these tasks, the listener was engaged in an informal conversation regarding his/her opinions of the test. Finally, s/he received the compensation and departed. The whole test took approximately 50 minutes.

3 TONE AUDIBILITY TEST

3.1 PARTICIPANTS

A professor and nineteen undergraduate and graduate students of Chalmers University and Göteborg University volunteered and were not compensated except with gratitude. Seven of the participants were female and the average age was 27.6 years (*SD* 8.0 and *IQR* 3)[1]. Ethnographcially, 9 were Swedish and the rest were from many different countries.

3.2 MEASURES

The audibility measure consisted of a forty-one point scale where the level of the tone differed by one decibel from point to point. The reference for the scale was the sixth highest sound level and was named "Zero dB." Thus, the highest point was referred to as "+5 dB" and the lowest was "−35 dB."

3.3 STIMULI AND PRESENTATION

Ten variations of typical tone and overtone combinations of stationary-power computer fans were created using Adobe Audition software. The tones were mixed with a binaurally recorded background noise supplied by Intel also using Adobe Audition. Table 3.1 describes the different components contained within each of the stimuli.

	700 Hz [dB(SPL)]	**1400 Hz [dB(SPL)]**	**2100 Hz [dB(SPL)]**	**Maximum Level [dB(A)]**
Sound 1	68	none	none	34
Sound 2	68	66	none	36
Sound 3	68	66	64	37
Sound 4	68	69	none	38
Sound 5	68	64	none	35
Sound 6	68	69	67	39
Sound 7	68	63	61	36
Sound 8	68	none	67	37
Sound 9	68	none	64	36
Sound 10	68	none	61	35

Table 3.1. The different stimuli used and their levels at maximum ("+5 dB").

---

[1] See page 3 for a definition of these statistical terms.

All the values shown are of the "+5 dB" case. The minimum values (the "−35 dB" point) were all the level of the background noise, which was $L_p \approx 29$ dB(A). The supplied background noise was a recording of a computer running in an anechoic room. The third-octave band spectra of the supplied noise and is shown in Figure 3.1. Also shown is the $L_p \approx 19$ dB(A) background noise of the room without any sound sources from the experiment. The testing area is portrayed in figure 3.2. The sounds were presented to the listeners using STAX headphones.

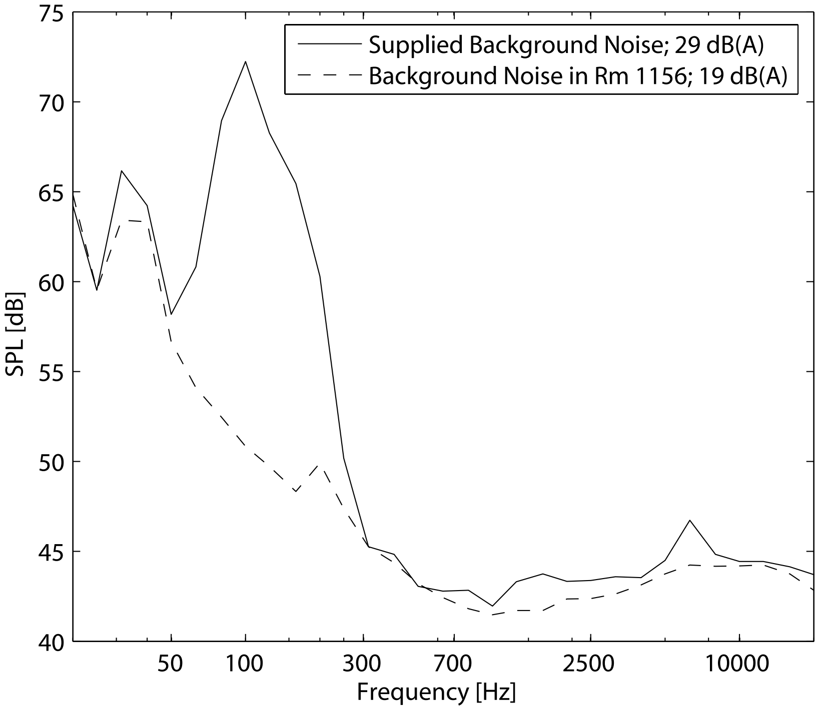


Figure 3.1. Background noises in third-octave bands, Room 1156 is in the Applied Acoustics building.

3.4 PROCEDURE

The test was taken individually and using a personal computer. The program was an implementation of the PEST[2] method that was designed by Peter Mohlin of the Department of Applied Acoustics. The order of the ten sounds was randomized by the program so the sounds were presented in a different order to each listener.

The following paragraphs will describe the adaptive, iterative nature of the PEST program. After receiving verbal instructions about how to take the test, the listener was left alone in the testing room. S/he pressed a button to start the first sound

---

[2] Parameter Estimation by Sequential Testing, an adaptive testing method developed by Taylor and Creelman (1967). There is a brief description of the method in 1.1.1.

Figure 3.2. The test taking environment.

and the maximum level of this sound was played. Since the tones being tested were stationary, the listener could, at any point during the playback, respond to the on-screen question “Can you hear a tone?” by pressing “Yes” or “No”.

When the listener presses “Yes,” the PEST program switches to the minimum level. Again, the listener presses a button to start playback. If the listener presses “No” during this playback, the program raises the level and selects a file between the extremes to be played. After starting the next sound, the listener makes a judgment and presses either button causing PEST to raise or lower the value accordingly.

As long as the user switches between louder and softer sounds, PEST decreases the size of the steps. That is, the first step is down 35 dB, from maximum to minimum, and the second step is up 23. The third step is back down 15 and so on. Finally, the listener converges to his/her audibility limen recognized by the PEST program as a vacillation between two minimally spaced steps apart (in this case one dB).

If the listener presses “Yes” or “No” two or more times in a row, PEST increases the step size while moving in the direction indicated. When the listener can (or cannot) hear the tone any longer, the program switches direction and again starts decreasing the step sizes. The test took a total of approximately thirty minutes.

4 RESULTS

4.1 RAMPED TONE DETECTION LISTENING TEST

Figures 4.1 and 4.2 show the results of the Ramped Tone Detection listening test involving ramped tones.[1] The data were calculated by multiplying the time of detection with the corresponding change rate in table 2.1 on page 5. The values for σ were calculated by normalizing the interquartile range[2] (IQR) of each point.

The horizontal threshold of hearing line in figure 4.1 was approximated using the results of the Tone Audibility listening test and the sound samples started from the bottom axis (350 Hz/ 44 dB SPL). For figure 4.2, the sound started at 700 Hz/ 64 dB SPL and decreased. Box-and-whisker plots of all these data are presented in Appendix A.2.

The rightmost two values for σ from 4.2 seem to be erroneous estimations of the distribution compared with the other results. The histogram in figure 4.3 shows that more than half of the points fall into just one of the bins which is indicative of the results for some of the extreme cases.[3] Thus for these points, while a good median can be calculated; a good indicator of the distribution cannot.

4.2 TONE AUDIBILITY LISTENING TEST

The data shown in table 4.1 indicate how far below the "Zero dB" level the threshold of hearing was. It is listed in these units because $L_p \approx 29$ dB(A) for all the sounds when the tones are at their limina. The data are listed from lowest to highest. (*cf.* Table 3.1 on page 9 for how the different sounds were constructed and see section A.1 on page 21 for more about the distribution.)

The values for σ were again calculated by normalizing the IQR. The third column shows the percentage of the listeners whose limen is the median. Since an exceptionally low value for n/N might indicate some unusual behavior, the histogram of the results from Sound Four is shown in Figure 4.4.

---

[1] The axes in these two figures should be read as happening simultaneously. The ramped tones changed in both amplitude and frequency at the same time so the two sets of axes are directly linked.

[2] See page 3 for a definition of the IQR.

[3] It should be noted that two of the points in this histogram have exceeded the outer limit. That is, two listeners clicked after the sound file had finished playing. Since this is when these users detected a change, the points have been included as is. However, since the IQR uses only the time from the end of the third quartile to the beginning of the second, the extreme points do not affect σ. The fact that the first quartile lacks outliers does affect it though, because the third quartile ends up encompassing some of the outliers which increases the IQR.

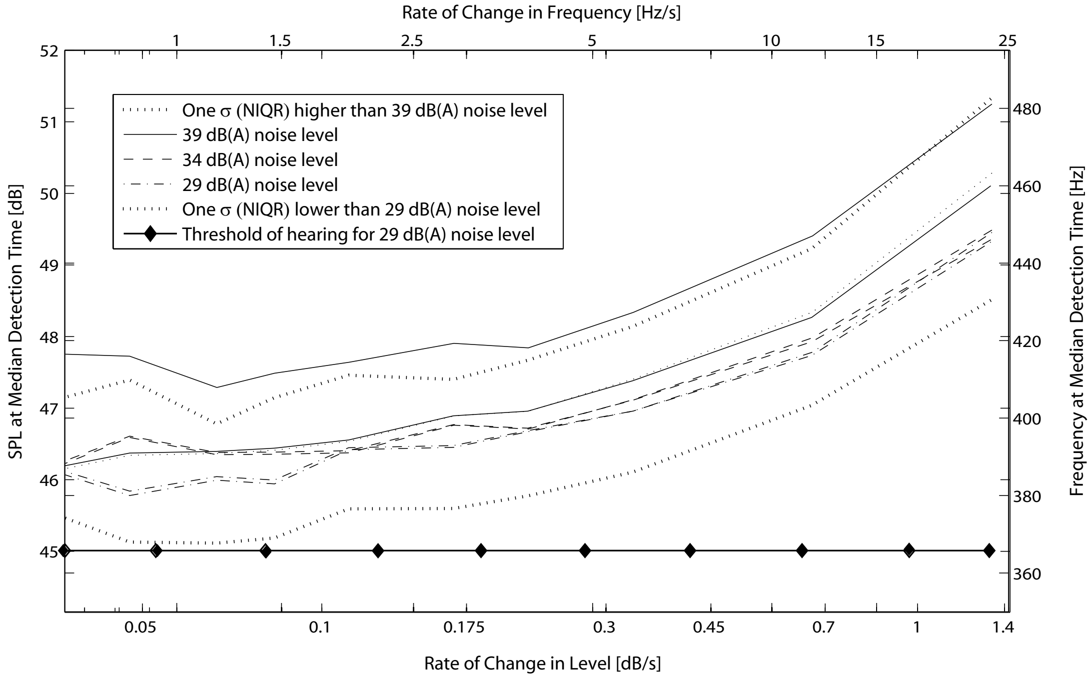


Figure 4.1. Median detection of upwardly ramped tone shown in terms of SPL and frequency.

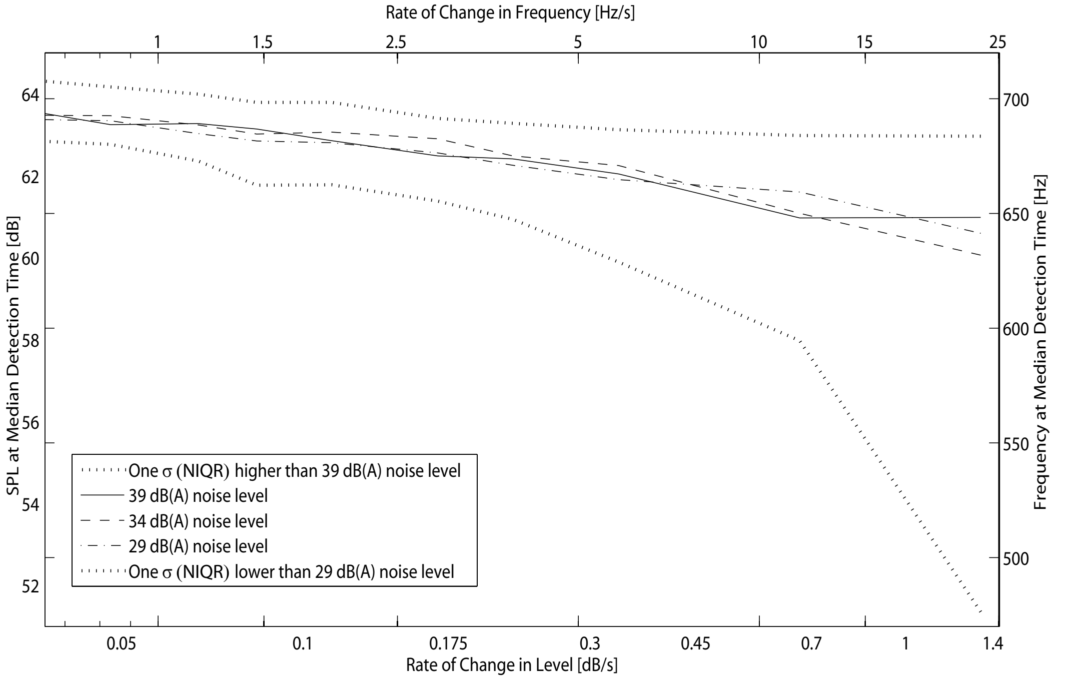


Figure 4.2. Median detection of downwardly ramped tones shown in terms of SPL and frequency.

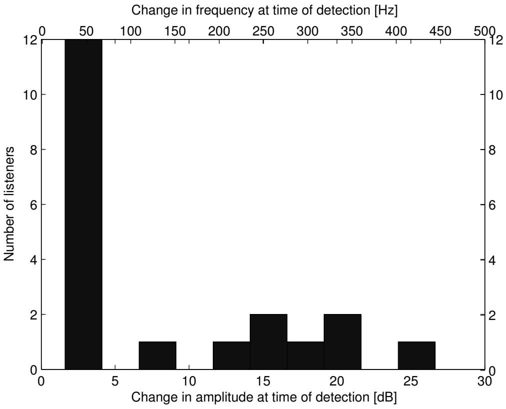


Figure 4.3. Histogram of the 15 second sample for 29 dB(A) noise level of the Ramped Tone Detection test.

	Median [dB]	**σ ($NIQR$)** [dB]	**n/N** [%]
Sound 6	-35	2.0	45
Sound 3	-32	3.9	20
Sound 4	-32	3.7	10
Sound 8	-31	3.3	20
Sound 7	-30	3.0	30
Sound 9	-30	3.0	25
Sound 1	-30	4.1	25
Sound 5	-30	3.9	10
Sound 2	-29	4.1	25
Sound 10	-29	3.3	30

Table 4.1. Median audibility levels and descriptions of their distributions. Median values are in dB of the tone relative the maximum. $N = 20$ and is the total number of participants and n is the number of participants whose limen was the median value.

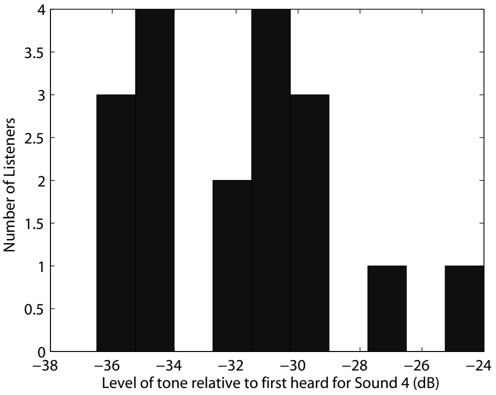


Figure 4.4. Histogram of Sound Four from the Tone Audibility test.

5 CONCLUSIONS

5.1 DISCUSSION OF RAMPED TONE DETECTION TEST

5.1.1 RAMPED TONE TEST UPWARD

The results for the upward ramping portion of the Ramped Tone test (figure 4.1) are very consistent. The data indicate that at these noise levels and in this frequency range, humans can detect the presence of a ramping tone at practically the same point in the ramp for change rates of $\Delta L_p/\Delta t \approx 0.17$ dB/s and $\Delta f/\Delta t \approx 2.9$ Hz/s and less. This point corresponds to a ramp time of three minutes and longer. The asymptotic appearance of the graph could suggest that people need the same amount of change in the ramp to determine that the sound had changed, but could also indicate the location of the threshold of hearing.

The change in background noise level seems to have made little difference in the detection time. The loudest took slightly longer, but is not outside the range of one σ. However, the three noise levels chosen are all very quiet and the spread between them is not very great.

The distribution of the data is very uniform across the entire range of change rates. This is somewhat surprising because it was conjectured that either the faster sounds would have a broad distribution due to varying reaction times or the slower sounds would have a broad distribution because different people would detect a change at different times.

5.1.2 RAMPED TONE TEST DOWNWARD

The results corresponding to the downward ramping portion of the test (figure 4.2) are also consistent, but offer little clue as to if/when humans can no longer hear that a sound is ramping. As with the upward portion, the slowest sounds were detected first but the range between the highest and the lowest values is about half in the downward case. In fact, these results are practically equal through the entire range of change rates.

This indicates some kind of communication breakdown between the test-giver and the listeners. It is possible that a good amount of the listeners simply interpreted the sudden presence of the tone as being a change from the quiet level they were hearing during the upward portion.

The distribution also reflects a problem with the instructions and/or question asked. For the long sounds, the variance is very low, while for some of the short samples, the variance is outrageously large. As with the upward case, the hypothesis was that the opposite of this graph would be the result. It was expected that the slower sounds would be detected as changing later and with a larger variance than the quicker sounds. Perhaps better results could be achieved with more comprehensive instructions.

5.2 Discussion of Tone Audibility Listening Test

The audibility levels found by the test are as expected, hovering in the range where the signal to noise ratio is zero. Additionally, all values of σ are low and similar to one another - an indication of comparability.

The sounds where $n/N \geq 25\%$ have a strong clear peak and most of those exhibit a classic normal distribution. The ones where n/N is low, however, appear similar to Figure 4.4 which seem to suggest two peaks. More testing is neccessary to see if the two peaks are signifcantly different or if they would blend together as one peak.

5.3 Future Work

This study was successful in that it offers clues as to what the limen of detection for ramped tones is. The upward portion suggests that the limen of ramped detectability is not simply the oft-researched difference limina of intensity and frequency. To find a more specific answer, these clues must be followed through further investigation.

For instance, the test design was not especially practical for finding out when a person does not notice a difference. The test specifically asked the subjects to try and notice a difference. Many modifications to the test using the same procedure could possibly fix this problem: using some tones that do not change at all, using some tones that only change in one of the two parameters, mixing the directions of change such, *etc.* Another possibility is to ask a different question, or use a different base stimulus. Perhaps actual recordings of ramping fans along with the question, “Can you hear a fan?” or, “Can you hear a fan changing speed?” would yield more conclusive results.

Once a more successful test has been implemented, it would be interesting to see how different frequency bands, broader frequency ranges, different SPL levels and ranges and/or louder background noises affect detectability. Along an entirely different tangent, it would be interesting to investigate whether or not the ramp times proposed in this report affect either working efficiency or the preference of computer sounds at all.

BIBLIOGRAPHY

DeMoss, Jeffrey (2001). Comparing sound quality preferences of personal computers between men and women. Noise-Con. Portland, Maine.

ECMA-109 (1996). Declared noise emission values of information technology and telecommunications equipment.

Harris, J.D. (1952). Pitch discrimination. *J. Acoust. Soc. Am.*, **24**, 750–755.

Hellweg, Robert D. (1996). Room sound pressure levels from computer and business equipment meeting sound power level criteria. Noise-Con. Seattle, Washington.

Hellweg, Robert D., Dunens, Egons K. and Baird, Terry (2003). The control of computer and portable equipment noise in classrooms. Noise-Con. Cleveland, Ohio.

Hellweg, Robert D., Dunens, Egons K., Baird, Terrence and Olsen, John N. (2005). Personal computer, printer, and portable equipment noise in classrooms. Noise-Con. Minneapolis, Minnesota.

ISO-9296 (1988). Declared noise emission values of computer and business equipment.

Jesteadt, Walt, Wier, Craig C. and Green, David M. (1977). Intensity discrimination as a function of frequency and sensation level. *J. Acoust. Soc. Am.*, **61**, 169–177.

Kaplan, Howard L. (1999). The Taylor and Creelman Procedure PEST: Parameter Estimation by Sequential Testing. *http://www.thrinberry-frog.com/VirtualSongbook/PEST.pdf.*

Man, Kaleen Xiuting C. (2004). Test code vs. in-situ sound pressure levels for computers. Noise-Con. Baltimore, Maryland.

Moore, B.C.J. (1973). Frequency-difference limens for short duration tones. *J. Acoust. Soc. Am.*, **54**, 610–619.

Moore, Brian C.J. (1997). *An Introduction to the Psychology of Hearing*. 4th edn. Academic Press, London.

Nelson, David A. and Balant, Anne C. (2000). Computer sound quality: Masking, loudness and prominence. Noise-Con. Newport Beach, California.

Nordmark, J.O. (1968). Mechanisms of frequency discrimination. *J. Acoust. Soc. Am.*, **44**, 1533–1540.

Parker, John (2001). Using sound quality to assess customer acceptance of sounds associated with the operation of personal computers. Noise-Con. Portland, Maine.

Rosenblith, W.A. and Stevens, K.N. (1953). On the DL for frequency. *J. Acoust. Soc. Am.*, **25**, 980–985.

Shaw, Jennifer A. and Cruz, Ethan E. (2003). Challenges in designing covers for high-end computer systems: Meeting acoustical and thermal requirements. Noise-Con. Cleveland, Ohio.

Shower, E.G. and Biddulph, R. (1931). Differential pitch sensitivity of the ear. *J. Acoust. Soc. Am.*, **3**, 275–287.

Taylor, M.M. and Creelman, C. Douglas (1967). PEST: Efficient estimates on probability functions. *J. Acoust. Soc. Am.*, **41**, 782–787.

von Békésy, G. (1947). Über ein neues audiometer. *Archiv für Elektrischen Übertragung*, **1**, 13–16.

Wier, Craig C., Jesteadt, Walt and Green, David M. (1977). Frequency discrimination as a function of frequency and sensation level. *J. Acoust. Soc. Am.*, **61**, 178–184.

Zwicker, E. and Fastl, H. (1999). *Psychoacoustics, Facts and Models.* 2nd edn. Springer-Verlag, Berlin.

A Additional Data and Graphs

A.1 Box-and-whisker Plot, Audibility Test

Figure A.1 is a standard Box-and-whisker plot wherein the horizontal line segments within the boxes show the median value for that collection of data. The upper and lower line segments show the top of the third quartile and bottom of the second quartile respectively. In other words, the distance from the top of the box to the bottom is the interquartile range (IQR)[1]. The "whiskers" show how spread apart the rest of the data is and stretch, at most, $1\frac{1}{2}$ times the IQR from the median. The markers outside the whiskers are the location of the remaining outliers.

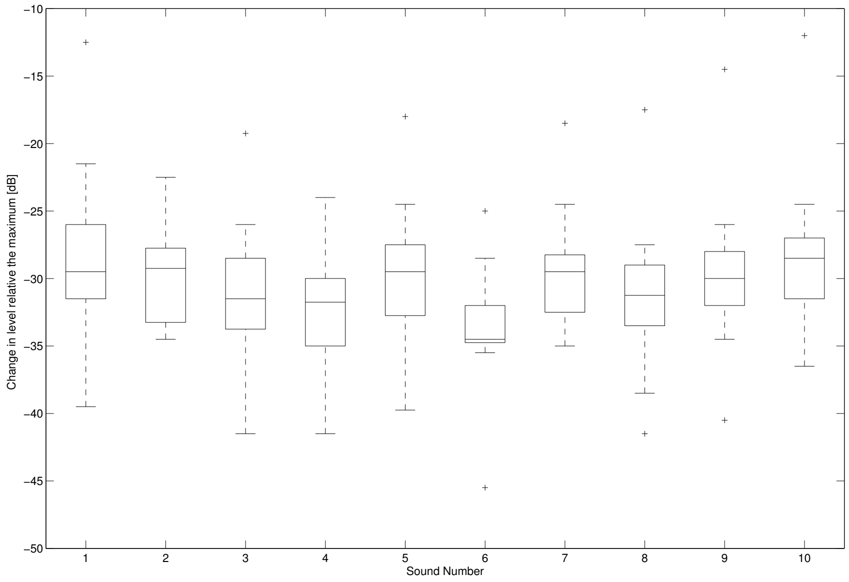


Figure A.1. Box-and-whisker plot for the ten sounds of the Tone Audibility test.

A.2 Box-and-whisker Plots, Ramped Tone Detection Test

The figures on the next few pages are modifications of the standard Box-and-whisker plots. The vertical axes are time and the horizontal axes are rates of change. So, the left-most data points are the slowest moving samples. The boxes and whiskers are as described in the previous section, a dashed line has been added showing the trends of the medians and a dotted line shows the maximum sample length.

---

[1] See page 3 for a definition of the IQR.

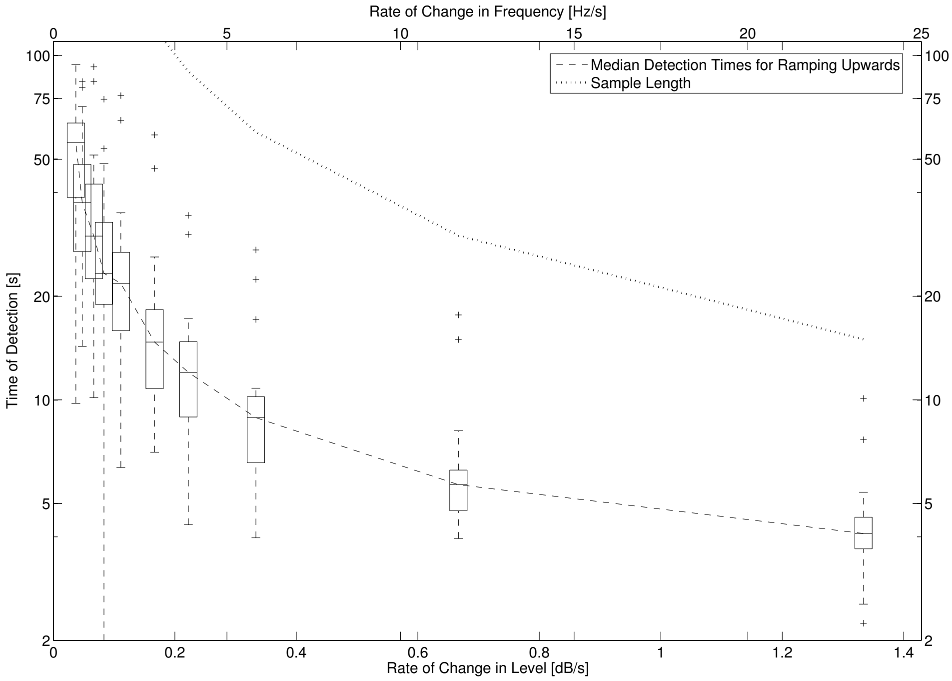


Figure A.2. Box-and-whisker plot for the upward case of the ten sound lengths of the Ramped Tone Detection test in the 29 dB(A) noise setting.

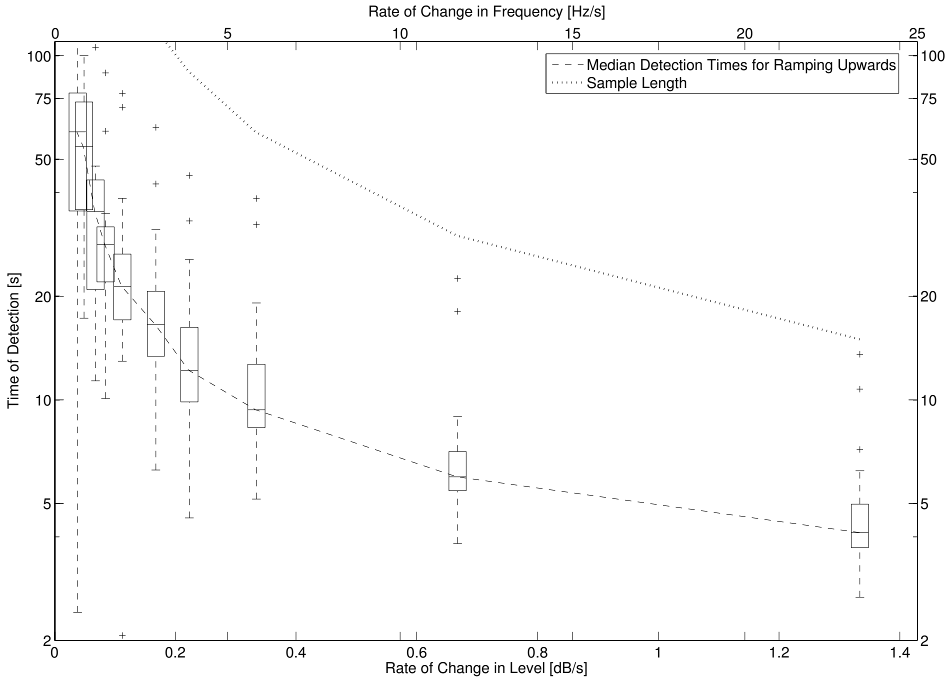


Figure A.3. Box-and-whisker plot for the upward case of the ten sound lengths of the Ramped Tone Detection test in the 34 dB(A) noise setting.

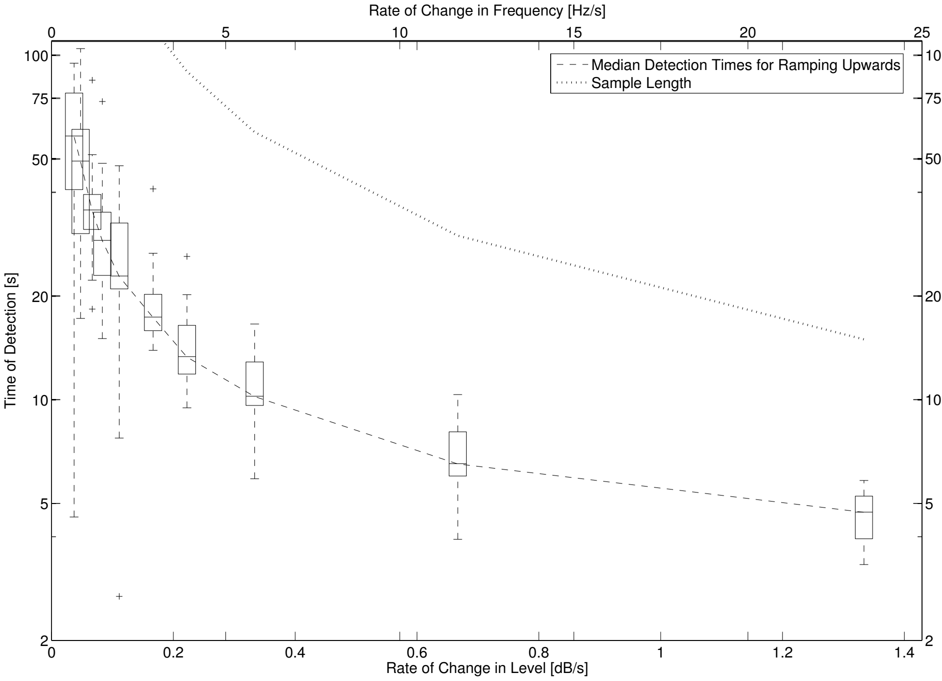


Figure A.4. Box-and-whisker plot for the upward case of the ten sound lengths of the Ramped Tone Detection test in the 39 dB(A) noise setting.

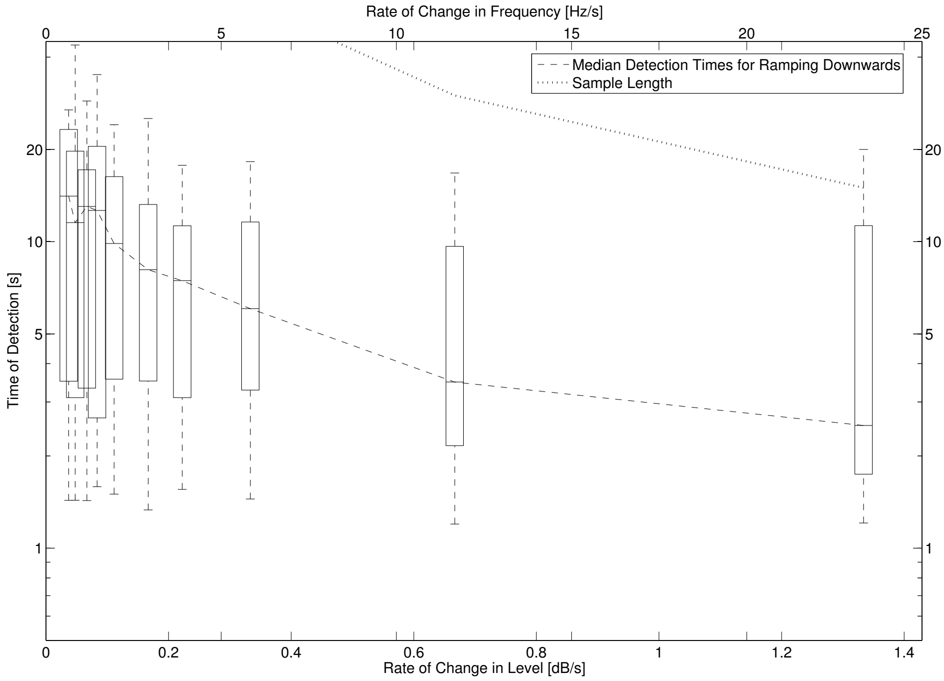


Figure A.5. Box-and-whisker plot for the downward case of the ten sound lengths of the Ramped Tone Detection test in the 29 dB(A) noise setting.

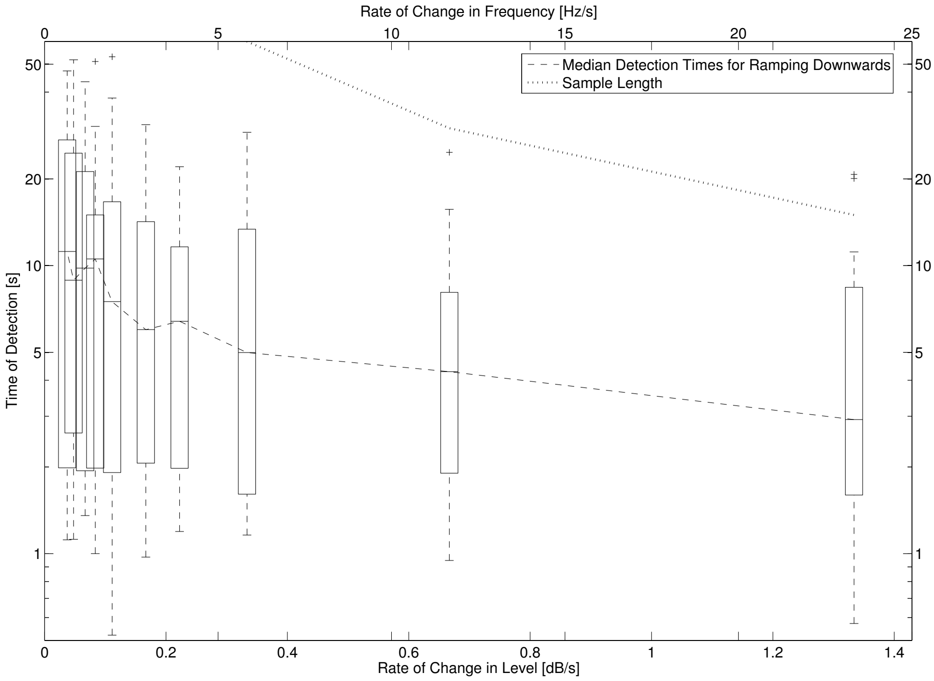


Figure A.6. Box-and-whisker plot for the downward case of the ten sound lengths of the Ramped Tone Detection test in the 34 dB(A) noise setting.

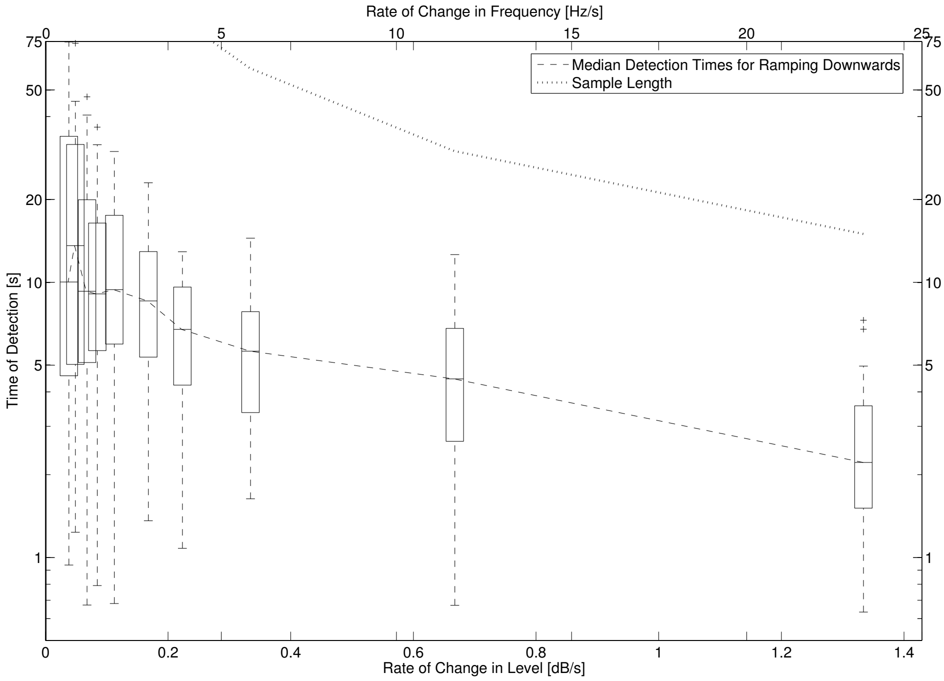


Figure A.7. Box-and-whisker plot for the downward case of the ten sound lengths of the Ramped Tone Detection test in the 39 dB(A) noise setting.

B Matlab Code

B.1 Listening Test Program

These will be presented in the basic order they are encountered when actually taking the test.

B.1.1 testbegin.m

```
clear all;
userid='8';
testiteration=1;
fi1=figure;
axis off
set(fi1,'Units','normalized');
set(fi1,'Position',[0.005,-0.1,0.99,1]);
set(fi1,'NumberTitle','off');
page1title=['Welcome to the Ramped Sound Listening Test!'];
set(fi1,'Name',page1title);

uptext1=text(0.0005,0.9,'Welcome to the Ramped Sound','FontSize',45);
uptext4=text(0.05,0.8,'Listening Test!','FontSize',45);

uptext2=text(0.05,0.7,'Please type your initials followed by your age',
'FontSize',20,'Color',[0.6 0.6 0],'FontWeight','bold');
uptext5=text(0.05,0.65,'in the text box with no spaces: mpm26',
'FontSize',20,'Color',[0.6 0.6 0],'FontWeight','bold');

uptext3=text(0.1,0.58,'Once that is done, please press the Okay
button.','FontSize',17);
uptext6=text(0.1,0.54,'Please follow instructions!','FontSize',17,
'Visible','off');

commentfield=uicontrol('Style','edit','Units','normalized','FontSize',
20,'Max',2,'String','Initials and age please!','Enable','on',
'Position',[0.2,0.45,0.4,0.05],'HorizontalAlignment','left',
'BackgroundColor',[0.6 0.6 0.9],'Visible','on');
okay_button=uicontrol('Style','pushbutton','Units','normalized',
'BackgroundColor',[0 0.9 0],'FontSize',14,'Enable','on','FontWeight',
'bold','String','Okay','Position',[0.35,0.33,0.15,0.07],
'TooltipString','Click here to move on and take the test',
'CallBack','okaycheck');
```

B.1.2 OKAYCHECK.M

```
userid=get(commentfield,'String');

idsize=size(userid);

if idsize(2)==5 | idsize(2)==4
    close all;
    reacttest;
else
    set(uptext6,'Visible','on')
end
```

B.1.3 REACTTEST.M

```
[pop,fs,nbit]=wavread('pop.wav');

N=5;
n=ceil(3*(rand(1,N)));
k=1;

T=timer('StartFcn','begintimer','StartDelay',n(k)+1,'TimerFcn',
't=cputime;wavplay(0.25*pop,fs);');
fi1=figure;
axis off
set(fi1,'Units','normalized');
set(fi1,'Position',[0.005,-0.1,0.99,1]);
set(fi1,'NumberTitle','off');
page1title=['Reaction Time Test']; set(fi1,'Name',page1title);

uptext1=text(0.10,0.9,'Did you hear two sounds?','FontSize',45);
uptext2=text(0.15,0.8,'Press START to begin and to play the first
sound','FontSize',20,'Color',[0.6 0.6 0],'FontWeight','bold');
uptext3=text(0.1,0.75,'Once you hear the second sound, press the
NOW button','FontSize',17);

start_button=uicontrol('Style','pushbutton','Units','normalized',
'BackgroundColor',[0 0.9 0],'FontSize',14,'Enable','on','FontWeight',
'bold','String','START','Position',[0.3,0.5,0.15,0.07],'CallBack',
'startreact');
detect_button=uicontrol('Style','pushbutton','Units','normalized',
'BackgroundColor',[0.6 0.6 0.6],'FontSize',14,'Enable','off',
'FontWeight','bold','String','NOW!','Position',[0.5,0.5,0.15,0.07],
'CallBack','nextpagereact');
test_button=uicontrol('Style','pushbutton','Units','normalized',
'BackgroundColor',[1 1 0.6],'FontSize',14,'Enable','on',
'FontWeight','bold','String','Sound Example','Position',
[0.4,0.4,0.15,0.07],'TooltipString','Click here to hear what
 the sound will be!','CallBack','wavplay(0.25*pop,fs)');
```

B.1.4 STARTREACT.M

```
delete(uptext2)
set(start_button,'BackgroundColor',[0.6 0.6 0.6],'Enable','off');
set(test_button,'Enable','off');
start(T);
```

B.1.5 BEGINTIMER.M

```
wavplay(0.25*pop,fs);
set(detect_button,'Enable','on','BackgroundColor',[1 1 0],'CallBack',
'nextpagereact');
```

B.1.6 NEXTPAGEREACT.M

```
reactiontime(k,1)=(cputime-t);

k=k+1;

if k<=N

uptext2=text(0.15,0.8,'Press START to begin the next iteration',
'FontSize',20,'Color',[0.6 0.6 0],'FontWeight','bold');

set(start_button,'BackgroundColor',[0 0.9 0],'Enable','on','String',
'START','Callback','startreact')
set(detect_button,'BackgroundColor',[0.6 0.6 0.6],'Enable','off',
'String','NOW!','Callback','nextpagereact')

else

    avgrctntime=sum(reactiontime)/N;
    delete(detect_button);
    delete(start_button);
    delete(test_button);
    delete(uptext1);
    delete(uptext3);
    figure(fi1);
    uptext1=text(0.15,0.9,'THANK YOU!','FontSize',45,
     'Color',[0 0 0.9]);
    uptext2=text(0.15,0.75,'You have finished the test!','FontSize',
     30,'Color',[0 0 0.9]);
    close all; clear pop;
    rampexamp;
end
```

B.1.7 RAMPEXAMP.M

```
noisetype=fix(3*rand);
fid=fopen([userid '_' int2str(noisetype+1) '.mat']);
if fid==-1
    if noisetype==0
        !"C:\Documents and Settings\Mike\Skrivbord\listening\
         noises\zerodb.wpl"&
    elseif noisetype==1
        !"C:\Documents and Settings\Mike\Skrivbord\listening\
         noises\fivedb.wpl"&
    elseif noisetype==2
        !"C:\Documents and Settings\Mike\Skrivbord\listening\
         noises\tendb.wpl"&
    end
else
    if noisetype==1
        !"C:\Documents and Settings\Mike\Skrivbord\listening\
         noises\tendb.wpl"&
        noisetype=2;
    else
        !"C:\Documents and Settings\Mike\Skrivbord\listening\
         noises\fivedb.wpl"&
        noisetype=1;
    end

end

attenuator=0.008;
soundtime=15;
cursam=1;
i=1;

[y_sound1,fs_play,nbit]=wavread('C:\Documents and Settings\Mike\
Skrivbord\listening\ramp\sounds\11000Hz\ramp11fund1.wav');
t=[0:length(y_sound1)-1]./fs_play;
tind=find(t<=soundtime);
y_sound1c=attenuator*y_sound1(tind,:);
clear y_sound1 t;
windowlength1=round(0.15*fs_play);
window1=log10((10/windowlength1).*[(windowlength1/10):
windowlength1]);
window1=fliplr(window1);
y_sound1c(length(tind)-length(window1)+1:length(tind))=
y_sound1c(length(tind)-length(window1)+1:length(tind))
.*[window1]';
clear tind;
y_sound1f=[zeros(2*fs_play,1);y_sound1c;zeros(windowlength1,1)];
clear y_sound1c window1;
y_sound1fdown=flipud(y_sound1f);

sound_upplay=audioplayer(y_sound1f,fs_play);
sound_downplay=audioplayer(y_sound1fdown,fs_play);
```

```
totsam=get(sound_upplay, 'TotalSamples');

fi1=figure;
axis off
set(fi1,'Units','normalized');
set(fi1,'Position',[0.005,-0.1,0.99,1]);
set(fi1,'NumberTitle','off');
page1title=['Welcome to the Ramp Test!']; set(fi1,'Name',
page1title);

uptext1=text(0.10,0.9,'Can you hear a change?','FontSize',45);
uptext2=text(0.15,0.8,'Press START to begin and play the sound',
'FontSize',20,'Color',[0.6 0.6 0],'FontWeight','bold');
uptext3=text(0.1,0.75,'If you hear a change press the one
button','FontSize',17);
uptext4=text(0.15,0.8,'PLEASE DO WHAT IS ASKED FOR IN THE TEXT
FIELD BELOW!','FontSize',20,'Color',[0.9 0 0],'FontWeight',
'bold','Visible','off');

% Callbacks
stopsound=['stop(sound_upplay);' 'stop(sound_downplay)'];

% Uicontrols

start_button=uicontrol('Style','pushbutton','Units','normalized',
'BackgroundColor',[0 0.9 0],'FontSize',14,'Enable','on',
'FontWeight','bold','String','START','Position',
[0.3,0.5,0.15,0.07],'CallBack','startbutton');
stop_button=uicontrol('Style','pushbutton','Units','normalized',
'BackgroundColor',[0.6 0.6 0.6],'FontSize',14,'Enable','off',
'FontWeight','bold','String','STOP','Position',
[0.3,0.4,0.15,0.07],'TooltipString','Click here to stop and
reset the current test sound','CallBack','stopbutton');

one_button=uicontrol('Style','pushbutton','Units','normalized',
'BackgroundColor',[0.6 0.6 0.6],'FontSize',14,'Enable','off',
'FontWeight','bold','String','One','Position',
[0.5,0.5,0.1,0.07],'CallBack','one');
two_button=uicontrol('Style','pushbutton','Units','normalized',
'BackgroundColor',[0.6 0.6 0.6],'FontSize',14,'Enable','off',
'FontWeight','bold','String','Two','Position',
[0.63,0.5,0.1,0.07],'CallBack','two');
three_button=uicontrol('Style','pushbutton','Units','normalized',
'BackgroundColor',[0.6 0.6 0.6],'FontSize',14,'Enable','off',
'FontWeight','bold','String','Three','Position',
[0.5,0.4,0.1,0.07],'CallBack','three');
four_button=uicontrol('Style','pushbutton','Units','normalized',
'BackgroundColor',[0.6 0.6 0.6],'FontSize',14,'Enable','off',
'FontWeight','bold','String','Four','Position',
[0.63,0.4,0.1,0.07],'CallBack','fourexamp');
```

B.1.8 RAMPTEST.M

```
load soundnames;
load userids;

iind0=size(soundnames); iind=iind0(1);
failvector=zeros(1,iind);
failvector_nofirst=zeros(1,iind);
i=1;
attenuator=0.0008;
impcursam(:,3)=[15 30 60 90 120 180 240 300 420 540];
nimpcursam(:,3)=[15 15 15 30 30 30 60 60 60 90 90 90];
cursec=impcursam;
ncursec=nimpcursam;

soundtimes=[15 30 60 90 120 180 240 300 420 540 15 15 15 30 30
 30 60 60 60 ];

[m,n]=sort(rand(1,iind));


[y_sound1,fs_play,nbit]=wavread([char(soundnames(n(i),1)),
char(soundnames(n(i),2))]);
t=[0:length(y_sound1)-1]./fs_play;
tind=find(t<=soundtimes(n(i)));

y_sound1c=attenuator*y_sound1(tind,:);
clear y_sound1 t;
windowlength1=round(0.15*fs_play);
window1=log10((10/windowlength1).*[(windowlength1/10)
:windowlength1]);
window1=fliplr(window1);
y_sound1c(length(tind)-length(window1)+1:length(tind))=
y_sound1c(length(tind)-length(window1)+1:length(tind)).*
[window1]';
clear tind;
y_sound1f=[zeros(2*fs_play,1);y_sound1c;zeros(windowlength1,1)];
clear y_sound1c window1;
y_sound1fdown=flipud(y_sound1f);

sound_upplay=audioplayer(y_sound1f,fs_play);
sound_downplay=audioplayer(y_sound1fdown,fs_play);

cursamup=1;
cursamdown=1;
totsam(i,1)=get(sound_upplay, 'TotalSamples');

fi1=figure;
axis off
set(fi1,'Units','normalized');
set(fi1,'Position',[0.005,-0.1,0.99,1]);
set(fi1,'NumberTitle','off');
page1title=['Page number ',int2str(i),' of ',int2str(iind)];
set(fi1,'Name',page1title);
```

```
uptext1=text(0.10,0.9,'Can you hear a change?','FontSize',45);
uptext2=text(0.15,0.8,'Press START to begin and play the sound',
'FontSize',20,'Color',[0.6 0.6 0],'FontWeight','bold');
uptext3=text(0.1,0.75,'If you hear a change press the one
button','FontSize',17);

% Callbacks

stopsound=['stop(sound_upplay);' 'stop(sound_downplay)'];

% Uicontrols

start_button=uicontrol('Style','pushbutton','Units','normalized',
'BackgroundColor',[0 0.9 0],'FontSize',14,'Enable','on',
'FontWeight','bold','String','START','Position',
[0.3,0.5,0.15,0.07],'CallBack','startbutton');
stop_button=uicontrol('Style','pushbutton','Units','normalized',
'BackgroundColor',[0.6 0.6 0.6],'FontSize',14,'Enable','off',
'FontWeight','bold','String','STOP','Position',
[0.3,0.4,0.15,0.07],'TooltipString','Click here to stop and
reset the current test sound','CallBack','stopbutton');

one_button=uicontrol('Style','pushbutton','Units','normalized',
'BackgroundColor',[0.6 0.6 0.6],'FontSize',14,'Enable','off',
'FontWeight','bold','String','One','Position',
[0.5,0.5,0.1,0.07],'CallBack','one');
two_button=uicontrol('Style','pushbutton','Units','normalized',
'BackgroundColor',[0.6 0.6 0.6],'FontSize',14,'Enable','off',
'FontWeight','bold','String','Two','Position',
[0.63,0.5,0.1,0.07],'CallBack','two');
three_button=uicontrol('Style','pushbutton','Units','normalized',
'BackgroundColor',[0.6 0.6 0.6],'FontSize',14,'Enable','off',
'FontWeight','bold','String','Three','Position',
[0.5,0.4,0.1,0.07],'CallBack','three');
four_button=uicontrol('Style','pushbutton','Units','normalized',
'BackgroundColor',[0.6 0.6 0.6],'FontSize',14,'Enable','off',
'FontWeight','bold','String','Four','Position',
[0.63,0.4,0.1,0.07],'CallBack','four');
```

B.1.9 STARTBUTTON.M

```
play(sound_upplay)
delete(uptext2)

set(start_button,'BackgroundColor',[0.6 0.6 0.6],'Enable','off')

set(one_button,'Enable','on','BackgroundColor',[1 1 0],
'CallBack','one');

set(stop_button,'Enable','on','BackgroundColor',[1 0 0])
```

B.1.10 STOPBUTTON.M

```
eval(stopsound);
set(one_button,'Enable','off','BackgroundColor',[0.6 0.6 0.6]);
set(two_button,'Enable','off','BackgroundColor',[0.6 0.6 0.6]);
set(three_button,'Enable','off','BackgroundColor',[0.6 0.6 0.6]);
set(four_button,'Enable','off','BackgroundColor',[0.6 0.6 0.6]);

uptext2=text(0.15,0.8,'Press START AGAIN to begin and to play
the sound again','FontSize',20,'Color',[0.6 0.6 0],
'FontWeight','bold');
set(start_button,'String','START AGAIN','BackgroundColor',
[0 0.9 0],'Enable','on','CallBack','startbutton');
```

B.1.11 ONE.M

```
cursamup(i,1)=get(sound_upplay, 'CurrentSample');
eval(stopsound);


play(sound_downplay); %% to play from the end

delete(uptext3);
uptext3=text(0.1,0.75,'If you hear a change press the two
button','FontSize',17);

set(one_button,'BackgroundColor',[0.6 0.6 0.6],'Enable','off');
set(two_button,'BackgroundColor',[1 1 0],'Enable','on');
```

B.1.12 TWO.M

```
cursamdown(i,1)=get(sound_downplay, 'CurrentSample');

eval(stopsound);

play(sound_upplay);

delete(uptext3);
uptext3=text(0.1,0.75,'If you hear a change press the three
button','FontSize',17);

set(two_button,'BackgroundColor',[0.6 0.6 0.6],'Enable','off');
set(three_button,'BackgroundColor',[1 1 0],'Enable','on');
```

B.1.13 THREE.M

```
cursamup(i,2)=get(sound_upplay, 'CurrentSample');
eval(stopsound);

play(sound_downplay); %% to play from the end

delete(uptext3);
uptext3=text(0.1,0.75,'If you hear a change press the four
button','FontSize',17);

set(three_button,'BackgroundColor',[0.6 0.6 0.6],'Enable','off');
set(four_button,'BackgroundColor',[1 1 0],'Enable','on');
```

B.1.14 FOUR.M

```
cursamdown(i,2)=get(sound_downplay, 'CurrentSample');

eval(stopsound);

set(four_button,'Enable','off','BackgroundColor',[0.6 0.6 0.6]);

set(one_button,'Enable','off','BackgroundColor',[0.6 0.6 0.6]);
set(stop_button,'Enable','off','BackgroundColor',[0.6 0.6 0.6]);

nextpage;
```

B.1.15 FOUREXAMP.M

```
cursam(i,4)=(totsam(i,1)-get(sound_upplay, 'CurrentSample'));
eval(stopsound);

set(four_button,'Enable','off','BackgroundColor',[0.6 0.6 0.6]);

set(one_button,'Enable','off','BackgroundColor',[0.6 0.6 0.6]);
set(stop_button,'Enable','off','BackgroundColor',[0.6 0.6 0.6]);

close all;
ramptest;
```

B.1.16 NEXTPAGE.M

```
if n(i)<=10
    impcursam(n(i),1:2)=cursamup(i,:);
    impcursam(n(i),4:5)=cursamdown(i,:);
```

```
    for j=1:2
        cursec(n(i),j)=impcursam(n(i),j)/fs_play;
    end
    for j=3:4
        cursec(n(i),j+1)=impcursam(n(i),j+1)/fs_play;
    end
else
    nimpcursam(n(i)-10,1:2)=cursamup(i,:);
    nimpcursam(n(i)-10,4:5)=cursamdown(i,:);
    for j=1:2
        ncursec(n(i)-10,j)=nimpcursam(n(i)-10,j)/fs_play;
    end
    for j=3:4
        ncursec(n(i)-10,j+1)=nimpcursam(n(i)-10,j+1)/fs_play;
    end
end

i=i+1; nofirst=0; yesfirst=0;

if i<=iind

page1title=['Page number ',int2str(i),' of ',int2str(iind)];
set(fi1,'Name',page1title);
uptext2=text(0.15,0.8,'Press START to begin and play the sound',
'FontSize',20,'Color',[0.6 0.6 0],'FontWeight','bold');
delete(uptext3);
uptext3=text(0.1,0.75,'If you hear a change press the one
button','FontSize',17);

set(start_button,'BackgroundColor',[0 0.9 0],'Enable','on',
'String','START','Callback','startbutton')

% MONO VERSION
[y_sound1,fs_play,nbit]=wavread([char(soundnames(n(i),1)),
char(soundnames(n(i),2))]);
t=[0:length(y_sound1)-1]./fs_play;
tind=find(t<=soundtimes(n(i)));
y_sound1c=attenuator*y_sound1(tind,:);
clear y_sound1 t;
windowlength1=round(0.15*fs_play);
window1=log10((10/windowlength1).*[(windowlength1/10):
windowlength1]);
window1=fliplr(window1);
y_sound1c(length(tind)-length(window1)+1:length(tind))=y_sound1c
(length(tind)-length(window1)+1:length(tind)).*[window1]';
clear tind;
y_sound1f=[zeros(2*fs_play,1);y_sound1c;zeros(windowlength1,1)];
clear y_sound1c window1;

sound_upplay=audioplayer(y_sound1f,fs_play);

totsam(i,1)=get(sound_upplay, 'TotalSamples');

else
    soundorder=soundnames(n,:);
```

```
    save([userid '_' int2str(noisetype+1)], 'n', 'impcursam',
      'nimpcursam', 'cursec', 'ncursec', 'avgrctntime',
      'reactiontime');
    if testiteration==1
        testiteration=2;
        rampexamp;
    else
        delete(one_button);
        delete(two_button);
        delete(three_button);
        delete(four_button);
        delete(stop_button);
        delete(start_button);
        delete(uptext1);
        delete(uptext3);
        delete(uptext4);
        figure(fi1);
        uptext1=text(0.15,0.9,'THANK YOU!','FontSize',45,
         'Color',[0 0 0.9]);
        uptext1=text(0.15,0.75,'You have finished the test!',
        'FontSize',30,'Color',[0 0 0.9]);
    end
end
```

B.2 Other Code

B.2.1 Phase Scrambler

```
clear all;
[buller,fs,nbits]=wavread('C:\Documents and Settings\CRAG\
Skrivbord\Mike\Intel\noise-making\Binaural_BG_quiet.wav');

nfft = length(buller(:,1));
N = [1:nfft/2+1];
hanni(1:(fs/10),1) = window(@flattopwin,fs/10);
w = [hanni(1:(fs/20),1); ones(nfft-(fs/10)-1,1);
hanni((fs/20):end,1)];

if size(buller(:,1),1) == 1, buller(:,1) = buller(:,1).'; end;

F(:,:) = fft(buller(1:nfft,:),nfft);
Fhalf(:,:) = F(1:nfft/2+1,:);
Fhalf(:,:) = (rand(size(angle(Fhalf(:,:))))*2*pi).*
abs(Fhalf(:,:));

F(:,:) = [Fhalf(:,:); conj(Fhalf(1:(nfft/2)-1,:))];
buller(:,:) = w.*real(ifft(F(:,:)));
wavwrite(buller,fs,nbits,'C:\Documents and Settings\CRAG\
Skrivbord\Mike\Intel\noise-making\BG_quiet_randomizedabs.wav');
```

CHALMERS UNIVERSITY OF TECHNOLOGY
SE 412 96 Göteborg, Sweden
Phone: + 46 - (0)31 772 10 00
Web: www.chalmers.se