uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Abimbola Y. Cole

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.A.Sc.(Electrical Engineering)

GRADE / DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Adaptive User Specific Learning for Environment Sensitive Hearing Aids

TITRE DE LA THÈSE / TITLE OF THESIS

T. Aboulsnar

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

C. Giguere & W. Guereb

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

H. Dajani

R. Goubran

Gary W. Slater

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

Adaptive User Specific Learning for Environment Sensitive Hearing Aids

By

Abimbola Y Cole

A thesis submitted to
The Faculty of Graduate and Postdoctoral Studies
in partial fulfillment of the requirements of the degree of
Master of Applied Sciences
in Electrical Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering
School of Information Technology and Engineering
University of Ottawa

Your file Votre référence
ISBN: 978-0-494-51832-8
Our file Notre référence
ISBN: 978-0-494-51832-8

Abstract

One of the main complaints of hearing aid users is the difficulty of and need for frequent adjustments they have to make to the device. Typically, volume control and other settings prescribed during the hearing aid fitting process still need to be fine-tuned in real life to meet the specific varying needs of the user in different environments. Recently, learning algorithms have been introduced to minimize these problems. By taking a weighted average of the past adjustments, the device is able to adapt to the preferred setting for each environment presented. Typically, fixed learning algorithms are used for all users and environments. In reality, users have different behaviors and the optimum learning time is expected to vary over users and environments.

In this thesis, we study the potential for user-specific adaptive learning of the user settings. Profiles of various types of user behaviors were generated and the optimum time constant for learning user preferences was determined in each case using fixed exponential smoothing. We show that the performance of the algorithm is clearly dependant on the user profile and no single fixed time constant is optimum for all users or situations. Three adaptive exponential smoothing methods were then evaluated. Four performance measures are proposed to evaluate these methods based on the users' preferred setting when they re-enter the same environment. For some user profiles, adaptive methods were found to perform as well as the optimum time constant when learning user preferences for volume control without having prior knowledge of the users' behavior. In particular it is shown that the method by W.M Chow (*Journal of Industrial Engineering*, 16, pp. 314-317, 1965) can be tuned to perform as well as the optimum time constant for a given user.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to take this opportunity to thank my thesis supervisors Dr. Tyseer Aboulnasr, Dr. Christian Giguere and Dr. Wail Gueaieb for their guidance and support during the time I spent as their student. I am grateful for all the advice and help they provided me during the course of this program and for their genuine concern for my development as a graduate student. Without them none of this work would be possible.

Finally, I would like to thank my friends and family for their continuous encouragement and prayers. In particular my parents, for always believing I can accomplish anything I set my mind to.

Chapter 1 Introduction

1.1 Background

The number of people who have some kind of hearing loss and need the assistance of a hearing aid is much larger than the number of people that actually consult audiologists to complain, get fitted for a hearing aid and use the device [1, 2]. This is in part due to the stigma associated with wearing such a device, which ranges from being labeled as "old" in adults to not being considered "normal" amongst younger people [1]. There is also the issue of the negative perception of hearing aid wearers termed the "hearing aid effect" [1], where they are evaluated negatively on areas such as intelligence, overall achievement and appearance by non-wearers because of the presence of a hearing device [1]. It has also been widely reported that many users are not satisfied with their hearing aids and would rather put them away. Some reasons given are the presence of background noise, no added benefit perceived, discomfort as well as the difficulty and frequency in which the volume control of the hearing aid has to be adjusted [3, 4].

In the hearing aid industry, a lot of research has gone into closing the gap between people who suffer from hearing loss and those who seek treatment, as well as improving the wearing experience for those who decide to use the device. Approaches used range from better noise reduction methods to minimizing the size of the hearing aid in order to make it less visible. The focus of this thesis is on learning the users' hearing aid control preferences so as to minimize the amount of adjustments the user makes to the device. This in turn draws less attention to the fact that they are wearing an assistive device.

During hearing aid fitting, the required gains programmed into the device are a result of very general prescriptive procedures [5] and still require fine-tuning after the user has worn the hearing aid in their daily life. The fine-tuning is done by the audiologist at follow-up visits and is based on comments/feedback from the hearing aid wearer. However, the comments are sometimes hard to interpret making the process harder for the clinician and frustrating for the patient.

The prescriptive procedures assume that the same gains and hearing parameters are suitable in all acoustic environments. However, it has been found that this is not the case and that different environments indeed need different settings [4]. This finding motivated the implementation of customized listening programs for different environments. These programs contain predetermined settings that are believed to optimize hearing in that environment. The user has the option to manually load the program they feel corresponds to the situation they are in.

Research in recent years has confirmed that users also have individual preferences in different environments [6, 7]. Also a case study carried out at the National Health Service (NHS) audiology clinic in the UK has established that most hearing aid wearers, both new and experienced, would prefer to have their devices equipped with a manual volume control [8]. Patients of the clinic where asked to wear their hearing aid with the manual volume control activated then deactivated. By the end of this period, it was found that most (73%) of the participants preferred to have the manual volume control activated. Also a higher percentage of experienced users (88.9%) than new hearing aid wearers (60%) preferred this feature [8]. A survey by MarkeTrak [47] found similar results except that in this study, users where asked if they were satisfied with their current device and if they felt the need for a manual volume control. Unlike in the first study, they did not have an opportunity to choose between having a manual volume control or not, but never-the-less most participants felt having one would be beneficial.

The concept of trainable hearing aids has also been proposed [7], they are devices which are able to learn from changes the user makes to the control settings in each individual environment that it identifies. This applies in particular to the volume control. When an environment is presented to the hearing aid, the volume control is automatically set to a weighted average of past volume settings for that specific environment. The learning rate is predetermined as an average one for all users. It is hoped that this learning will minimize user interventions over time.

1.2 Thesis Objective

Currently, when learning the user preferences, a fixed learning algorithm or smoothing constant is typically used in the algorithm that is programmed into the hearing aid [9]. Since the device is to be used by a wide range of users, the question is raised whether a fixed learning time constant would be best for all users despite their varying behaviors. In this thesis, we address this important question and investigate if users have the same "optimum" time constant to learn their preferences quickly and accurately. If indeed the optimum time constant is dependant on the user behavior, then it might be worthwhile to adapt the learning parameter to the user behavior. Further, a given individual may have different behaviors in different environments and their behavior may change over time. In this research, we investigate several techniques to adapt the learning constant to the user behavior to identify the potential improvement and suggest an approach to achieve it.

The objective of this thesis is to learn what control settings the user prefers in different environments with the goal of being able to predict what the user would like when they re-enter these environments, for all adjustable user controls and in particular the volume control. Over the course of the day the user switches between different environments. The device should be able to recommend the best initial setting when the user returns to a specific environment (the settings can still be over ridden by the user at any time).

1.3 Contribution

This thesis contributes to the research done on hearing aids in the following ways:

- Showing that the optimum time constant for learning volume control preferences is dependant on the behavior of the user and hence justifying the need for an adaptive time constant.
- Confirming that the use of a fixed time constant within a margin of 10 – 20 hours in some devices is reasonable if a fixed time constant is to be used in the learning algorithm.
- Application of time-series forecasting techniques in learning hearing aid settings.

- Investigation of the performance of three adaptive exponential smoothing methods in learning hearing aid settings.
- Investigation of the conditions in which adaptive techniques would result in superior learning.

1.4 Thesis Layout

The rest of this document is laid out as follows:

Chapter 2: Hearing Aids Technology

This chapter explains briefly the basic principles of hearing aids, outlines the hearing aid fitting process and some of the challenges experienced. Also a review of the state of the industry in the area of learning user control preferences is provided.

Chapter 3 Learning Algorithms

This chapter first introduces Exponential Smoothing using a fixed time constant, then details the adaptive methods considered in this thesis. Four algorithms in which the time constant adapts based on the settings observed by the hearing aid are presented.

Chapter 4 Simulations and Performance Measures

This chapter presents the behavioral data that was generated for the purpose of testing the learning algorithms. Assumptions made in the simulations are outlined. The measures used to evaluate the algorithms are defined and described. Exponential Smoothing using a fixed time is analyzed and the need for using an adaptive time constant is justified.

Chapter 5 Evaluation of Algorithms

In this chapter, three adaptive methods are modified to suite the requirements of this thesis and then they are evaluated. The optimum parameter(s) for each method are determined and used to learn the preferences for a wide range of behaviors. The results are presented and discussed.

Chapter 6 Conclusions and Future Directions

This chapter summarizes the findings in this work and outlines recommendation on the next steps to take based on these findings.

Chapter 2 Hearing Aid Technology

The ear is made up of three parts, the outer ear, middle ear and inner ear. A diagram of the human auditory system is shown in figure 2.1 [10]. The outer ear is made up of the pinna which is the visible part of the ear and the ear canal (or external auditory meatus). The function of the outer ear is to amplify sound waves in the front of the ear and to aid in sound localization which is the ability to detect the origin of a sound in space. The middle ear is an air-filled cavity containing three tiny bones, the hammer, incus and stape. These bones are referred to as the ossicles. The hammer is attached to the ear drum while the foot of the stape is embedded in the inner ear at an aperture called the oval window. The ear drum (or tympanic membrane) can be found between the outer and middle ear.

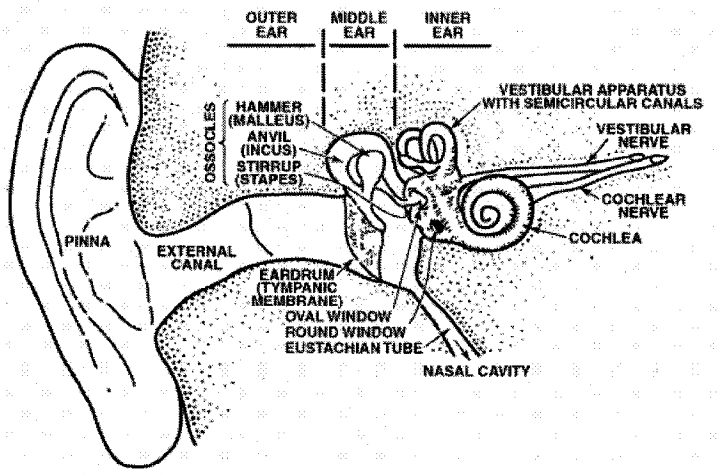


Diagram from http://cobweb.ecn.purdue.edu/~ee649/notes/figures/ear.gif (March 2008)[10]

Figure 2.1: Human Auditory System

The main component of the inner ear is the cochlea, which is a fluid filled snail-shaped organ. The oval window is attached to the cochlea. The function of the ossicles is to transfer the sound energy by matching the impedance of the ear canal to the higher impedance of the cochlea fluids thus reducing reflection of the sound energy [11]. Once in the cochlea the sounds are compressed to fit the 120dB dynamic range of hearing. By analyzing the sound frequencies, the cochlea is able to convert the signal into electrical pulses which are then

transmitted to the many nerve endings that are connected to the brain. It is in the brain that the impulses are finally decoded and the sounds can be heard and identified [12].

2.1 Hearing Loss

Hearing loss is caused by partial or complete damage to parts of the human auditory system. There are three main types of hearing loss: conductive, sensorineural and mixed hearing loss [13].

Conductive hearing loss is related to damage in the outer or middle ear. It can be caused by inflammation of the skin of the outer ear due to bacterial infection since this part of the ear is the most exposed to external conditions [12]. Another cause is the rupturing of the ear drum from exposure to excessive noise or perforation by obstructive objects. This perforation also opens the middle ear to external conditions which could cause damage. The above mentioned conditions cause certain frequencies of sounds to be attenuated. Conductive hearing loss can also result from a condition known as otosclerosis which is an abnormal growth of the middle ear bones. The ossicular bones become hardened and are not able to perform their impedance matching function. People with this type of hearing loss will tend to speak lower and softer as their voices appear to sound louder to them [13]. They are also able to tolerate loud sounds [12]. Some of the damage caused by conductive hearing loss can be corrected through surgery. The perforation in the ear drum can be patched with muscle tissue while the ossicular bones can be replaced through a transplant or artificial replacement [12]. Hearing aids are also able to correct this hearing loss by amplifying the signal such that it gets through the middle ear as if no damage occurred [11]. The device does this by applying gains at the affected frequencies.

Sensorineural hearing loss which is the most common kind of hearing loss is caused by damage to the hair cells of the cochlea or the auditory nerves. This can usually be due to old age or excessive exposure to noise [11]. It can also occur in children due to diseases like rubella or measles [12]. One of the symptoms of this type of hearing loss is that affected individuals will often speak louder as they cannot hear their own voice [12]. People with this

type of hearing loss suffer a lot of interference from the presence of noise. This is particularly a problem if the noise is at a frequency that is close to that of the desired signal [5]. They are also not able to hear sounds that are quickly followed by a loud noise due to temporal masking. Finally, the dynamic range of the individual is reduced, that is the range of levels that can be heard is decreased. Weak sounds cannot be heard and intense sounds are too loud and cause discomfort [5]. There is no medication to cure sensorineural hearing loss and surgery does very little to help. Currently the most effective solution is the use of hearing aids to compensate for the hearing loss [14]. The hearing aid must process the signal to restore it to what would have been received if there was no damage [11]. Typically, the device gives more amplification to soft signals to raise them to the residual dynamic range and applies less gain to louder signals to reduce discomfort.

Sometimes a mixture of both conductive and sensorineural hearing losses occur and the affected individual will suffer from both kinds of symptoms. This individual is said to be suffering from a mixed hearing loss.

2.2 Hearing Aid Overview

Hearing aids are made up of some basic components such as microphones, amplifiers, filters and receivers. The microphone collects the sound energy from the environment and converts it to electrical signals. Filters are used to amplify the input at different frequencies or to split up the signal into different bands. An amplifier is then used to increase the amplitude of the signal. After all the processing is finished, the signal is transmitted to the receiver which converts it into an acoustic signal which can be heard. The hearing aid is equipped with a battery to supply power for all the required processing. When the signal passes through the entire chain of components, one after the other, the hearing aid is said to be a single band

device or to have a serial structure [5]. An example is shown in figure 2.2 below.

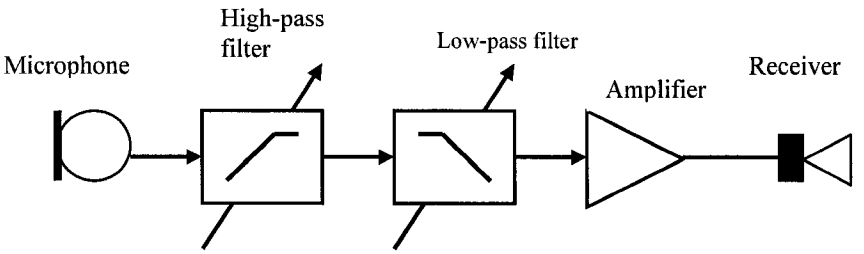


Figure 2.2: Single band hearing aid example

In the case where the signal is split into various frequency bands and amplified separately it is called a multi-channel hearing aid. Figure 2.3 shows an example of a 7 channel hearing device. The amplified signals in each band are summed up and transmitted to the hearing aid receiver.

In the past, these components have been made from analog parts like resistors, transistors etc. Fully digital hearing aids are presently the norm and are able to carry out complex signal processing.

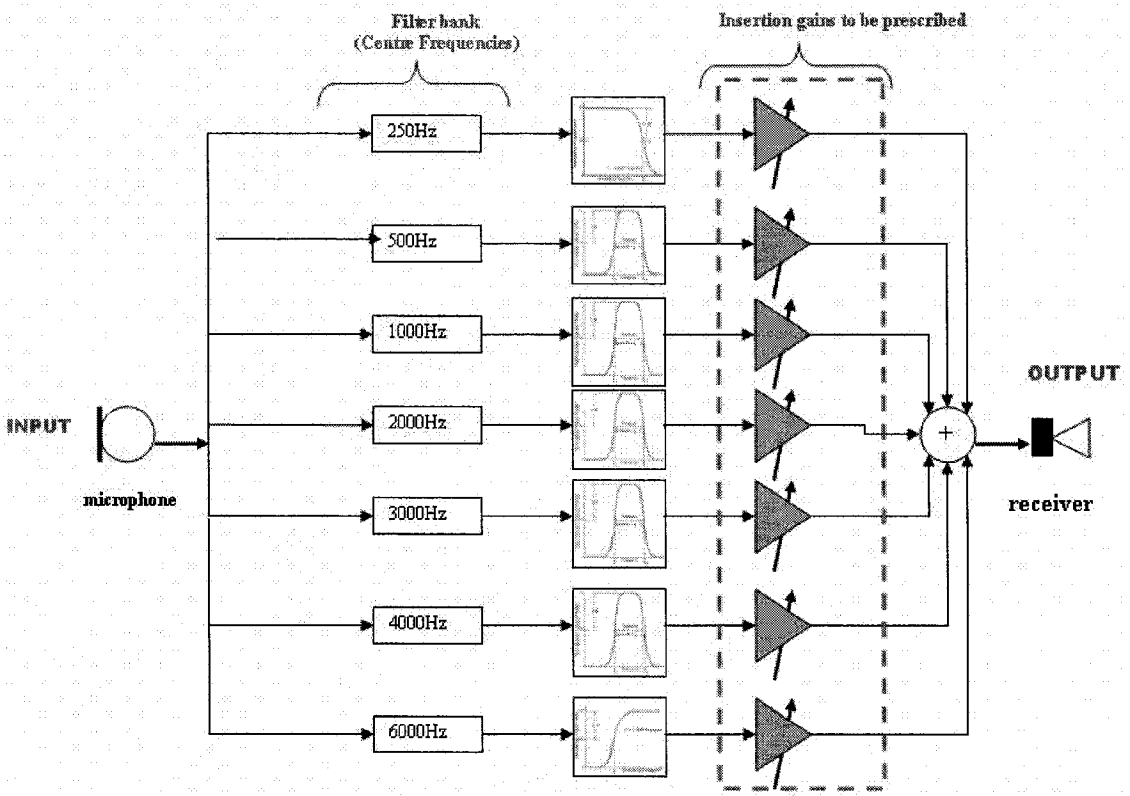


Figure 2.3: Seven band hearing aid example

Hearing aids can also be classified according to their style or physical configuration. The largest is the Behind-The-Ear (BTE) hearing aid; its components are kept in a plastic casing that fits behind the pinna. A tube connects the casing to a mould that is placed at the entry of the air canal. The tube carries the amplified sound from the receiver into the ear. Next, in descending size are In-The-Ear (ITE) hearing aids which fit in the concha followed by In-The-Canal (ITC) hearing aids which fit mostly in the ear canal and the smallest devices are the Completely-In-the-Canal (CIC) which fit directly in the ear canal [5].

2.3 Hearing Aid Fitting Process

The initial stage of the hearing aid fitting process is the administration of diagnostic tests to identify if the individual has some form of hearing loss and is a candidate for hearing aid fitting. The most basic test is the Pure Tone Test which measures the minimum sound level or absolute threshold that a person can barely detect in quiet at various frequencies [13]. The results are plotted as an audiogram [48] (see figure 2.4) which provides a visual representation for the audiologist.

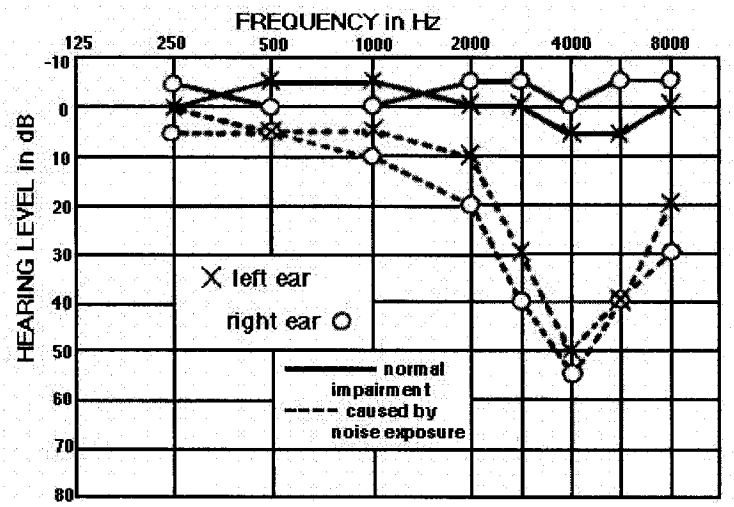


Diagram from: http://www.sfu.ca/sonic-studio/handbook/Graphics/Audiogram.gif (March 2008)[48]

Figure 2.4: Audiogram used to plot hearing thresholds

The tone frequencies (Hertz) sent through headphones to the patient during the administration of the Pure Tone test are plotted on the x-axis. The vertical axis is the corresponding threshold of hearing for each frequency in dB HL. Test data from the left ear

is recorded with an **x** and a circle denotes recordings from the right ear. Hearing thresholds are usually measured at 250Hz, 500Hz, 1000Hz, 2000Hz, 4000Hz and 8000Hz. In figure 2.4, the plots in solid lines are the test results from a person with normal hearing. For an average normal hearing person, hearing thresholds of 0 dB HL are expected at every test frequency. However, thresholds in the range of 0-25dB HL can still be considered normal [13]. The plots in dashed lines are for an individual who is losing hearing at higher frequencies due to prolonged exposure to noise. Once the individual has been identified as a candidate for a hearing aid, the next step is to prescribe the amount of gain and other configurations the device would need in order to compensate for their hearing loss.

The amount of amplification required to compensate for the type of hearing loss can then be computed from the thresholds and other measures in already established formulas (or prescriptive procedures) to determine the required insertion gain at each frequency and level. The gains calculated are usually referred to as insertion gains. Many hearing aid manufacturers often have their own proprietary fitting algorithms for their devices. There are two main types of gain prescription procedures, linear and non-linear.

Linear amplification procedures are calculated based on the hearing thresholds and provide the same gains for all input sound levels up to a maximum amplification level. The gains are different from frequency to frequency depending on the hearing loss profile. Two popular methods of this kind are:

Prescription of Gain and Output (POGO): This procedure inserts a gain that is half the hearing loss at the tested frequency plus a constant k_i that depends on frequency.

National Acoustic Laboratories – Revised (NAL-R): This procedure calculates the insertion gains so that speech frequencies appear to sound at the same level [15], with the aim to maximize speech intelligibility. The steps involved are outlined below:

1) Calculate the average of the hearing thresholds at 500, 1000 and 2000 Hz.
2) The value calculated above is multiplied by 0.15.
3) The insertion gain is calculated as the sum of the values in 1), 2) and a constant

k_i that depends on frequency.

Both the above methods have been revised over time to accommodate individuals with severe hearing loss i.e. thresholds greater than 60dB.

Non-Linear amplification procedures unlike the linear ones have a different gain-frequency response for different sound levels. Two examples of this type of procedure are discussed below:

National Acoustic Laboratories – Non-Linear 1 (NAL-NL1): Like the linear method created by this group, the objective is still to maximize speech intelligibility but maintains the overall loudness at a level that would be comfortable for a person with normal hearing. The creators of this procedure found the optimum insertion gains by analyzing 51 different hearing loss profiles and taking into consideration metrics such as Articulation Indexes and loudness algorithms [16]. The computations are very complex and are proprietary information. Usually it is provided in a software format to the audiologist along with instructions on how to use it.

FIG6: This non-linear prescriptive method was created by Killion & Fikret-Pasa [5] and is named after figure 6 of the paper they authored outlining their procedure. The method factors in the sound level of the input signal in particular sounds at 40, 65 and 95dB SPL. The primary objective of this method is to normalize loudness. Table 2.1 summarizes the calculations for this method. The insertions gains for input levels outside the three targets are obtained by linear interpolation [5].

Input Level (dB SPL)	40	65	95
Hearing Threshold at frequency I (dB HL)			
$H_i < 20$	$IG_i = 0$	$IG_i = 0$	
$20 \leq H_i \leq 60$	$IG_i = H_i - 20$	$IG_i = 0.6(H_i - 20)$	
$H_i < 40$			$IG_i = 0$
$H_i > 40$			$IG_i = 0.1\ (H_i - 40)^{1.4}$
$H_i > 60$	$IG_i = O.5*H_i + 10$	$IG_i = 0.8H_i - 23$	

Table 2.1: Summary of FIG6 Non-Linear Prescriptive Procedure

Irrespective of the fitting method, the patient usually wears the hearing aid for a trial period and comes back into the clinic for a follow-up visit where they give feedback on the sound quality of their device. The audiologist in turn tunes the prescribed gains based on feedback from the patient. This fine-tuning is an on-going process of hearing aid fitting [5, 7, and 14]. Some fine-tuning strategies employed by clinicians are discussed later in this thesis as well as some inadequacies in the fitting process. Recent improvements to the process are also outlined.

In the next section, some of the complaints of hearing aid users are highlighted and how the industry has attempted to address these issues is discussed.

2.4 Advances in Hearing Aids

A lot of innovation has gone into the development of hearing aids to improve the users' listening experience and hearing aid prescription. Some areas that have been addressed are discussed next.

2.4.1 Acoustic Feedback Cancellation

The presence of a "howling" or "whistling" sound is one of the top complaints of hearing aid wearers [3]. Acoustic feedback is responsible for this effect and is caused when the hearing aid output signal leaks out of the ear canal, is picked up by the microphone and re-amplified by the hearing aid in a positive feedback loop. This occurs at high gains and reduces the dynamic range of the device [17], figure 2.5 [18] below illustrates this problem.

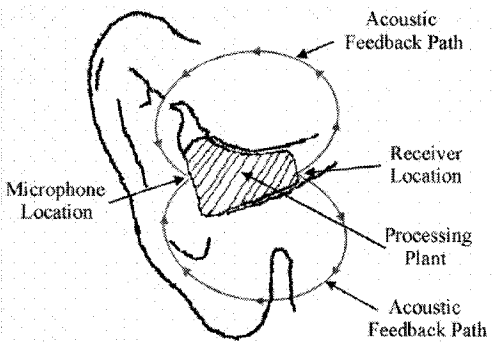


Diagram from: http://www.ee.ucla.edu/~spapl/rod/aeptf.htm (March 2008)[18]

Figure 2.5: Acoustic Feedback Path in Hearing Aids

Acoustic feedback is more prevalent at higher volumes and typically the user will feel the need to adjust the volume control. To reduce the number of interventions from the user, it is therefore necessary to minimize this howling effect. It has been shown that acoustic feedback can be suppressed through the use of adaptive filters [19-21]. This is done by modeling the feedback signal through the use of adaptive filters and subtracting it before it is amplified. One way is through non-continuous adaptation where a training sequence of the noise is presented to the hearing aid periodically. A howling detector is also needed to recognize the occurrence of the howl and is used in the cancellation which takes place during periods that are quiet. If the detector is not very good then the acoustic feedback will not be cancelled properly. This method interferes with the users' listening and reduces the

overall SNR [21]. The preferred method for eliminating acoustic feedback is the continuous one. Here adaptation is done all the time and is not limited to periods of silence. Adaptation is also solely based on the input signal and does not require an interfering training sequence [21]. Figure 2.6 below shows how the acoustic feedback can be modeled in hearing aids. The adaptive filter estimates the feedback loop which is in turn subtracted from the input to the hearing aid.

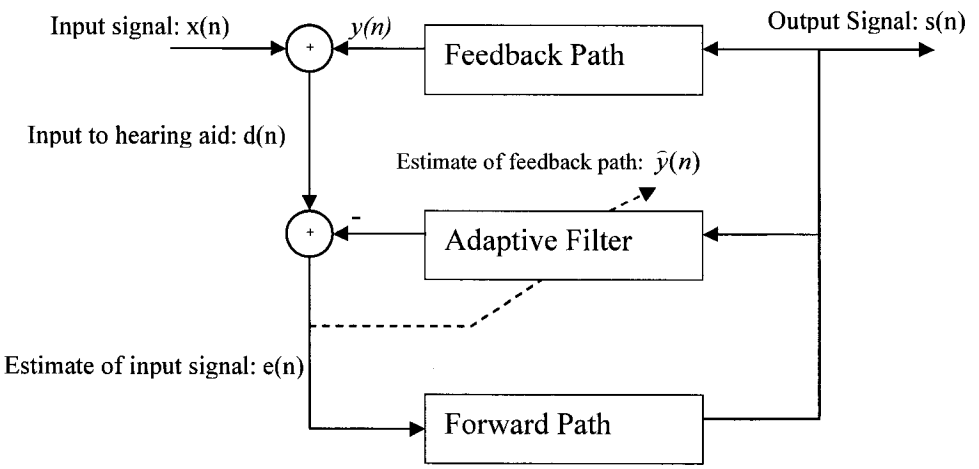


Figure 2.6: Model of Acoustic Feedback Cancellation in Hearing Aids

However, the issue with this approach is that the input signal and the output signal are correlated and results in misconvergence of the adaptive filter coefficients [17] and a poor feedback cancellation. The two signals are usually decorrelated by including a delay in the model usually in the forward path [20] or in the cancellation path [21].

2.4.2 Directional Microphones

One of the complaints of hearing aid users is a reduced ability to understand speech in situations with a lot of noise [22]. Directional microphones can be used to help improve hearing in some cases. The underlying assumption with the use of directional microphones in hearing aids is that the target/desired signal is often in the frontal position i.e. a person will usually face the source while noise may be distributed around the user. Directional

microphones focusing on the frontal direction will improve signal-to-noise ratio. Hence they may help to improve speech intelligibility in noisy environments.

The performance of these microphones is evaluated by a Directivity Index (DI). The DI is a measure of the signal-to-noise ratio (SNR) for a signal coming from the front direction and noise coming from all other directions (diffuse signals). To measure how well the directional microphone improves speech intelligibility an Articulation Index (AI) measure is often used along with the DI. The AI-DI is a weighted average of the DI across frequencies that affect speech intelligibility.

Behind-the-ear (BTE) hearing aids are usually equipped with two omni-directional microphones separated by a set distance (there is a distance restriction due to the small size of the hearing aid, so it is usually around 1cm). The output signals of the two microphones are differentially processed in a first order differential array [22]. By introducing a delay to the output of the rear microphone and subtracting it from the signal picked up by the front microphone, a directivity pattern can be achieved. The pattern achieved is dependant on the ratio of the internal delay and the external one created by the physical distance between the two microphones.

An improvement of the above method is the differential processing of signals from three microphones in a combination of first and second order directional processing. This was found to improve the AI-DI by 2dB in many listening conditions [22].

The next development in this area is the design of algorithms that can handle the desired source being in any direction i.e. no longer restricts the use of the directional microphones to cases where the source is in the front. Also, in reverberant environments, the desired sound and noise reflections coming in different directions reduces the benefits of directional microphones in this type of environment. Therefore since there are some situations where the use of directional microphones is not desirable, the user should have the ability to control this feature and switch it on or off as needed.

2.4.3 Noise Reduction

While directional microphone arrays are very helpful in removing noise from directions other than that of the speaker, it is difficult to use multiple microphone arrays or directional microphones [22] in ITC hearing aids due to size restrictions and reduction of the free sound field since the device is placed in the ear canal. It is therefore necessary to develop robust signal processing techniques to perform noise reduction with one microphone.

Background noise makes speech less intelligible and causes discomfort for hearing aid wearers. Hearing aids have mostly tried to tackle the latter issue. The noise consists mostly of low frequencies which contribute mainly to its loudness [5]. To reduce this discomfort, one approach is to minimize the gain at low frequencies especially in environments with increased noise levels [5]. The gain is based on the signal-to-noise ratio at each frequency, giving smaller gain if the SNR is poor. This technique thus reduces discomfort only when the signal and noise are in different frequency segments.

In cases where the speech is in the same frequency band as the noise, a Wiener filter based technique [22] can be used to reduce the noise. The gain at each frequency of the filter is the desired signal power spectral densities divided by the sum of the signal and noise power spectral densities [22]. The challenge however is that the known spectrum is a combination of both the desired signal and noise but what is required is the spectrum of only the noise or only the desired signal. For a desired signal which is just speech, its presence can be detected because of its distinct temporal envelope [5]. A non-speech detector can be used to measure the average spectrum when there are pauses in the input signal where speech's temporal envelope is not present. Once the noise spectrum is measured, it can be subtracted from the input signal power spectral density to get that of the desired signal alone. These values are then used in the gain calculation. Spectral Subtraction which is similar to the Wiener filter approach, is another noise reduction method used in hearing aids. The magnitude of the noise spectrum is subtracted from the magnitude of the noise plus speech spectrum [5]. Again, a non-speech detector can be used to detect periods where there is no speech and the magnitude of just the noise can be measured.

The issue with the Wiener filter and Spectral Subtraction methods is that the estimates of power spectral densities or spectrum magnitude measured are just for a short period of time and do not capture the full characteristics of the noise which may vary over time. Therefore, they perform best when the background noise is stationary in nature. Also the audio quality of the signal produced by noise reduction methods is not very high due to the inaccurate estimation of the noise. Even though these techniques do not always improve speech intelligibility they have been shown to reduce discomfort [5].

Since these methods are not always desirable, i.e. when the noise is non-stationary such as traffic, it would be beneficial for the user to be able to switch it on and off as preferred. Alternatively, the frequency response of the gain could be adjusted automatically by the device depending on the environment. For example, the full frequency spectrum can be amplified in quiet but if a noisy environment is detected, the hearing aid reduces the gains at low frequencies [23].

2.4.4 Environment Classification

It has been established that different environments (e.g. restaurants, concerts, lecture halls, quiet/nature) need different hearing aid parameters [4]. To address this, hearing aid manufacturers create different listening programs with parameters specific to a given environment.

The listening programs contain different signal processing strategies to optimize hearing in specific environments. For example, the noise reduction algorithms and directional microphones maybe be switched on and off accordingly. The programs are then downloaded on to the device. The user has the choice to manually switch to the program that they think best suites the environment they are in. Having to identify the environment and then switch to the appropriate program can be considered annoying and cumbersome for them [7]. There are also some users that might not be able to perform this task e.g. older people and children. If this switching could be done automatically, the hearing experience of individuals would be improved greatly.

The hearing aid has to be able to extract certain features from the environment in order to identify and load the correct listening program. The main challenge in environment classification is identifying a robust set of features that will be able to identify the acoustic characteristics of several environments. In [4] the author used simple features such as fluctuations in the signal envelop, slopes of the high and low frequency components and finally the average frequency [4]. These features were used to classify everyday background noises like traffic, typing and washing dishes. Other approaches have used neural networks and Hidden Markov Models as part of their classification strategy [6].

Modern hearing aids are now equipped with classification systems that are able to detect which acoustic environment the user is in and automatically load appropriate hearing aid parameters or programs that will give the best listening experience for that environment. For example, current hearing devices from Siemens are able to successfully classify four acoustic environments speech in quiet, speech in noise, noise and music [22], while Phonak hearing aids can classify speech in broadband noise from other signals [24].

Another challenge is to decide which processing parameters are relevant in each environment. Also, after classification has been completed successfully, what should be the initial values of these parameters? Whether these parameters are to remain fixed or continuously adjustable by the user is another issue. These problems will be analyzed in a later section when learning the user's preferences within an environment is discussed.

2.4.5 Hearing Aid Fitting

The amplification prescribed by clinicians is a good starting point for the gain required to compensate for an individual's hearing loss. The prescribed gains however still need to be adjusted after the patient has worn the hearing aid in real life situations. This fine-tuning is done on subsequent visits to the clinic where the patient gives feedback to the audiologist on the quality of sound produced by their device.

There are a few strategies audiologists use to aid the fine-tuning process. One is called Paired Comparisons where two sounds from different gain-frequency responses are presented to the user and they are asked to identify which one sounds better to them. Usually the user is asked to pick their preference based on criteria like intelligibility, loudness and comfort. If there is more than one frequency response to be investigated, they can be presented to the users in pairs with each other or they may be paired with a base response obtained from the prescribed procedure [5]. The audiologist keeps a tally of how many times each response is identified as producing the preferred sound. The one picked the maximum number of times is taken as the optimum for that individual and programmed into the device [5]. The issue with this method is that it does not tell the clinician how good or bad the quality of the sound is, just whether the user liked it better [5]. Also if this method is to be implemented properly it could take a long like time to present all combinations of comparisons to the patient. Another fine-tuning strategy is Absolute Ratings of Sound Quality. Here the user is given a chart with a rating scale from 0 (being the worst imaginable) to 100 (being the best imaginable). In between values could be 10, 20, 30 etc. or "bad", "average", "good" etc. When the sound stimulus for a given response is presented to the user through the hearing aid, they are asked to rate the sound quality on the scale. Different sound quality criteria such as loudness, tone quality and comfort are ranked using this scale. Each rating by the user corresponds to a specific increase or decrease in gain by the audiologist. These methods can be effective but would require a lot of time to be invested by the patient and clinician. Also it is sometimes frustrating for the patient to describe what they are hearing and difficult for the audiologist to interpret what the problem is. Therefore there is still room to improve this part of hearing aid fitting.

The possibility of using neural networks in the fitting process has been investigated and seen to be feasible [14, 26-27]. Audiologists usually keep a record of gains that have worked well for their patients; these records can be used to develop a databank of audiometric data and corresponding prescribed gains. This information can be used to train a neural network since input (audiometric data) and output (gains that were used to compensate the hearing loss) are available. This will provide a neural network which when given a patients pure tone test results will output the gains that the patients needs. Therefore, for future patients their test results could be fed to the network and the corresponding gains can be obtained [14]. This method is advantageous because the audiologist can quickly select a hearing device for their patient. However, it is not uncommon that two people with the same type and degree of hearing loss end up ultimately having very different preferred gains. It is therefore still necessary to customize the gains to each user.

Unfortunately, fitting is usually done in a clinical environment which is not realistic of any of the noise and reverberation characteristics that are present in real-world settings [7, 14]. Also, in trying to simulate real environments it is impossible to simulate all combinations of environments in life and narrow it down to those that the hearing aid user is exposed to. While optimizing the gain needed in one environment, the clinician could undo adjustments made for previous environments which would produce an overall poor fitting [7].

Going forward, there is a need to improve the fitting process and reduce the amount of time required for tuning the hearing aid and getting to its optimum parameters. It has been proposed that the user be directly involved in the process [28]. They are to be presented with a set of controls with which they can adjust the gain parameters of their hearing aid. Since they are already familiar with adjusting the controls for other common audio equipment, this should not be a difficult task for the patient. While wearing the device in their day-to-day activities, they are to adjust the response to optimize the quality of sound they hear. To support this shift from prescribed fitting to a perceptual one [28], the authors of this proposal carried out a study which confirmed that hearing aid users do want to be involved in the fitting process and find the idea of having direct control appealing. This study also found that the users can be trusted to eventually settle on the same optimum value after adjusting

the controls for a while. Since the hearing aid will be updated instantly with the adjustments, the time frame to optimal fitting will be greatly reduced. This is a more natural and practical method to fine-tuning the prescribed gain than forcing the user to pick between stimulus presented during the Paired Comparisons and Absolute Ratings of Sound Quality methods. It is also deemed to have more benefit since real-life acoustic sounds are being used to tune the device.

This paradigm can be extended further by having the user also decide the values of other hearing aid parameters such as noise reduction, spectral balance or directional microphones in each environment.

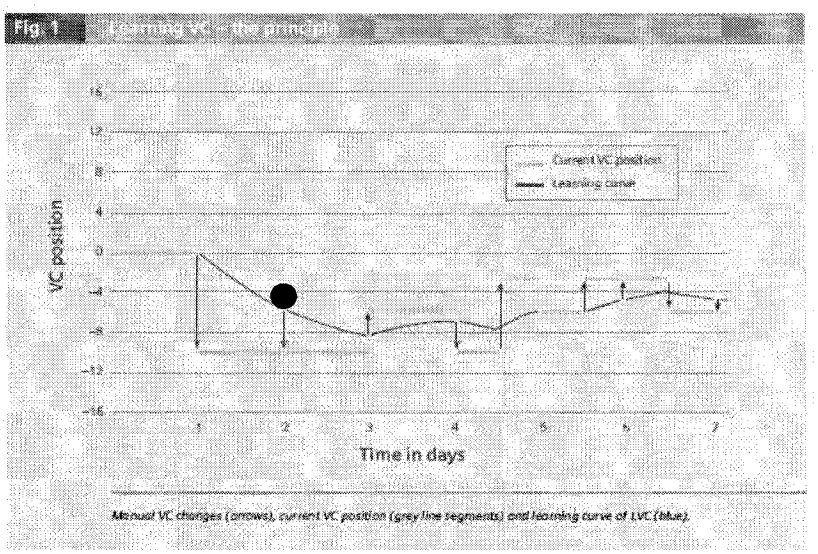
2.5 Self-Learning Hearing Aids

Personalization and recommendation of products and services is a growing trend to meet consumer's needs and demands [29]. This is of particular interest when making purchases through the internet, based on what products the user selects to buy, the system can recommend other items of interest and also predict what kind of products they would like in the future [30]. For example in audio, text and image retrieval from shopping databases such as Amazon, searches are made for movies, music, clothing e.t.c, only items relevant to the user will be returned as the users preferences are learned [29].

The concept of self-learning hearing aids addresses the issues of fine-tuning and optimal adjustment by learning what settings the user prefers in environments that are presented on a daily basis. The device has memory and remembers previous settings. At specific points in time, it adjusts the present settings to some combination of the previous ones. Some research has already gone into this and a few devices with self-learning features are currently on the market. The SAVIA© hearing aid from Phonak has a datalogging component that analyzes and stores the volume changes made by the user as they go about their day-to-day activities in each environment. Based on the analysis a volume setting is suggested [31]. The clinician makes a decision on whether or not to include this suggestion at next follow up visits as the device does not automatically update the volume. Another

hearing aid manufacturer, GN Resound, approaches this issue in a slightly different way. The gain function of the device is adjusted after each volume control change. Features are extracted from the input signal and used in the Automatic Volume Control (AVC) module to calculate a gain. The gain is applied to the input to produce the output that will be heard by the user. The system absorbs changes in the volume control register into the AVC module [25] every time. The parameters of the AVC are adjusted so that the gains applied will put the input at a level preferred by the user.

The CENTRA© hearing aid by Siemens is able to learn user preferences for volume in different listening environments. Each time the hearing aid is turned on it calculates the weighted average of past volume preferences and sets the volume accordingly. The device is typically able to learn the volume preferences for a specific environment on average by the end of the first week using a fixed learning constant of 17 hours [9,46].

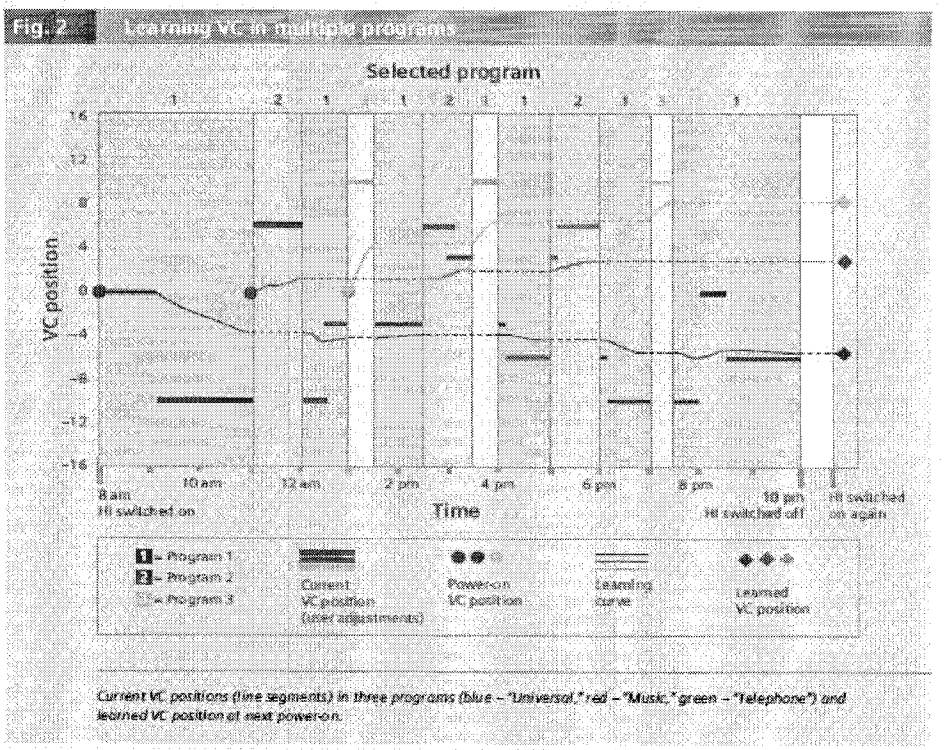


From Siemens, "DataLearning & Learning VC : Learning Preferred Gain Settings in Everyday Life"[9]
Figure 2.7: Learning Volume Control Concept

Figure 2.7 shows an example of the learning process in the CENTRA© over a week for a given user and listening program. When the user gets the hearing aid, it is set to the gains from the prescriptive procedures and this corresponds to a volume control (VC) position of 0. From then on, the user may decide to override this volume setting and a learning process is established. The horizontal axis is the time in days that the device has been worn. The

numbers corresponds to the end of a day and the commencement of learning for the next day (e.g. 1 is the end of learning for day 1 and beginning of learning for day 2). The thick curve is the output of the volume control learning algorithm while the actual volume control values set by the user are the light horizontal lines. The user reducing the volume is denoted by a downward arrow while if the user increased the volume an upward facing arrow is used.

In figure 2.7, the user does not change the volume during Day 1. On the second day, the user reduces the volume by 10dB. At the start of the third day, when the device is turned on, the hearing device computes a VC position of -6dB (see dot in figure 2.7) which is a combination of the previous VC positions. On the third day however the user prefers to leave the VC at -10dB instead of the position suggested by the learning algorithm so the learned volume reduces slightly. On the fourth day, the learning algorithm suggests -8dB and the user increased the position a little to -6dB. Over time the amount of change between the learned volume control setting and the user manual setting decreases (smaller arrows). In the CENTRA©, learning takes place independently in each listening program. The learned value is an exponentially-weighted average of past volume settings in each listening program and it only takes effect the first time each listening program is selected after the device is turned on. Figure 2.8 shows an illustration of the learning process in three listening programs of the Siemens's CENTRA ©.



From Siemens, "DataLearning & Learning VC : Learning Preferred Gain Settings in Everyday Life"[9]

Figure 2.8: Learning VC in multiple programs

Figure 2.8 shows the user's choice for the volume control (thick lines) and the learning algorithm behavior (thin lines) for one day independently in three listening programs. Different listening programs are selected within a day as the users' environment changes. When a new listening program is selected, the volume is set to the last volume control value set by the user in the previous occurrence of the listening program during that day. At the end of the day the device calculates a learned VC position. The hearing aid will be initialized to this VC setting the next time it is turned on. This is done independently in each listening program. The learning is implemented with a non-adaptive learning algorithm using an exponential averaging equation with a fixed time constant.

2.6 Summary

In this chapter, some of the problems experienced by hearing aid users have been highlighted as well as some of the advances that have been made by the hearing aid community. Hearing aids are still not widely accepted and there is a need for further improvements, particularly in the area of self-learning or trainable hearing aids to customize the devices to the particular needs and preferences of the users [7]. This customization is also needed because as discussed previously there are situations where certain features of the hearing aid should be turned off. More specifically, research is needed on the best strategies to learn what the user prefers in each environment to anticipate hearing aid control adjustments when required.

This thesis focuses on advancements in the learning of user control preferences in order to minimize the amount of adjustments made to the hearing aids over time. Reducing the need for frequent adjustments would create a better listening experience and draw less attention to the fact that an assistive hearing device is being worn.

Chapter 3 Learning Algorithms

During hearing aid fitting, the prescriptive procedures used by audiologists are very general and still need to be fine-tuned after the patient has worn the device for a while. The user gives feedback to the clinician on the quality of the sounds they can hear and then the gains are adjusted to optimize hearing. It has been established that the optimum hearing aid parameters vary in different environments [28, 32]. Even within a specific environment, the range of acoustic and listening conditions is such that the hearing aid user may require on-going adjustments of the control settings. Continuously adjusting the volume settings of the hearing aid has been identified as one of the reasons why individuals are not satisfied with their hearing aids, and as a result, may not wear them [3]. As described in Section 2.5, there is a trend among manufacturers to design trainable hearing aids that learn the preferred volume control settings in each environment to minimize user interventions and improve customer satisfaction. There are many different methods in existence that are able to learn or track a time series like the use of neural networks, linear predictive coding and kalman filtering. However because of the real-time processing constraint in hearing aids, simple algorithms are usually targeted.

In self-learning hearing aids, the user preferences need to be learned as efficiently as possible and should require minimum intervention from the user. Tracking user behavior by taking an exponentially-weighted average of past control settings in each environment, as is done for the CENTRA© (figure 2.8), is a simple way that has been used to achieve this goal. Exponential smoothing techniques are used in forecasting in many fields to make predictions of the next value in a time series [33, 34]. Of the wide range of exponential smoothing algorithms, the basic ones use fixed time constants. The issue however is in deciding what values for these constants to use in the algorithm since they directly affect how well it performs. Therefore, many schemes consider using adaptive parameters to improve performance. To use exponential smoothing algorithms in hearing aids, there is a need to explore whether using strategies with fixed or adaptive constants would be more beneficial for this application of interest. The above methodology will be the focus of this

research as it is a natural extension of the strategy employed in the CENTRA© which uses fixed exponential smoothing for learning. The basic averaging method using a fixed time constant is introduced and then methods in which the time constant is made adaptive are discussed.

3.1 Fixed Exponential Smoothing

Exponential smoothing is a common technique used in making forecast or future predictions based on historical data. Businesses use it to make strategic decisions such as operations planning and inventory control [33, 34], it is used for predicting job run times on shared processors [35] and channel behaviors on Wireless LANs [36]. The smoothing process tries to forecast what will happen in the future i.e. at $t+1$ based on what has occurred up to this point in time t. For example, the revenue from sales for an upcoming month can be predicted based on the trend of revenues seen in previous months. In exponential smoothing, the observed values of the time series are weighted unequally with more recent observations having greater weight than previous values.

The basic equation used is defined as [33 - 35]:

$$\hat{y}_{t+1} = \alpha y_t + (1-\alpha)\hat{y}_t \qquad (3.1a)$$

α is the smoothing constant $(0 \leq \alpha \leq 1)$.

$\hat{y}_{t+1}$ is the forecast for the next time period $t+1$

$\hat{y}_t$ is forecasted/prediction for the current period t

y_t is what was actually observed at time t

The relationship between α and the time constant τ is shown below in equation (3.1b), details of its derivation are shown in Appendix I.

$$\alpha = 1 - e^{-\frac{T}{\tau}} \qquad (3.1b)$$

where T is the time interval between samples and τ is the time it takes for the weighted average to reach 63% of its final value, i.e. at time t = τ = T, the right hand side of equation (3.1b) becomes $1-e^{-1}$ which is 0.63. Equation (3.1a) is typically initialized by setting

$\hat{y}_t(0)$ to the average of the first couple of values in the time series if they are known. The forecasted estimate $\hat{y}_{t+1}$, depends on a fraction of the newly observed value y_t and a fraction of the previous estimate $\hat{y}_t$. The amount of dependence is controlled by the smoothing constant. Ideally the difference between the forecast $\hat{y}_t$ and the actual observation y_t should be as close to zero as possible. The value of this difference is called the forecast error *e(t)* and is defined in equation (3.2).

$$e(t) = \hat{y}_t - y_t \tag{3.2}$$

Figure 3.1 shows how the algorithm works for a sample time series (plotted in a solid line) using a smoothing constant α of 0.2 and $\hat{y}_t$ is initialized to zero. The exponentially weighted average calculated at each time period t is plotted in dash lines.

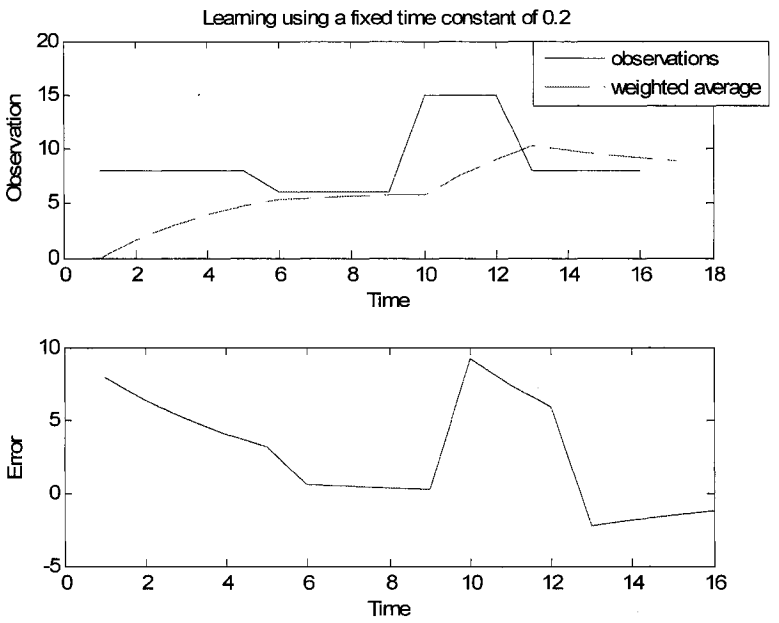


Figure 3.1: Simple exponential smoothing with a constant α of 0.2

The average over all the absolute forecast errors obtained from using an α of 0.2 was 3.6 for the above plotted time series.

The choice for α should depend on the nature of fluctuations expected in the time series. A lower value of α (or longer time constant τ) is better at smoothing out randomness because only a small fraction of observed values is used in the forecast. Therefore, if the fluctuations are merely random then using a lower α will help ignore these types of changes. However, if the fluctuations are valid and are due to a level change or change in trend of the time series, using a low α will be very slow in responding to this change. A higher α (or short time constant τ) value will respond to changes faster. We do not always want the algorithm to respond to noise in the data because it might not be able capture the underlying trend of the time series. In practice, the degree to which the data will fluctuate is typically unknown and knowing the optimum value of α to use in the calculation becomes quite a challenge. Next, some methods in which the value of the smoothing constant is updated during the analysis are introduced.

3.2 Adaptive Smoothing Constants

In the previous section, an overview of Fixed Exponential Smoothing was given. This method assumes that a smoothing constant α is sufficiently accurate for long time forecasting of the data. A challenge however is to find the best smoothing constant for the data. Moreover, a fixed smoothing constant that is optimal in cases of a dynamically changing time series such as predicting user behavior in setting hearing aid controls poses a greater challenge. It would be more beneficial if α could be updated with changes in the data observed. Different values of the smoothing constant would be used at different times in the analysis to produce the forecast. Typically, one could use a large α in the beginning (small τ) when there are few observations and a smaller α value later on to give more weight to the past observations (large τ). Also, a large α can be used when the data is varying and a small α when its values are fairly stable [37]. Adaptive-control procedures can be used to update the smoothing constant α depending on the changes in the values observed. Various adaptive exponential smoothing methods have been researched and evaluated and a few are discussed next.

3.2.1 Adaptive Control of the Exponential Smoothing Constant

In this approach, based on the work of Chow [38], a primary time constant is used to make the next forecast $\hat{y}_{t+1}$, along with two secondary smoothing constants that are used to detect if changes in the primary constant are warranted. Based on previous observations, the primary smoothing constant is continuously updated if one of the two secondary constants is found to give a better forecast. The two extra constants used are α_U, an upper bound, and α_L, a lower bound on the primary α. They are used to detect fluctuations in the data and are defined in equations (3.3) and (3.4) as,

$$\alpha_U = \alpha + d \qquad (3.3)$$

$$\alpha_L = \alpha - d \qquad (3.4)$$

where d is a small deviation in the smoothing constant which sets the speed of adaptation. Its value is specified by the designer and it stays the same throughout the processing. Three forecasts are made using each of the constants as shown in equations (3.5) to (3.7) but only the one for α is actually used in the analysis.

$$\hat{y}_{t+1} = \alpha y_t + (1-\alpha)\hat{y}_t \qquad (3.5)$$

$$\hat{y}_{t+1,U} = \alpha_U y_t + (1-\alpha_U)\hat{y}_{t,U} \qquad (3.6)$$

$$\hat{y}_{t+1,L} = \alpha_L y_t + (1-\alpha_L)\hat{y}_{t,L} \qquad (3.7)$$

However, if α_U or α_L is found to give a better forecast than the primary constant α a posteriori, then in the next time period α is changed to this "best" value. The best constant is defined as the one that gives the minimum forecast error *e(t)* and is defined for all three constants in equations (3.8) to (3.10).

$$e(t+1) = \hat{y}_{t+1} - y_{t+1} \qquad (3.8)$$

$$e(t+1)_U = \hat{y}_{t+1,U} - y_{t+1} \qquad (3.9)$$

$$e(t+1)_L = \hat{y}_{t+1,L} - y_{t+1} \qquad (3.10)$$

where $\tilde{y}_t$, $\tilde{y}_{t,U}$, $\tilde{y}_{t,L}$ are the forecasts made by α, α_u and α_L respectively for the observation y_t. Ideally, the difference between these two values, i.e. *e(t),* is close to 0. Appendix I provides the details of this algorithm.

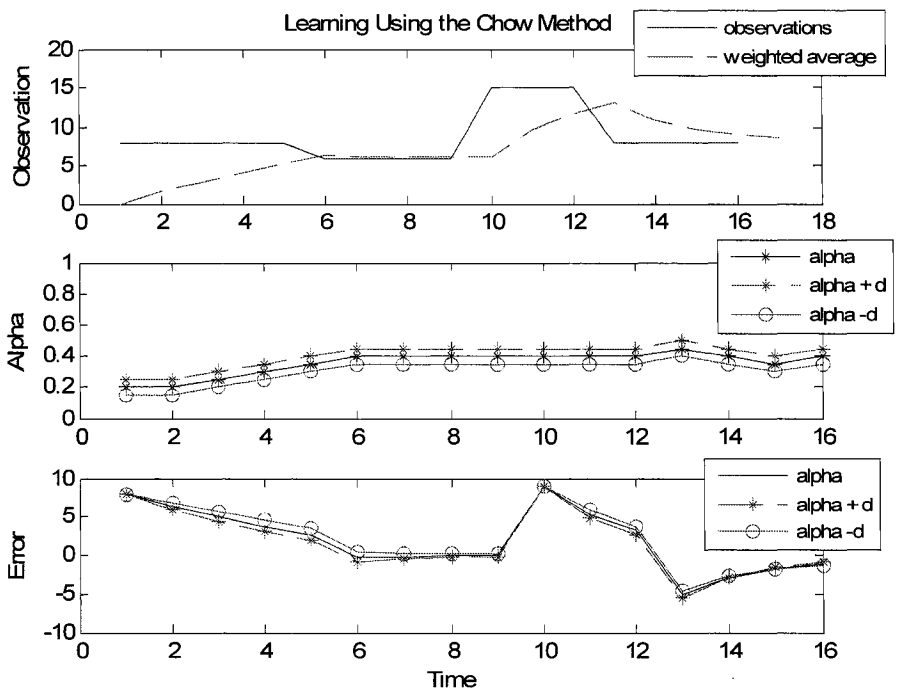


Figure 3.2: Adaptive Exponential Smoothing using Chow

The top panel in figure 3.2 above shows how this method works for a sample time series (same as the time series in figure 3.1) which is plotted in a solid line. The output of the algorithm is plotted in dash lines. The initial estimate $\tilde{y}_{t(0)}$ is set to zero, α is initialized to 0.2 and the deviation $d = 0.05$. The middle panel shows the changes in α during the process. For example, at t = 2 we have $\alpha = 0.2$; but it is observed that at t = 3, α increases to 0.25. This is because that at t = 2 $\alpha_u = 0.25$ gave a smaller error than both α and α_L and therefore constant α is updated to this value and the secondary constants are adjusted accordingly using equations (3.3) and (3.4). Similarly, at time t = 14, α decreases from 0.45 to 0.4 because α_L gave a smaller error at time t =13. From t = 6 to 12, α is not changed because the secondary constants did not perform any better. The average absolute forecast error obtained for the given series using this method was 3.4, this is slightly lower than that obtained using

a fixed time constant of 0.2 (as discussed in Section 3.1). The error value depends on the choice of d.

The mean absolute deviation is another criterion that can be used for deciding which of the three constants to use in the next period [38]. The equation is described in (3.11) below:

$$D(\alpha, N) = \frac{\sum_{t=1}^{N} |e_t(\alpha)|}{N} \qquad (3.11)$$

where N is the number of observed values and e(t) is the single-period-ahead forecast error. $D(\alpha_L, N)$, $D(\alpha_U, N)$ and $D(\alpha\ , N)$ are calculated for the corresponding smoothing constants according to this equation. When $D(\alpha\ , N)$ is lowest, then smaller errors are generated using this constant, therefore α is left unchanged. If $D(\alpha_L, N)$ is lowest, α is reduced to α_L. New values are then calculated for α_L and α_u using equation (3.3) and (3.4). If $D(\alpha_U, N)$ is the lowest of the three, α is increased to α_u. The values for α_L and α_u are recalculated using equations (3.3) and (3.4). After each update of α, $D(\alpha_L, N)$, $D(\alpha_U, N)$ and $D(\alpha\ , N)$ are initialized to zero.

The disadvantage of the method by Chow is that three forecasts and errors have to be calculated at every point in time [38] using all three constants thus increasing the number of computations to be carried out. There is also the issue of the appropriate selection of d in the algorithm. Some added benefit of this method is that the rate of change of α can be controlled by the designer and even though there is more computation, it is not very complex.

3.2.2 An adaptive extrapolative forecasting algorithm

This approach, based on the work of Pantazopoulos and Pappis [39], aims to find the optimum smoothing constant for past observations and applying this knowledge to forecast future values in the time series. The optimum value for α that would have given a forecast error of zero as defined by equation (3.8) is calculated based on the previous observations [39].

If one assumes that y_{t+1} is known at time *t*, it can be substituted for $\hat{y}_{t+1}$ in the basic exponential smoothing equation (3.1a) to give:

$$\hat{y}_{t+1} = y_{t+1} = \alpha_t y_t + (1-\alpha_t)\hat{y}_t \qquad (3.12).$$

Using equation (3.12), the value of α that would produce a forecast error of zero at time t+1 can be determined a posteriori as follows:

$$\alpha_t = \frac{y_{t+1} - \hat{y}_t}{y_t - \hat{y}_t} \qquad (3.13)$$

In reality, this information is not available at time t and changes from one instant to the other. However, equation (3.13) can be used to find the past optimum smoothing constant at times *t-1*, *t-2*, *t-3* etc. [39]. For example at t-1, we have:

$$\alpha_{t-1} = \frac{y_t - \hat{y}_{t-1}}{y_{t-1} - \hat{y}_{t-1}} \qquad (3.14)$$

The authors propose to use α_{t-1} determined as in equation (3.14) for the estimation at time t. This is based on the assumption that the time series does not vary too fast and is fairly stationary.

It is proposed that the absolute value is taken to prevent negative solutions for α_t [39]. The proposed form is therefore defined as follows,

$$\alpha_t' = \left| \frac{(y_t - \hat{y}_{t-1})}{(y_{t-1} - \hat{y}_{t-1})} \right| \qquad (3.15)$$

To accommodate cases where the value for α_t' is unstable (i.e. $y_{t-1} - \hat{y}_{t-1}$ is very close to zero) and to keep it within the bounds of [0, 1] it is set to 1 if the value calculated in equation (3.15) is greater than 1. The smoothing constant is not updated if it performed well and gave a forecast error of zero. The final form of the method as proposed by the authors is summarized in equation (3.16):

$$\alpha_t = \begin{cases} \alpha_{t-1}', & if \ e_{t-1} = 0 \\ 1, & if \ \alpha_t' > 1 \\ \alpha_t', & otherwise \end{cases} \qquad (3.16)$$

Once the smoothing constant calculated according to equation (3.16) is found it is then substituted into equation (3.1a) to find the next forecast $\hat{y}_{t+1}$. One disadvantage with this method is that the best α value tends to remain at the ceiling value of 1 and the forecast ends up being the last observed value. This is because equation (3.13) is really the two-step ahead forecast error divided by the one-step ahead forecast error (with the latter usually larger) and the equation is often greater than one. Therefore, according to the rules of the method stated in equation (3.16), α would be set to 1. The requirement that the time series has to be fairly stationary is also another limitation of this method. Nonetheless, the method is attractive because unlike other adaptive methods there are not any additional parameters that need to be considered, in contrast to the previous method (Section 3.2.1) which requires specifying the size of adaptation step d. The algorithm for this method is summarized in Appendix I.

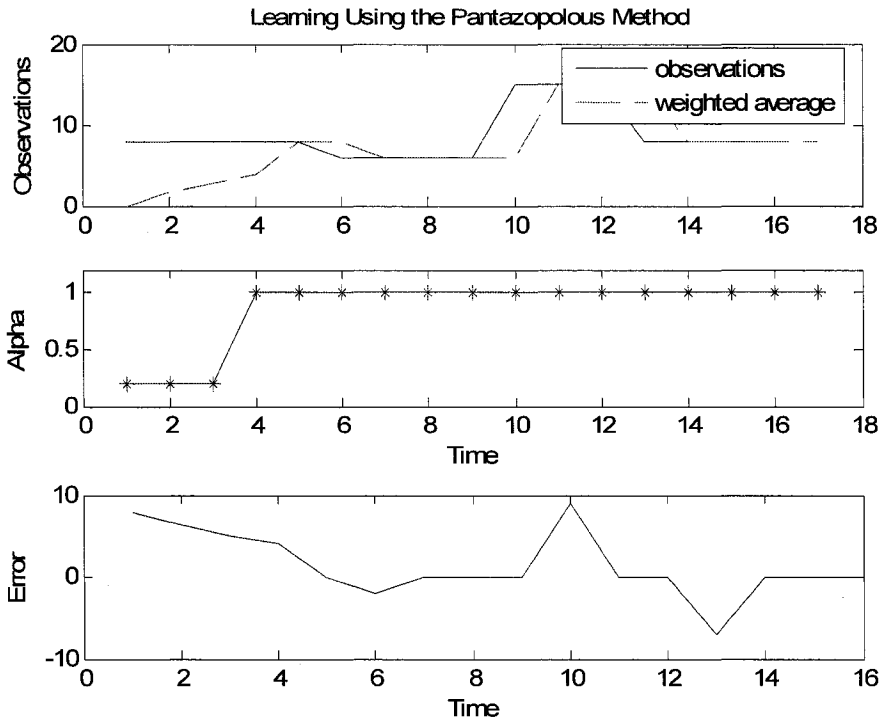


Figure 3.3: Adaptive Exponential Smoothing using the Method by Pantazopolous

The time series used in the learning example in figure 3.3 is the same as that used in illustrating the Chow method in Section 3.2.1 with α initialized to 0.2. Of particular interest is the plot of α versus time in the middle panel in figure 3.3. After time t = 3 α has a value of 1 for the rest of the analysis. The average absolute forecast error was 2.6 using this method which is also lower than that obtained using a fixed time constant of 0.2 in the exponential smoothing equation.

3.2.3 Exponential Smoothing with an Adaptive Response Rate

In this approach, based on Trigg and Leach [40], the authors view the problem at hand as one where noise has been added to the time series and the aim of the forecasting system is to filter out this noise. During forecasting the system might go out of control if there is suddenly a large value for the noise. This spike could be random or might be a real change in the level of the time series. If the former is the case then α can be reduced to limit the effect of the noise. However, if the series is indeed changing then a lower α will result in a biased and one-sided estimate of the series and the forecast will take a long time to reach the appropriate level. For level shifts, we would want α to increase to give more weight to current values so the forecast can reach the new level quickly. To help in differentiating between these two cases and prevent biased forecasts, the authors propose the use of a tracking signal to monitor the noise.

The tracking signal is calculated in equation (3.19) as a ratio of the smoothed forecast error A_t to the smoothed absolute forecast error M_t defined in equation (3.17) and (3.18) [40]:

$$A_t = \phi e_t + (1-\phi)A_{t-1} \qquad (3.17)$$

$$M_t = \phi|e_t| + (1-\phi)M_{t-1} \qquad (3.18)$$

$$\alpha_t = tracking\_signal = \left|\frac{A_t}{M_t}\right| \qquad (3.19)$$

where e_t is the forecast error between the previous estimate and the current observation, and ϕ is the smoothing constant for the error. The authors recommend that ϕ be 0.1 or 0.2. The purpose of the tracking signal is to detect any bias in the estimates, i.e. if the series is being consistently under or over estimated. The tracking signal is always in the range [0 1]. It goes to the higher limit for larger negative or positive bias errors and fluctuates around zero for small errors. The algorithm reacts to this error signal by setting the smoothing constant ϕ to the absolute value of the tracking signal as shown in equation (3.19).

When e_t is very small or A_t and M_t are close in value, α will be close to 1. That way, for a higher bias, α is increased to capture the current behavior faster and once this happens it is reduced as the bias decreases. The algorithm is summarized in Appendix I.

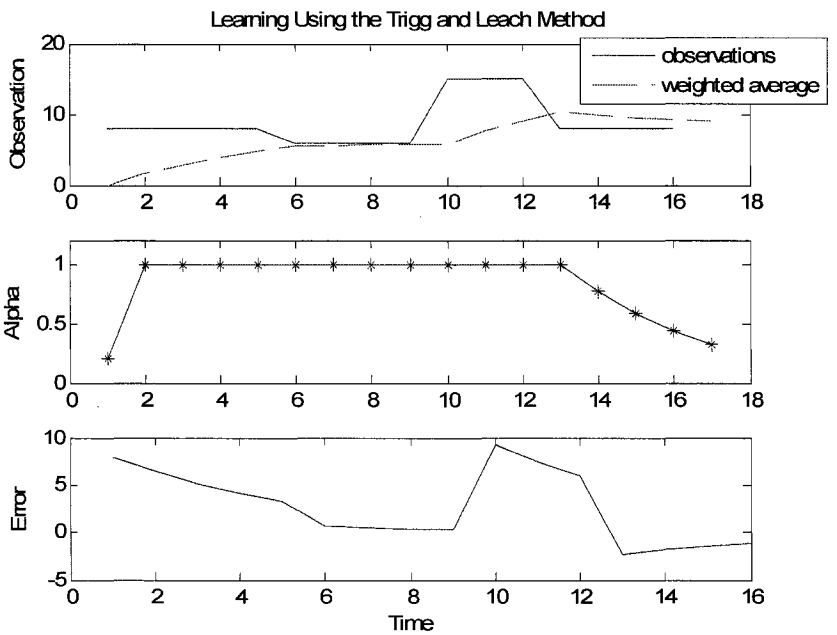


Figure 3.4: Adaptive Exponential Smoothing using the Method by Trigg and Leach

Figure 3.4 illustrates the Trigg and Leach method using the same time series as before. The time series is plotted in solid lines while the forecast is plotted in dashed line. The bottom panel shows the changes in α with time. In this example, α is set to 1 most of the time. This is because of the level shifts; the algorithm was constantly underestimating the series, therefore α was increased to try and capture the current nature of the data. The average of the absolute forecast error obtained from using this method was 2.8, also lower than the total error obtained using a fixed time constant of 0.2.

This method is very popular because of its ability to react rapidly to sudden changes and various algorithms are based on it [36]. However, there is the issue of what the best value ϕ should be in equations (3.17) and (3.18). It also sometimes results in an unstable forecast because of the possible division by zero.

3.2.4 Smooth Transition Exponential Smoothing

In economic theory, it is hypothesized that the economy behaves differently depending on the region that certain variables lie, e.g. high versus low interest rates or high vs low employment rates [41]. The switching between regions or regimes is in reality gradual and the economy reacts slowly to these changes [42]. This gradual switching is modeled as a smooth transition in a class called Smooth Transition Regression models. In the approach proposed by Taylor [43], one of the parameters is modeled as a function of a transition variable [43]. The basic equation of one of these models is defined as follows in equation (3.20):

$$y_t = a + b_t x_t + e_t \qquad (3.20)$$

where

$$b_t = \frac{\omega}{1+\exp(\beta+\gamma V_t)} \qquad (3.21)$$

and a, ω, β and γ are constant parameters and V_t is a transition variable [43]. If $\gamma < 0$ and $\omega > 0$ then b_t becomes a monotonically increasing function of V_t bounded between 0 and ω. Taylor applies this method to adaptive exponential smoothing by adjusting the smoothing constant based on a transition variable that is specified by the user. The model by Taylor used is shown in equation (3.22) below:

$$\alpha_t = \frac{1}{1+\exp(\beta+\gamma V_t)} \qquad (3.22)$$

where β and γ are constants and V_t is a transition parameter that influences the value of α_t. If γ is negative then α_t increases with an increase in V_t and the opposite is the case with a positive γ. Figure 3.5 below shows how α will vary with positive and negative values of γ.

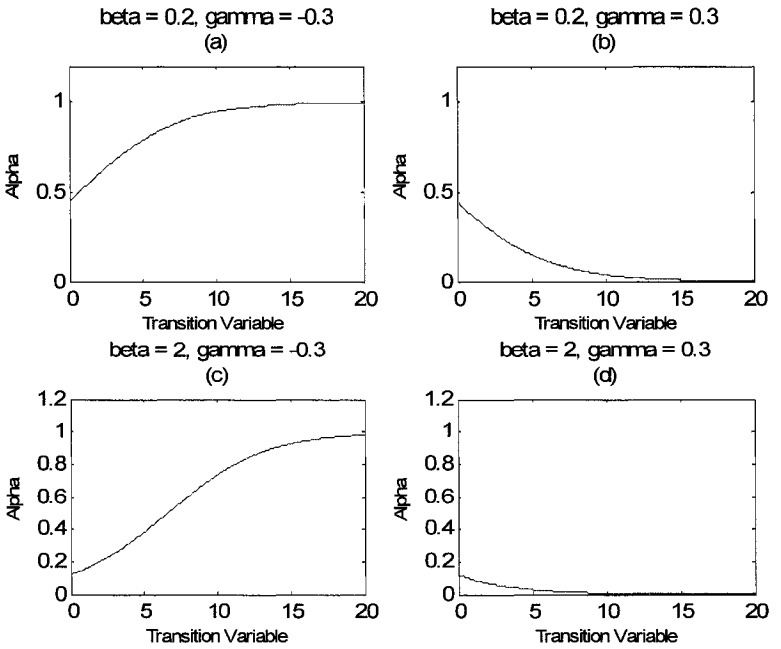


Figure 3.5: Variation of α for positive and negative values of γ

The value of V_t is calculated as observations come in and α_t is updated accordingly. Suggestions for the value of the transition variable V_t are the square (e_t^2) or absolute value $|e_t|$ of the forecast error from the previous period e_t as well as the mean or squared errors over time. The value obtained from equation (3.22) is used in equation (3.1a) to compute the weighted average. The algorithm is summarized in Appendix I

Taylor [43] carried out a large scale simulation study using time series generated from three processes. One series contained only level shifts, another contained only outliers and the third contained both level shifts and outliers. For the time series with only level changes, it was found that the sign of γ was irrelevant and the method performed well for both positive and negative values of γ. The proposed method was found to react quickly to level shifts and converged to the series very quickly. Also the adaptation of α was found to be very stable only adapting when an overall level change occurred. For the same data, the Trigg and Leach behaved just as well in converging to the series quickly but the adaptation of α was more erratic and seemed to adjust for all changes in the data.

For a series with outliers, the method performed better when a positive constraint for γ was used [40]. The method did not react too much, instead the value of α was decreased to give less weight to the outlier. This is desirable because we do not want the forecasts to be affected greatly by random fluctuations in the time series. This method performs poorly when the data is made up of both level shifts and outliers. Also there is the issue of choosing the best values for β and γ. The changes in α produced during the adaptation are however very stable and the method performs well for time series with solely level shifts or outliers. The authors suggest using the absolute value or square of the forecast error *e(t)* as the transition variable.

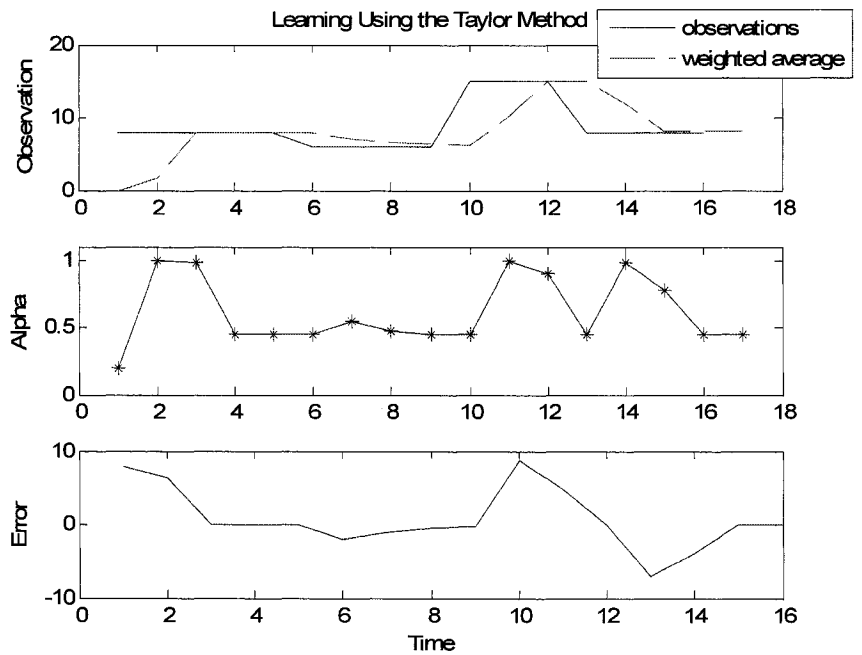


Figure 3.6: Adaptive Exponential Smoothing using the Method by Taylor

Figure 3.6 shows the output for a sample time series that was processed using the proposed method with $\alpha = 0.2$, $\beta = 0.2$ and $\gamma = -0.1$. The behavior of the adaptation of α will be similar to that illustrated in the top left plot in figure 3.5(a), the constant will be 0.5 for low errors increasing gradually to 1 as the error value increases. In the bottom plot in figure 3.6, the error value is close to zero at time periods 3 to 9, causing α to stay close to 0.5 (see the middle plot). At $t = 1$, the error is fairly large (10) and so at time $t = 2$, α increases to 1 to try and capture the time series characteristics faster. Similar behavior is observed at times $t = 10$

and 13, α is increased at t = 11 and 14 respectively for the same reason. The average absolute forecast error obtained from using this method was 2.7, which is lower than that obtained using a fixed time constant of 0.2.

3.3 Summary

In this chapter we discussed four adaptive methods for learning user profiles. All the methods performed better than the simple fixed exponential smoothing method used in the examples presented. They were also found to give better estimates and fewer errors in the simulations carried out by the respective method authors. Next, they will have to be modified to suit the purpose of this work which is to learn hearing aid volume control settings. The chapters that follow discuss the simulations and tests that were carried out to compare the different methods and the results that were obtained.

Chapter 4 Simulations and Performance Measures

To test the different exponential smoothing algorithms discussed so far, user behavioral data on volume control preferences is required. Unfortunately, hearing aids that have the data logging feature do not necessarily store every single setting due to limited memory. Data could be collected through the use of a laptop or PDA where the control settings are transmitted wirelessly from the hearing aid. However, getting recordings for a wide range of user behaviors would take years making it very difficult to get access to real life data. In this work, simulated/modeled user behavior was used in testing. To evaluate the adaptive methods, performance measures are needed for comparisons. This chapter discusses the bank of data that was generated and the error measures identified for evaluating the performance of the various algorithms. Next, the need for exploring the use of an adaptive time constant instead of a fixed one in the learning algorithms is determined based on analysis of the fixed exponential smoothing method.

4.1 User Behavioral Data

Since very little real life data is available for testing, test data had to be generated. Profiles of some expected user behavior was created to simulate what volume control setting they would prefer at a given point in time and the duration of time that the volume control is maintained at that value. The user preferred settings are assumed to be Gaussian distributed with a mean μ which represents the average desired gain specified for the user and a standard deviation σ which represents how much the user will vary in the desired gain. A high σ means the settings specified by the user fluctuate over a large range, and the converse occurs for a small σ. Plot (a) in figure 4.1 shows an example of a user with an average preferred volume setting of 5dB and standard deviation of 2. The range of possible preferred average gain settings is from -10 to 10. The duration that the setting is maintained varies and was modeled as having a uniform distribution. An added factor of the amount of time the user spends in an environment is also considered and is assumed to also be uniformly distributed. An example is illustrated in the plot (b) of figure 4.1, where the user is assumed to be in a specific environment for a minimum of 3 hours and a maximum of 8 hours. For

each setting generated from plot (a), its corresponding duration is obtained from the distribution in plot (c) of figure 4.1. The minimum duration which the user can apply a setting in this example is 1 hour and the maximum duration is 5 hours.

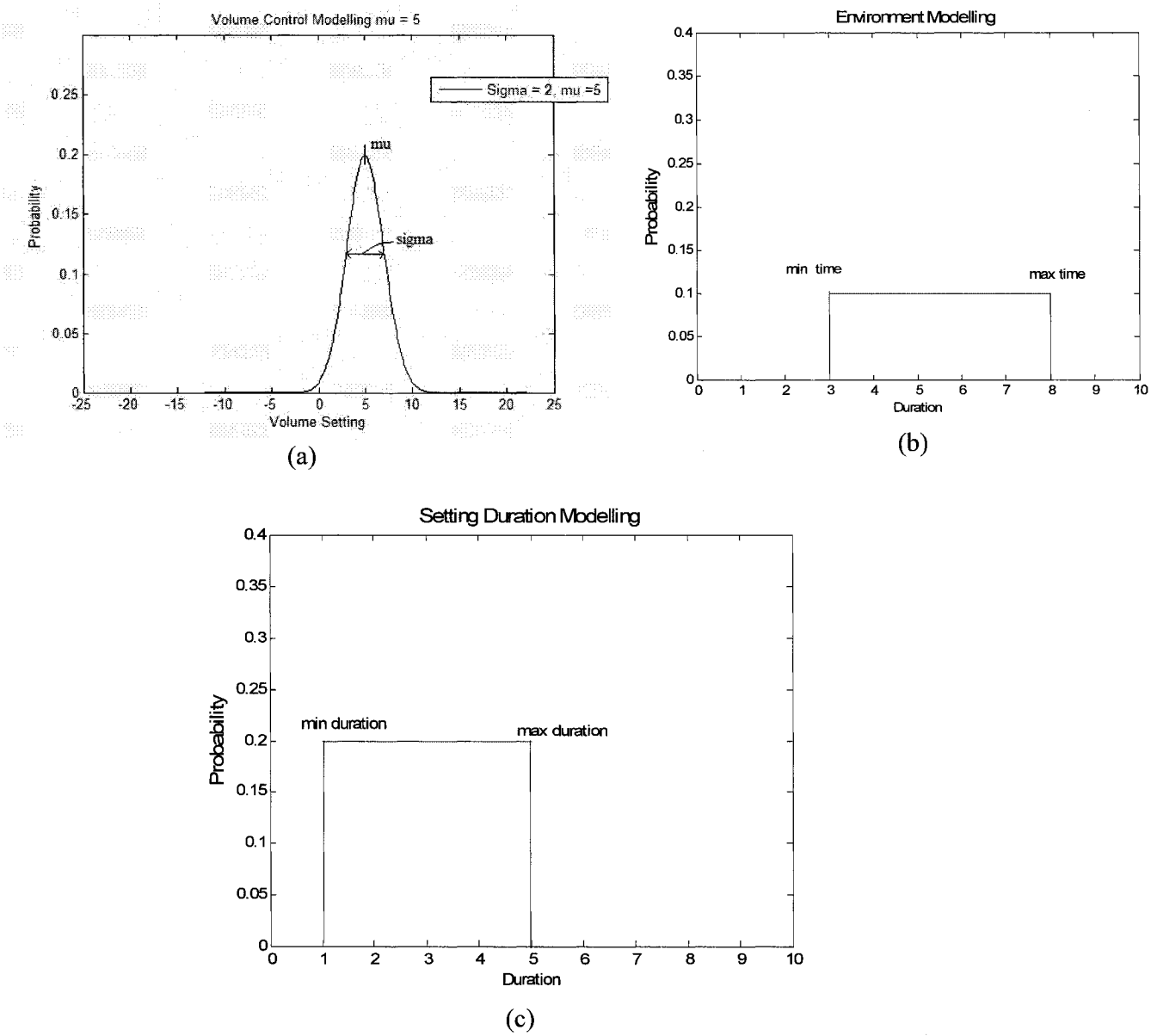


Figure 4.1: Modeling volume control (dB), environment length (hr) and setting duration (hr) for user behavior

The function normrnd(μ,σ,m,n) in MATLAB generates random numbers from the Gaussian distribution specified by mean μ and standard deviation σ, m and n are the dimensions of the

m-by-n output matrix. This function was used to generate the actual volume control settings of a user. By specifying the average setting μ and amount of variance from this mean σ a resultant *m-by-n* matrix with values reflecting this model is generated with this function. The fuction unidrnd(N) returns an array of random numbers chosen uniformly from the set {1,2,3,...N}. This function was used to determine the duration that a setting is maintained and the amount of time a user spends in an environment. For each value generated by the normrnd function a corresponding duration value is generated using the unidrnd function. The choice of N will be discussed later on in this chapter.

The block diagram in figure 4.2 shows an overview of how the data was simulated. First, the number of phases Q in a profile is generated randomly where a phase is defined as the period of time a specific environment is presented to the hearing aid until the user exits and moves onto another environment. Next the length of time N a person would spend in that occurrence of the environment is generated from a uniform distribution. The length N is then divided into S_a smaller divisions where $1 \le a \le n$. The next step was to generate the volume level for the n divisions created previously.

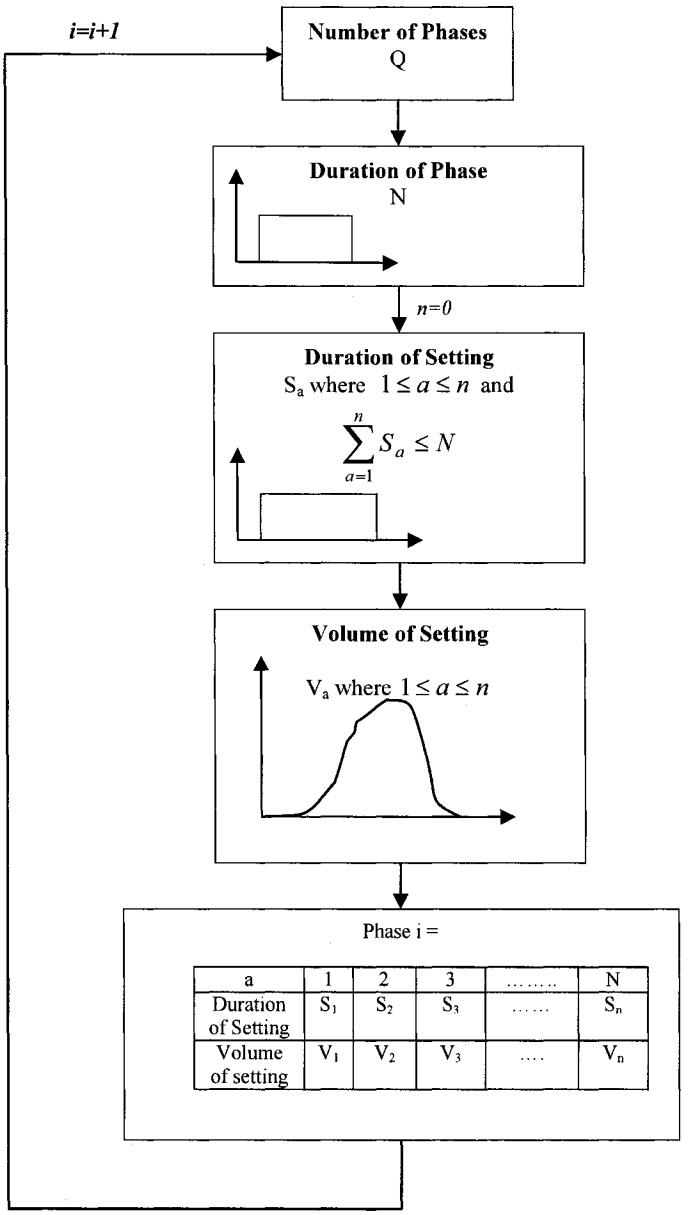


Figure 4. 2: Block diagram of process to simulate user behavior

Three basic user profiles were generated using these functions and can be applied to any environment. Other secondary profiles were generated as a mixture of the basic ones. While the profiles studied could deviate from actual real profiles, the purpose here was to create a database of profiles encompassing a wide range of behaviors to test the performance of

different algorithms over widely varied conditions. These profiles have a limitation that the preferences are purely overall volume changes and do not include overshoots and sporadic changes however they are a good starting point for investigating learning hearing aid volume control preferences.

4.1.1 Profile Description

Profiles from three basic models in which the user has a homogenous behavior were generated. They are: fairly constant, fairly fluctuating and fluctuating behaviors. All other profiles are a combination of these three basic ones as described next. For all types of profiles (except short profiles) generated, the duration in an environment was chosen randomly using the unidrnd(N) discussed in Section 4.1 with N = 10 which is the maximum number of hours a user is assumed to spend in a specific environment in one day. For short profiles the user is assumed to be in a specific environment for less than 5 hours every day.

Fairly constant user (FC)

This type of user would be the ideal case, the person does not change the setting too often and is usually close to the desired mean by choosing a low σ . For any given μ, a standard deviation $\sigma = 2$ was used. The duration for each setting for this user is also coded in the simulations and is assumed to always be greater than or up to 1 hour

Fairly Fluctuating (FF)

For this user a standard deviation $\sigma = 4$ was used to model more fluctuation in the user interventions from the desired mean. Also the fairly fluctuating user changes the setting more often than the FC user therefore the duration of the setting was coded to be between thirty minutes and 2 hours using a uniform distribution between these limits.

Fluctuating User (Fluc)

The fluctuating user changes the settings more often and is less precise than the other two preceding behaviors. It is described by a $\sigma = 10$ for a given desired mean. The duration

modeling for this type of behavior was set to be uniformly distributed between fifteen minutes and 1 hour.

Mixture of Behaviors

The profiles for this type of user is a combination of the different types of behaviors (e.g. they could be of fairly constant behavior for a period of time and then fairly fluctuating behavior for another period). This is to portray how the user duration preferences change with time. When the hearing aid is used initially, the user's behavior could be fluctuating and would be changing the volume frequently since they are not quite used to the device. With time, the user should adjust the volume less and settle down to a fairly constant behavior.

Changing Preferences

These profiles are of a specific behavior but the desired mean changes two or three times during the profile. This could occur in cases where the environment is not homogenous. For an environment classified as speech in noise, the types of noise could vary and the class might need to be split to accommodate for this variation.

Short Profiles

The profiles generated for this scenario models the users' behavior in environments that are presented for only a short amount of time (usually < 5hrs) as there are some environments that are inherently short like being on the phone. Short profiles were generated for each of the three basic behaviors.

Varied Mean

In this model, the desired mean volume is constantly increased or decreased until it settles at a specific value. This can occur upon receiving a new hearing aid. Depending on how far off the prescribed gains are, the user might find it very unsatisfactory and consistently increase or decrease the volume as they get used to the device and settle at a satisfactory level.

Table 4.1 summarizes the behaviors modeled and the abbreviations that denote them in this thesis.

Behaviour	Notation	Description	Standard Deviation
Fairly Constant	**FC**	User makes only small changes to the settings and does not change settings very often.	2
Fairly Fluctuating	**FF**	User makes medium changes to the settings and changes settings more frequently than the above.	4
Fluctuating	**FL**	User makes large changes of the settings frequently	10
Mixture of Behaviors	**M**	User has some combination of the three basic behaviors above	
Changing Preferences	**CP**	User has one of the basic behaviors but changes their desired mean a few times in the profile.	
Short Profiles	Prefix - **SH**	User has one of the basic profiles, but the time spent in an environment is < 5hrs daily.	
Vary Mean	**VM**	User is consistently increasing/decreasing the volume until the desired one is reached.	

Table 4.1: User Behaviors Generated

User profiles were generated for the different behaviors above. Each sequence or phase corresponds to a particular user's behavior in an occurrence of an environment. Profiles made up of just one of the basic behaviors contained 60 phases where a phase is defined as the period of time a specific environment is presented to the hearing aid until the user exits and moves onto another environment. The start and end of each phase corresponds to the user entering and leaving the environment respectively (or switching on and off the device respectively). Sixty phases were found to be enough to give stable outputs for the error measures proposed to measure performance in this work. In figures 4.3 – 4.5, phases 1- 5 for samples of three profiles exhibiting each of the three basic behaviors are shown. Each panel

in the figures is a plot of the user's preferred volume control setting versus time for a defined period of an environment.

Figure 4.3 shows the profile for a fairly constant user. In this plot it can be observed that the volume control settings are quite constant and changes in the values are very small. For example in the top panel in that figure, the setting was increased slightly from -3 to -2 and this was the only change made for the six hours the environment was initially presented. The second panel shows the preferences when next that same environment is presented. The mean of this user behavior is a setting of -2 and it is seen that the volume control settings do not vary too much from this defined mean

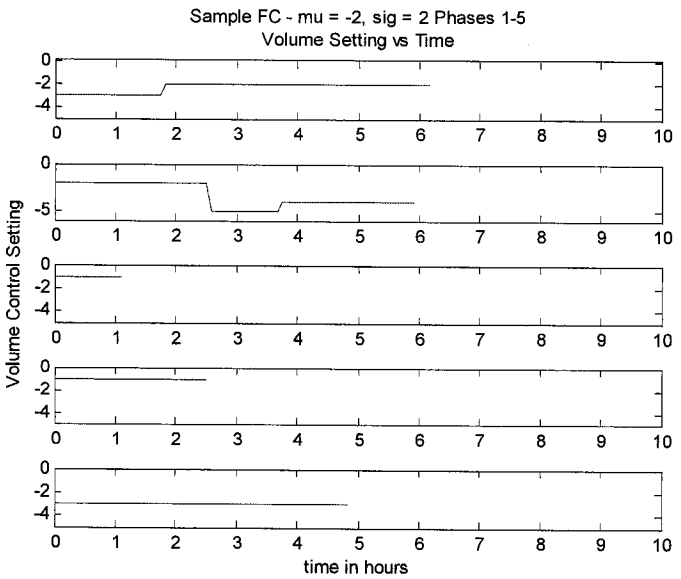


Figure 4.3: Phases 1-5 of a sample profile for a fairly constant user with $\mu = -2$ and $\sigma = 2$

In figure 4.4 the profile for a fairly fluctuating user is shown. In the top panel of the plot, the user changed the volume only once but they were only in the environment for 2 hours and the amount of change is more than that of the user in figure 4.3. The defined mean for this user is 6 but it can be observed that their preference varied from 10 to about -10 in the first panel.

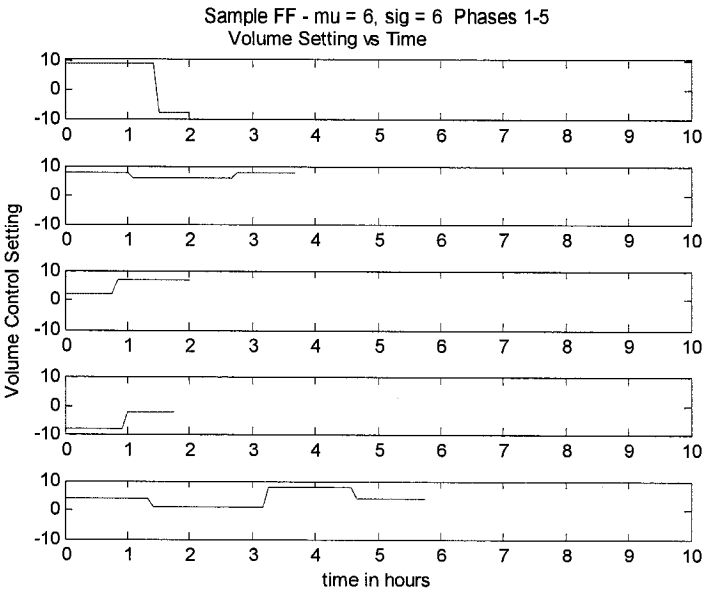


Figure 4.4: Phases 1- 5 of a sample profile for a fairly fluctuating user with a $\mu = 6$ and $\sigma = 6$

In figure 4.5, the user's behavior is more erratic and contains many changes to the settings both within a phase and over the whole profile. This is an example of a fluctuating user behavior. The desired mean is modeled as 4 but the user's preference varies as high as +16 (in the fourth panel) and as low as -12.

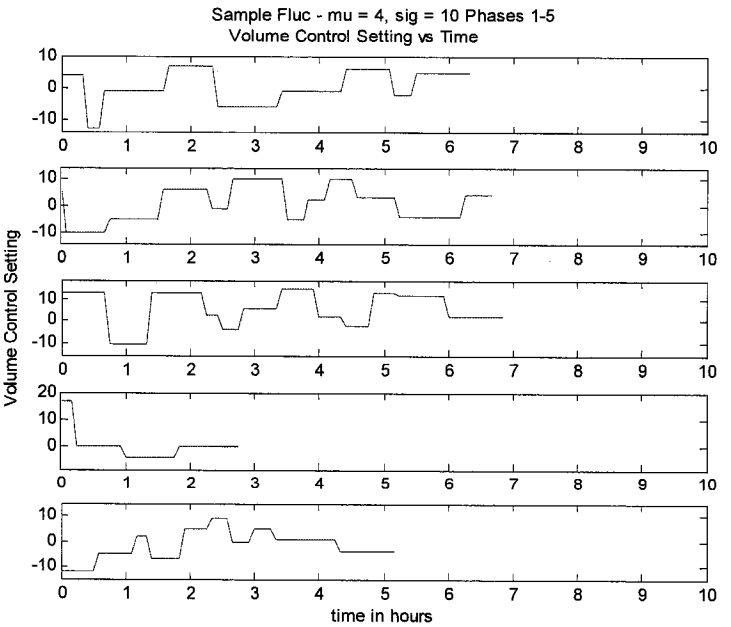


Figure 4.5: Phases 1-5 of a sample profile for a fluctuating user with a $\mu = 4$ and $\sigma = 10$

After each phase the algorithm calculates a learned value which would be the suggested volume control setting for the next time that environment is presented. The first phase is used solely for learning the initial conditions. An example of how the learning is performed over the different phases is shown below in figure 4.6. The circle is the final output from learning for that phase and it is the same as the suggested setting (*) in the next phase.

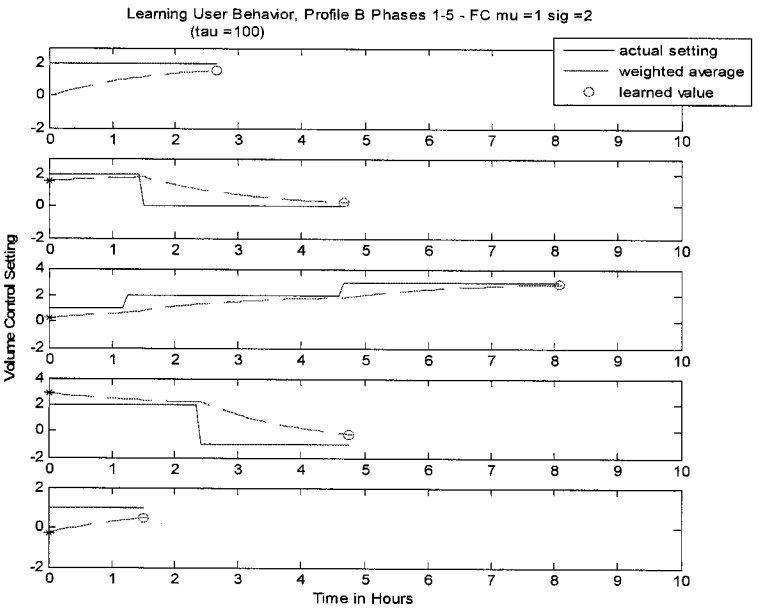


Figure 4.6: Learning of sample profile for a given simulated user ($\mu = 1$ $\sigma = 2$)

4.2 Measuring Performance

In this research, the goal of the learning algorithms is to minimize user intervention when entering a new phase in a given environment. New phases are continuous occurrences of an environment within a day. In this scenario, the initial recommended value of the volume control setting at the beginning of a phase (which is the final learned value in the new phase) is being compared to the actual user interventions in that phase. User interventions in this case are instances when the volume setting is unsatisfactory and causes the hearing aid wearer to adjust the volume controls.

Based on the learning, the hearing aid gives a suggested value when a new environment is presented. At this point, if the user is unhappy with this setting they will adjust the volume control right away. The algorithms will be evaluated relative to this first initial setting by the user. A good learning algorithm should reduce the likelihood that the user changes the settings and minimize that change where applicable. It is with these considerations in mind that the four performance measures discussed in this section were proposed.

4.2.1 Error Measure #1: Difference between the learned value and the first initial setting

A good algorithm should be able to successfully predict what the user prefers for the next phase, the absolute value of the difference between the learned and the first setting when the user returns to the environment is the first logical measure (shown in figure 4.7). For a set of observed control settings $\underline{x}_i = [x_1, x_2, x_3 \ldots x_N]$ in a phase i, $\boldsymbol{error1_i}$ is defined as follows:

$$error1_i = \left| T_{i-1} - \underline{x}_i\ [1] \right| \qquad (4.1)$$

where i = current phase ($i > 1$), N is the number of times the user set a value during phase i, $\underline{x}_i[1]$ is the first initial setting by the user in the current phase and T_{i-1} is the value learned at the end of the previous phase. The initial condition is $T_0 = 0$, which would correspond to the setting of the hearing aid upon delivery to the user from the manufacturer after the prescribed gains have been programmed into the device.

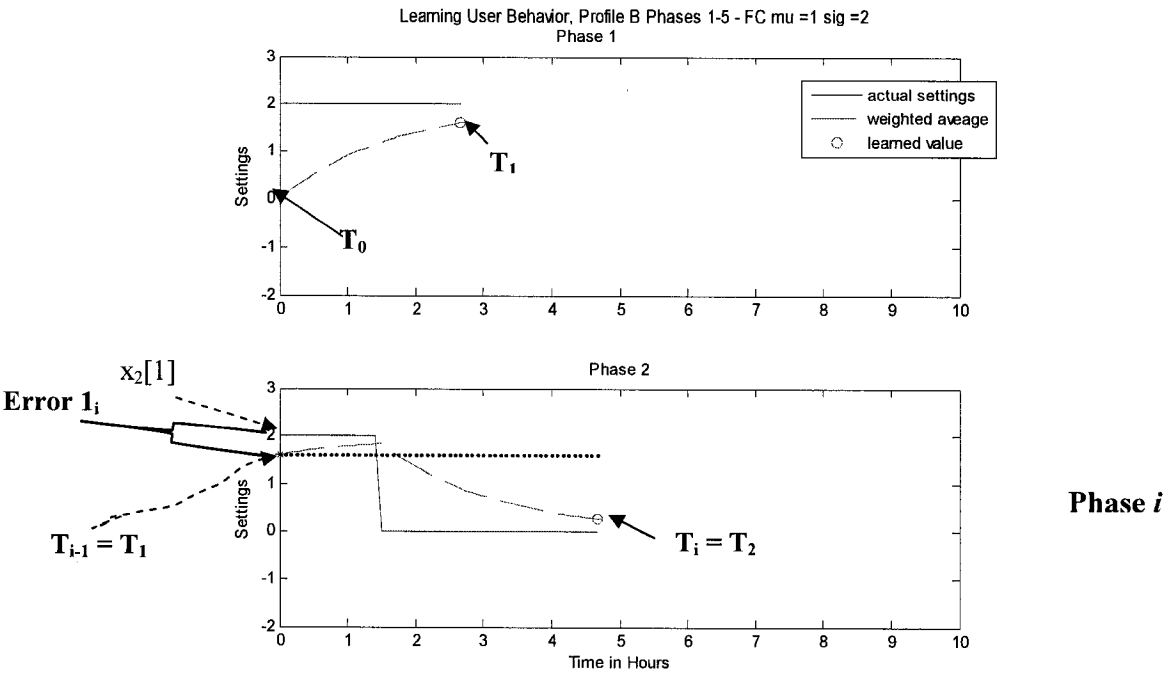


Figure 4.7: Error 1 measurement

4.2.2 Error Measure #2: Average Error

The second error is defined as the average error between the learned value and the actual settings during the entire phase as defined in equation (4.2) and shown in figure 4.8.

$$error2_i = \frac{1}{N}\sum_{j=1}^{N}(T_{i-1} - \underline{x}_i[j]) \qquad (4.2)$$

T_{i-1} is the learned value that the hearing aid is set to for phase i, where $i > 1$, $\underline{x}_i[j]$ is the j^{th} value set by the user (during the same phase i) and N is the number of times the user set a value during phase i. Error 2 reflects the bias in the estimate over one phase. We would like the error to be as close to zero as possible.

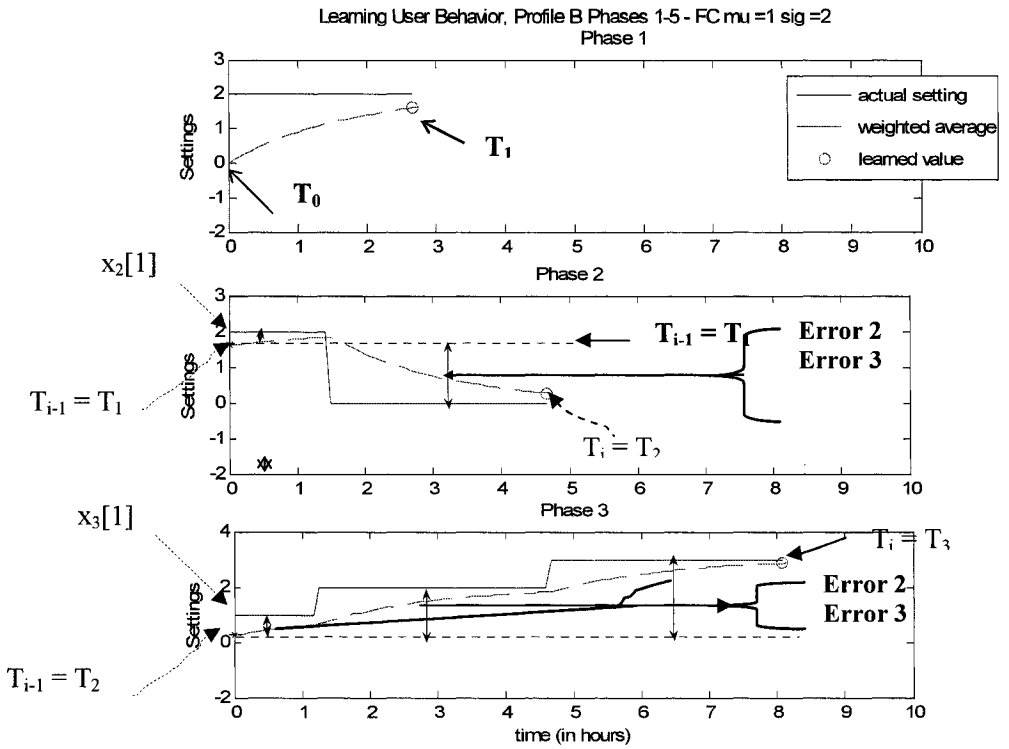


Figure 4.8: Error 2, average of the difference between learned (dashed line) and actual settings and Error 3, root mean square of the difference between learned (dashed line) and actual settings

While this error measures the bias in the errors, it does not however indicate the magnitude of the difference between the learned value and actual settings in a phase.

4.2.3 Error Measurement #3: RMS Error

The third error defined is the Root Mean Squared (RMS) error between the learned value and the actual settings during that phase. It is an extension of the error discussed above (see figure 4.8). It is defined in equation (4.3) below as,

$$error\,3_i = \sqrt{\frac{1}{N}\sum_{j=1}^{N}(T_{i-1} - \underline{x}_i[j])^2} \qquad (4.3)$$

where T_{i-1} is the learned value that the hearing aid is set to for phase i, where $i>1$, $\underline{x}_i[j]$ is the j^{th} value set by the user (during the same phase i) and N is the number of times the user set a value during phase i. This error measures how much the learned value deviates from the actual settings in the present phase. Unlike error 2 it gives an indication of the amount of the total (not just bias) error because the square of the difference is taken.

4.2.4 Error Measure #4: Percentage of time within error margin

In real conditions, the human ear is only able to detect an increase in sound level of typically 3dB or higher [44]. Therefore, if the algorithm is able to forecast estimates within this minimum threshold with respect to the user preference it could be considered as doing a good job. It is assumed that if the suggested estimate at the beginning of a phase is less than 3dB of what the user would have wanted, they are not likely to need to adjust the volume. Error measure #4, measures the percentage of phases for which the magnitude of the difference between the learned value T_{i-1} and the user's initial setting in phase i is greater than 3dB. In other words, it monitors how often the error1 measured value (as in figure 4.7) is greater than 3. Therefore, error 4 focuses on larger errors that may require more immediate user intervention, it is defined as follows:

$$error4 = \left(\frac{count(error1_i \geq 3)}{\#\ phases - 1}\right) * 100 \qquad (4.4)$$

where i is the current phase of the environment and $i > 1$.

4.3 Analysis of Fixed Exponential Smoothing

Preferences for volume settings will vary from user to user and the manner in which the adjustments will be made will also differ. Due to this variation it is difficult to accept that there exists a single time constant that will be able to adequately learn all these different behaviors. If it can shown that the optimum time constant varies from one user to another then that would justify investigating the use of an adaptive time constant in the learning algorithm. Therefore, to establish if using an adaptive learning constant would be more beneficial than using a fixed one, the τ value that would perform best in equation (4.1a) was found manually for different user profiles. Using the results obtained from the optimum time constant based on the performance measures outlined in Section 4.2, baseline results with which to compare the performance of the adaptive learning algorithm were established.

4.3.1 Simulation Assumptions

First, a database of different profiles was generated based on the behaviors discussed in table 4.1. Once again it is important to note that a wide range of behaviors was covered to include extreme cases. The summary of profiles generated can be seen in table 4.2 below.

Type of Profile	Number of profiles generated
Fairly Constant (FC)	20
Fairly Fluctuating (FF)	20
Fluctuating (FL)	20
Mixture (M)	50
Changing Preferences (CP)	20
Short (SH)	45
Varying Mean (VM)	13

Table 4.2: Summary of Profiles Generated

Every user profile is assigned an identifier composed of the type of behavior followed by a unique index for example, a profile labeled as FC1 is for user number 1 who has a fairly constant behavior, FL18 would be user number 18 who has a fluctuating behavior and a label that starts with the letter SH indicates that it is a short profile.

The volume control settings are assumed to be sampled at an interval of 5 minutes. For each profile, learning was performed using a fixed time constant in the basic exponential smoothing exponential (equation (4.1a)). Values of the time constant were varied to cover a wide range of learning rates from very fast to extremely slow. The time constant τ was changed in steps from $1 - 10^6$ minutes. For each value of τ, error 1, error 2 , error 3 and error 4 were calculated for each phase according to equations (4.1), (4.2), (4.3) and (4.4) over all phases of a user profile. To assess the quality of estimates provided for a user, the average of the calculated errors for a given profile is computed. For example for a profile with 60 phases, errors1- 4 is calculated for each phase then the average of each error over the 60

phases is taken. The equations used for a profile with Q phases are defined in equations (4.5), (4.6) ,(4.7) and (4.8).This was repeated for each τ for every profile in table 4.2.

$$error1 = \frac{\sum_{i=2}^{Q} error1_i}{Q-1} \qquad (4.5)$$

where $error1_i$ is given by equation (4.1) and Q is the number of phases in the profile,

$$error2 = \frac{abs(\sum_{=2i}^{Q} error2_i)}{Q-1} \qquad (4.6)$$

where $error2_i$ is given by equation (4.2) and

$$error3 = \frac{\sum_{i=2}^{Q} error3_i}{Q-1} \qquad (4.7)$$

where $error3_i$ is given by equation (4.3) and

$$error4 = (\frac{count(error1_i \geq 3)}{Q-1})*100 \qquad (4.8)$$

where $error1_i$ is given by equation (4.1).

The absolute value is used in equation (4.6) so that we get a sense of a magnitude of the bias error over the entire profile. These average errors were plotted against τ so that the optimum time constant value could be identified. The optimum time constant is defined as the one with the least amount of error. A logarithmic scale is used for the τ-axis. Some of the results obtained are discussed next.

4.3.2 Performance Assessment of Fixed Exponential Smoothing Algorithm

The underlying statistics of the profiles do not allow the Exponential Smoothing (ES) algorithm to reach a value of zero for error1 and error3 since the standard deviation is set in the behavioral profiles to a value greater than zero. Therefore it not expected that the minimum value for error1 and error3 to be much less than 2dB for fairly constant, or much less than 4dB for fairly fluctuating and much less than 10dB for fluctuating profiles. Error 4 is also expected to be higher for the fluctuating profiles since it is an extension of error 1 which is already expected to be greater than 3dB for these profiles. For the bias error error2, it is expected that short time constants will give smaller values for this error because they able to catch up to the time series faster due to the higher weight put on current observations. Also since the learned value T_o is initialized to zero, the exponential algorithm is expected to initially overestimate phases with a high negative mean or underestimate those with a high positive mean, especially for long time constants. From the analysis of the simulated profiles, it was found that the optimum time constant varies across profiles therefore, there is not an optimum range that would be best for learning different behaviors. Also the use of a fixed time constant of 17 hours is confirmed as it appears to be a middle point for learning fairly constant and the fluctuating profiles.

In the case of the fairly constant users (see figures 4.9 and 4.10), it was seen that the shorter time constants gave errors that were very close in value (i.e. approximately the same) then a gradual drop occurs as the optimum time constant is reached. At longer τ values a large error value is found. This is illustrated in the user with profile FC18 shown in figure 4.10. For this user FC18, τ from 1 – 100minutes gives an average error1 of 2.5 then drops to 2.1 at about 460 minutes and then finally increases gradually to 5 dB for much longer time constants. In some cases, a certain range of time constants gave similar error values then after a certain point the error started to increase as seen for the user with profile FC16 shown in figure 4.9. Here the minimum dip in the curve is very shallow. The maximum errors in the two plots are different in that FC18's is less since it has a smaller desired mean than FC16. The detailed result per user profile is provided below.

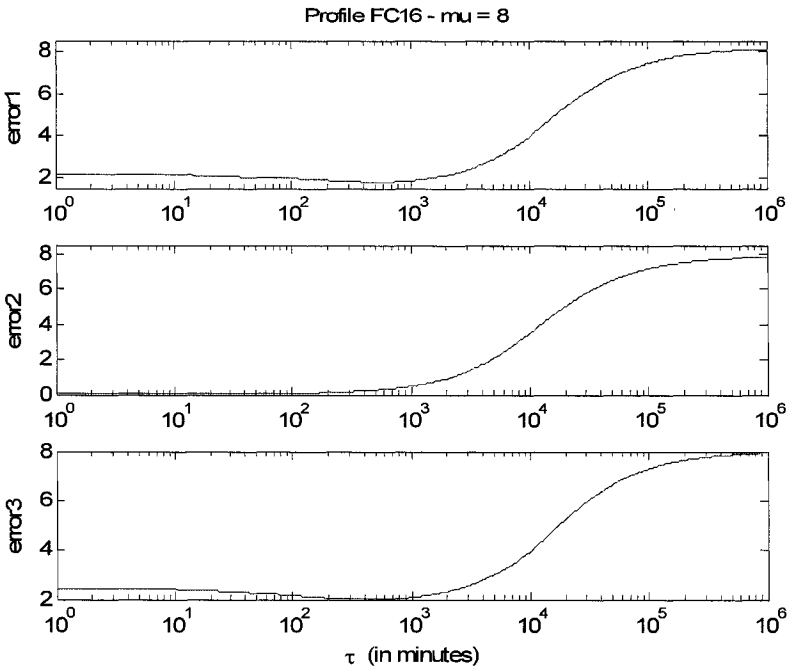


Figure 4.9: Error1, Error2, Error3 plots for profile FC16 μ = 8

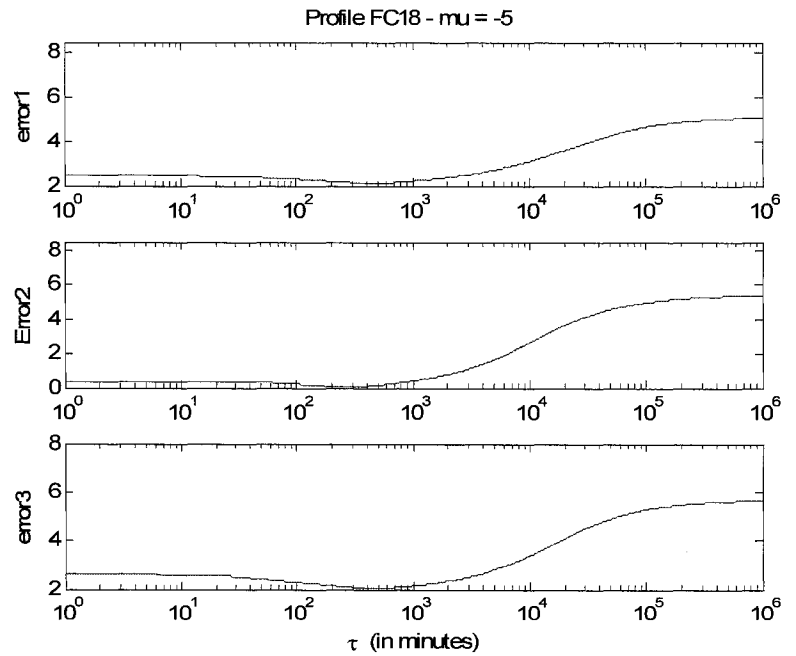


Figure 4.10: Error1, Error2, Error3 plots for profile FC18 μ=-5

The use of longer time constant values is detrimental for fairly constant behaviors because very little weight is being placed on the most recent user settings. Since the user's preference is not changing too quickly it is better to learn the control settings faster.

For FC profiles, there was usually a range of time constants that gave the minimum error4 measure. For example, the best performance for FC16 based on error4 was about 20% (see figure 4.11) i.e. the percentage of time the algorithm's learned value was greater than the initial setting by 3dB in a new phase was 20%. This minimum value occurred at time constant values in the range 200 – 1200 minutes. As observed in the other error measures for this type of behavior, longer time constants also led to poorer performance for error 4 (greater than 50% of the time).

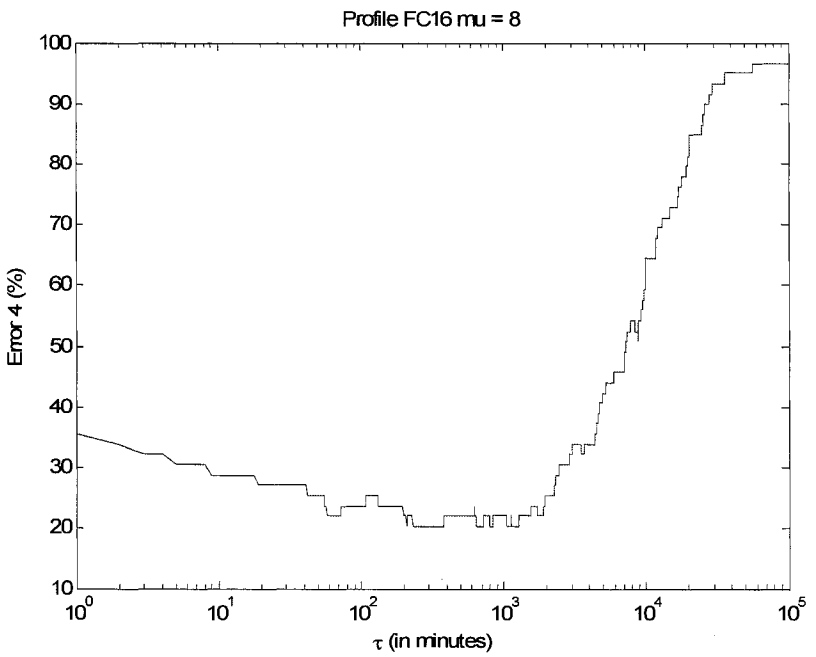


Figure 4.11: Error 4 measured for FC16 μ = 8

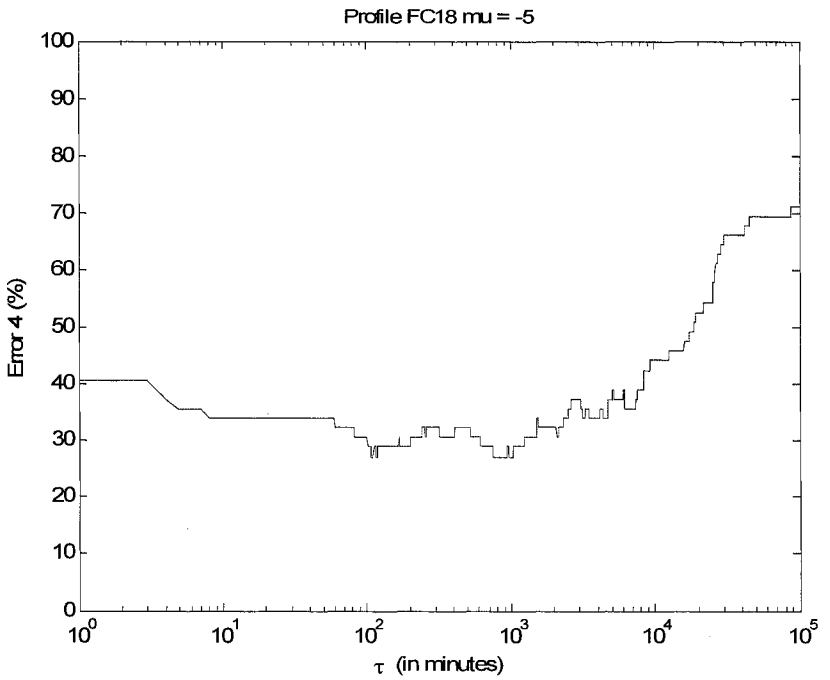


Figure 4.12: Error 4 measured for profile FC18 μ = -5

Similar conclusions can be drawn from the plot of the average error 4 versus τ shown in figure 4.12 for FC18. The optimum range is between 100 - 1000 minutes and the error value increased by about 30% above the optimum for much longer time constants.

In the case of the fairly fluctuating users, the error values for these behaviors are much larger due to the introduction of more fluctuations into the user settings (larger σ). This caused short time constants to be unfavorable in minimizing error 1 and error 3. The weight assigned to current user settings need to be reduced since the user's preferences are changing quicker and some drastic increases or decreases in the volume are noise and should be ignored since they are far from the desired mean. This is illustrated in figure 4.13 for user profile FF9, the desired mean assigned was -3. In this plot it is observed that quickly learning this user's behaviors gave much greater errors.

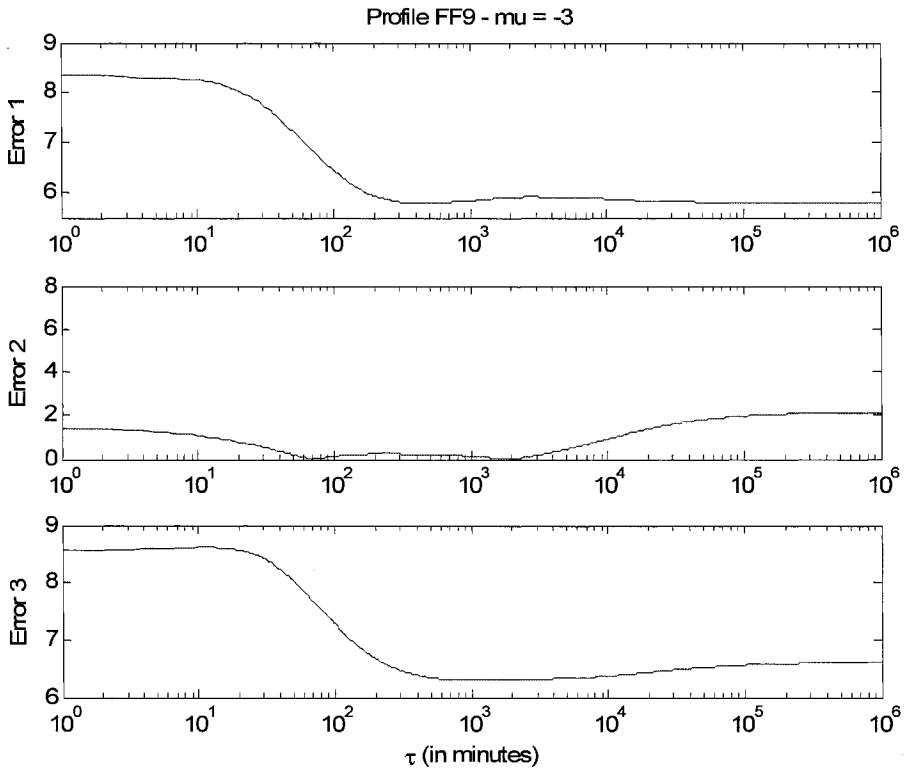


Figure 4.13: Error1, Error2, Error3 plots measured for user profile FF9 μ = -3

However, using a long time constant in this case did not always give the best results for error 1 for this type of behavior. The location of the range of optimum time constant depended greatly on the value of the desired mean volume assumed for the user when their profile was generated. If the mean is relative low say between -4 to 4 then the dip at the optimum τ was very small and time constants larger than this dip did not perform any better. An example is found with the user with profile FF9 μ = -3 shown in figure 4.13. For this user, the optimum τ range can be identified as being between 350 – 900 minutes giving an error 1 value of 5.8. It should be noted that higher time constants still performed approximately the same. This could be because the preferred mean is so low that even by giving little weight to current settings; the estimates provided by the algorithm at the beginning of a phase are not too different from what the user would have preferred.

For a higher desired mean, the drop in error 1 and the increase after the optimum time constant is very clear. This is because the over/under shoot for the estimate of the user's

initial setting in a phase is much larger than in the previous case with a low desired mean. This is illustrated in figure 4.14 where the performance for user profile FF8 is shown. The mean for this user is 7 and from the plot it can be seen that using a time constant greater than 10^4 minutes does not give good results for error 1.

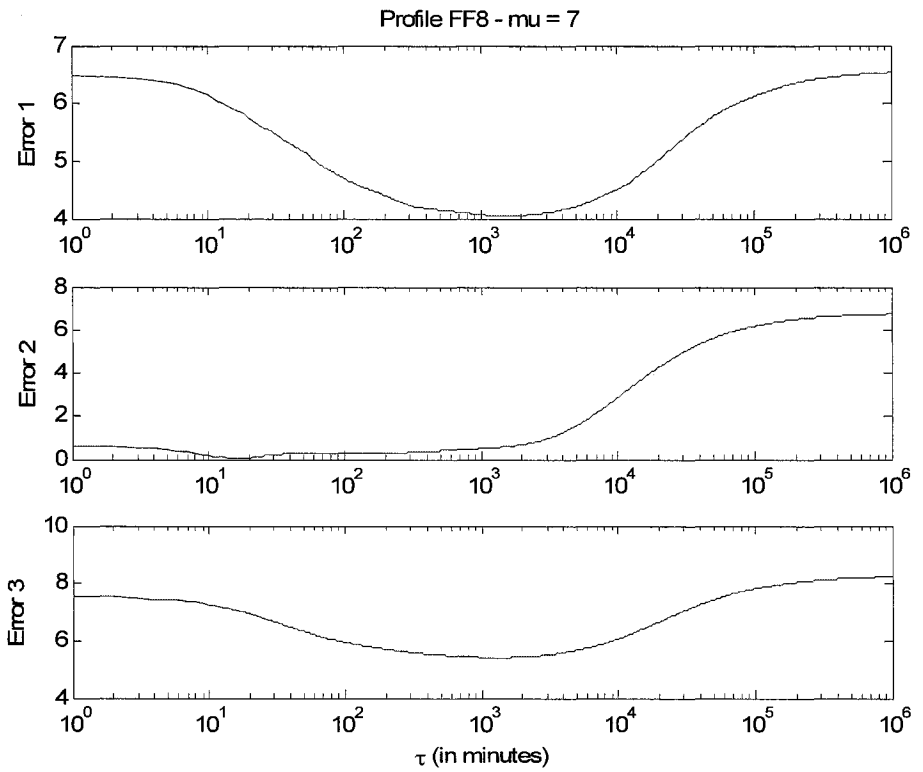


Figure 4.14: Error1, Error2, Error3 plots for user profile FF8, μ = 7

Error 2 for FF8 is zero for a time constant of 17 minutes and increases with the time constant. As expected, short time constants resulted in lower bias since they learn the time series faster by giving greater weight to current observations.

In general, error4 measured for the fluctuating profiles (FF and FL) is much greater compared to that obtained for the fairly constant (FC) behavior as shown in figures 4.15 and 4.16. This is in part due to the higher values of σ used in these profiles. It can be assumed that for these types of profiles any fixed time constant value would cause the user to frequently adjust the volume because the difference between the learned and actual setting is often greater than 3 dB, usually greater than 55% of the time.

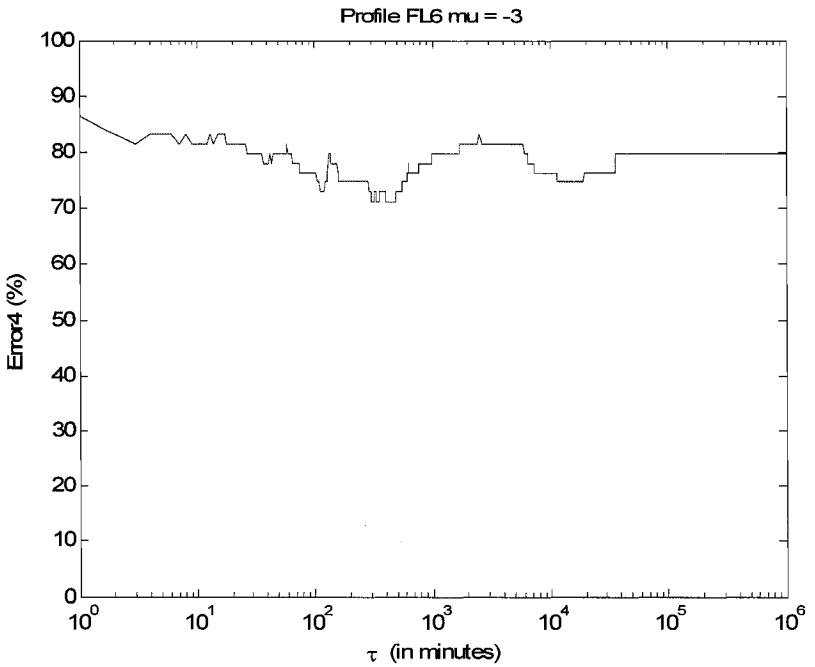


Figure 4.15: Error4 measured for user profile FL6

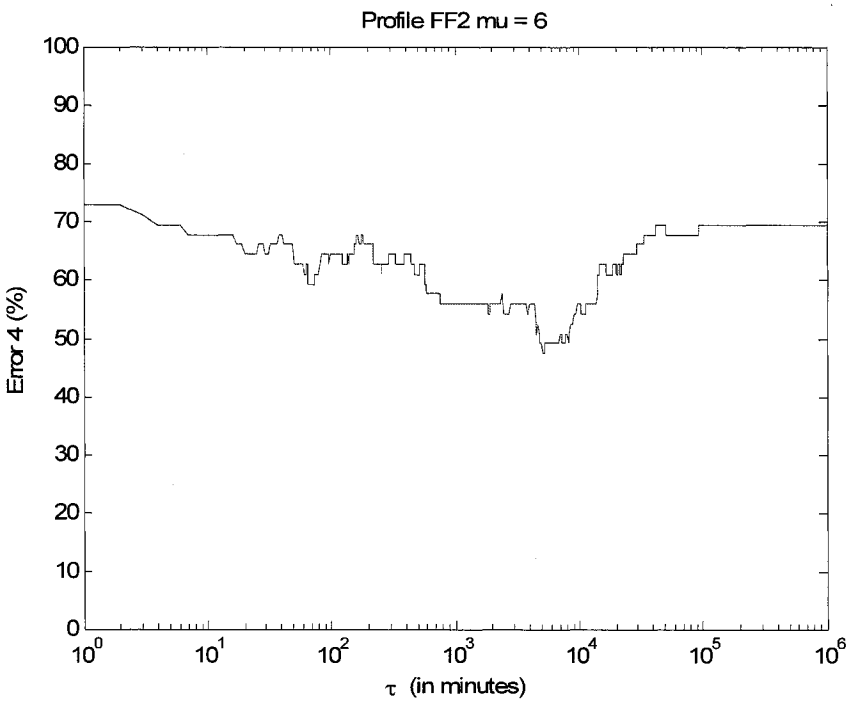


Figure 4.16: Error4 measured for user profile FF2

The observations for the fluctuating profiles (FL) were very similar to the fairly fluctuating (FF) cases. Error 1 and error 3 performances for longer time constants were also found to depend on the value of the assumed desired mean for a given profile. Figure 4.17 shows the results for the user with profile FL4. The mean for this user is 1. It can be seen that the longer τ ($> 10^4$) does not perform better or worse than the optimum time constant. For profiles similar to this, what matters is not to pick a time constant that is too short.

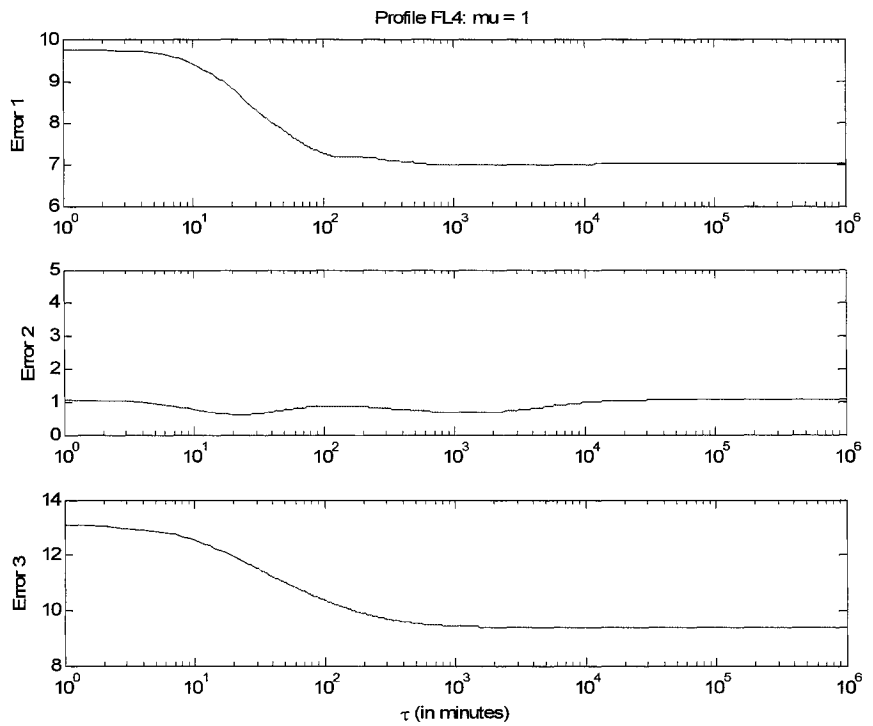


Figure 4.17: Error1, Error2, Error3 measured for user profile FL4 μ =1

A different result though was observed with higher desired mean values. For example, the mean of -9 for the user with profile FL10 shown in figure 4.18. Learning the volume settings too quickly $\tau < 50$ minutes or too slowly $\tau > 10^4$ minutes gives worse results for error 1 and error 3. The optimum range is between 200 to 2000 minutes. Extra care has to be taken in selecting the time constant for these kind of profiles.

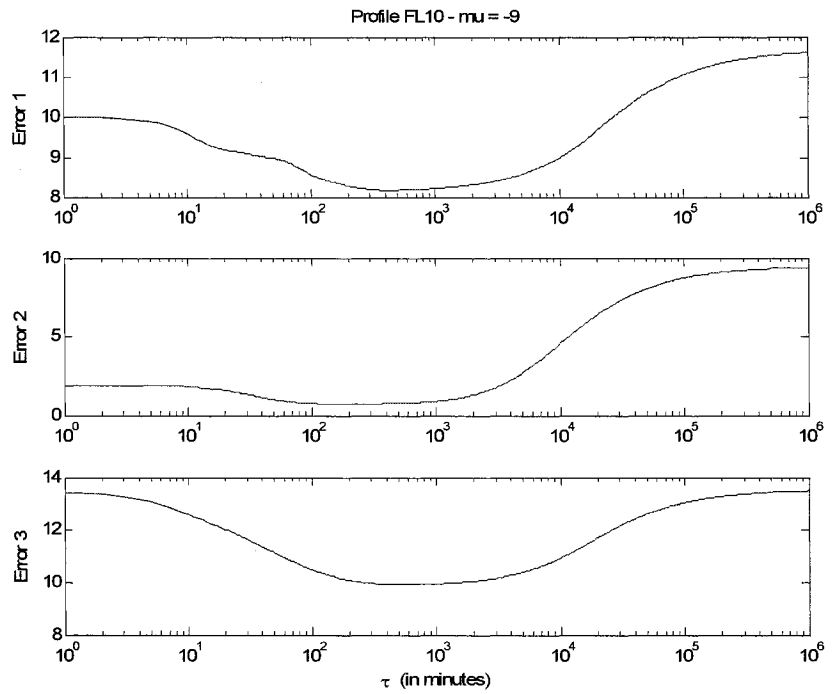


Figure 4.18: Error1, Error2, Error3 measured for user profile FL10 μ = -9

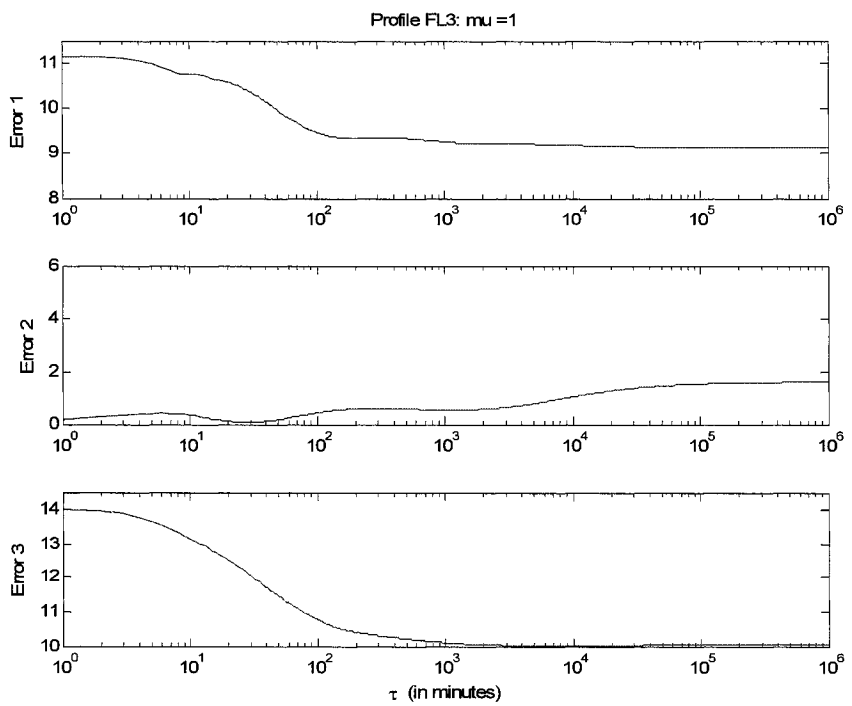


Figure 4.19: Error1, Error2, Error3 measured for user profile FL3 μ = 1

The maximum bias error is also higher for longer time constants, about 2dB for low means (see figure 4.19) and close to 10dB for higher means (see figure 4.18). This is because longer time constants will systematically overshoot or undershoot the users initial preferred settings since small weight is put on current observations.

With a fast time constant, the algorithm only remembers the last couple of values and this does not adequately represent the overall behavior. There are generally greater error1 and error3 values for shorter τ which decreases afterwards to the optimum τ. The bias error (error2) is generally close to zero but starts to increase for much slower time constant values.

For the non-homogenous profiles like the M (mixture of behaviors) and CP (changing preferences), there were greater differences for the time constants that gave minimum error values for the different generated profiles. The errors in this case are more sensitive to the value of the time constant. Figures 4.20 and 4.21 show the plots of the errors measured for two M profiles M17 and M25. Error1 is minimum in the range of 10^3- 10^4 minutes for profile M17 and 10^3 minutes for profile M25. In the CP plot in figure 4.23 for profile CP5 it is observed that this error is optimum in the range of 10^2-10^3 minutes. Bias errors of zero are obtained close to 100 minutes for M17 and M25 but this is not the case in CP5 which was optimum at 5100 minutes and M52 (see figure 4.22) with an optimum time constant of 10 minutes for this error.

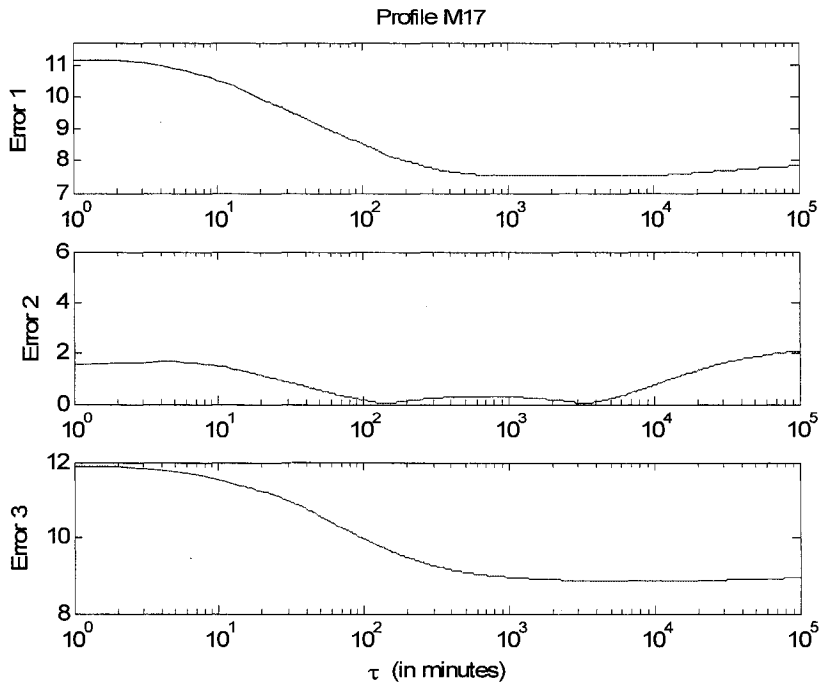


Figure 4.20: Error1, Error2, Error3 measured for user profile M17 (contains FF μ=-1 and FL μ=-2)

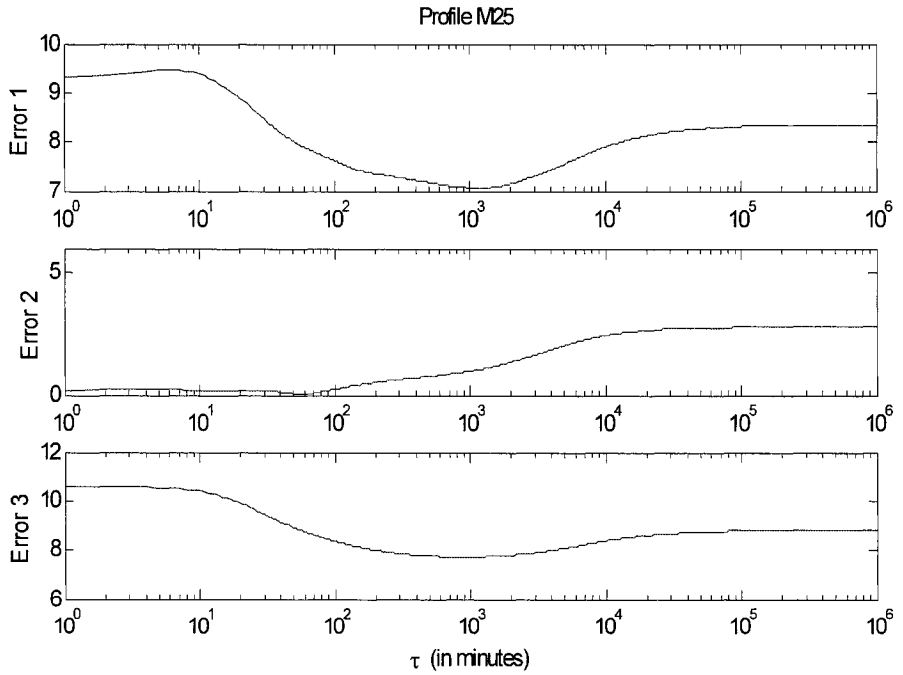


Figure 4.21: Error1, Error2, Error3 measured for user profile M25 (contains FL μ=-1 and FF μ = 6)

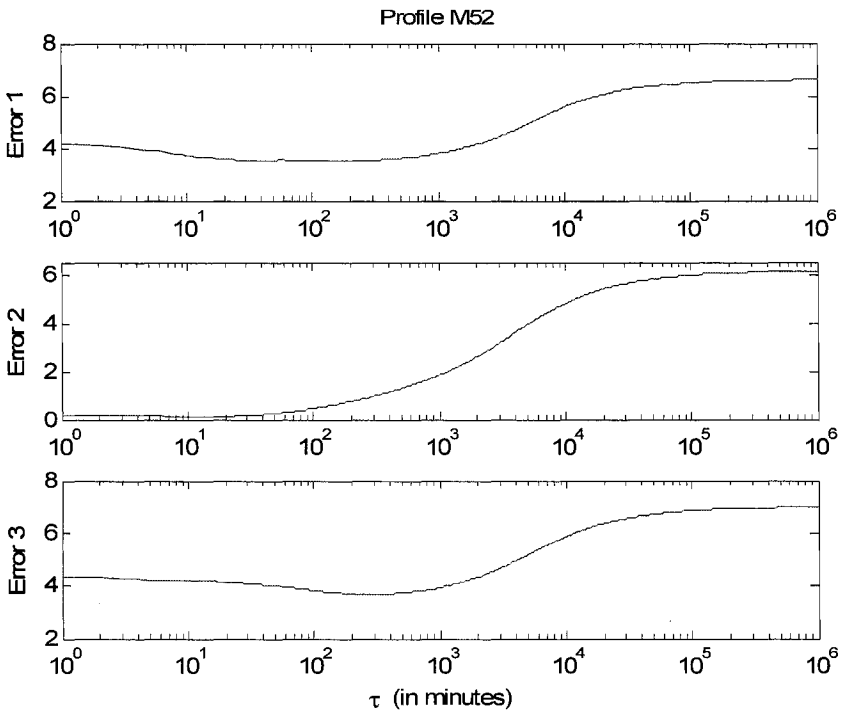


Figure 4.22: Error1, Error2, Error3 measured for user profile M52 (contains FC μ = -3, FF μ = -6 and FC μ = -9)

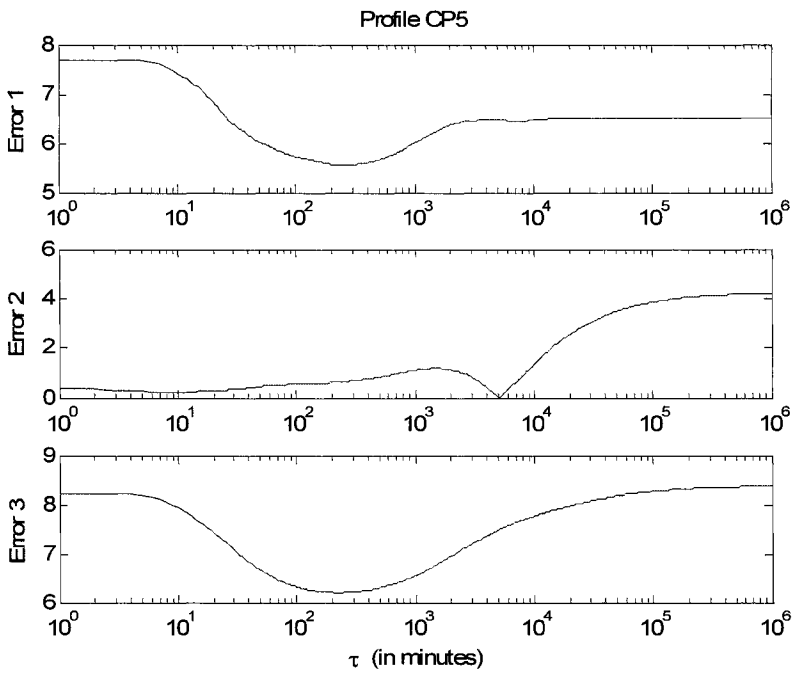


Figure 4.23: Error1, Error2, Error3 measured for user profile CP5 (contains FF μ = -9, FF μ = -10, FF μ = -4 and FF μ=5)

The short profiles also showed some variation in optimum time constant. In figure 4.24, the error measured for profile SHFL16 is shown. The optimum τ for error 1 is 500 minutes and error 2 is lowest at 100 minutes. However it was seen that for profile SHFF9 (see figure 4.25), longer time constants were found to give lower errors for error1 and error3. The bias error 2 is still very high for long τ values.

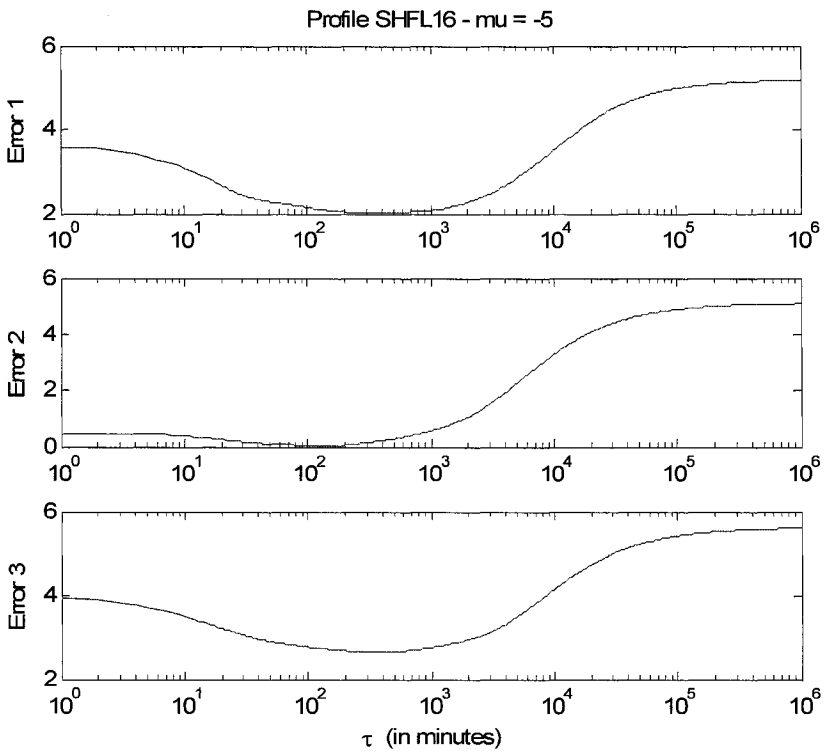


Figure 4.24: Error1, Error2, Error3 measured for user profile SHFL16 μ = -5

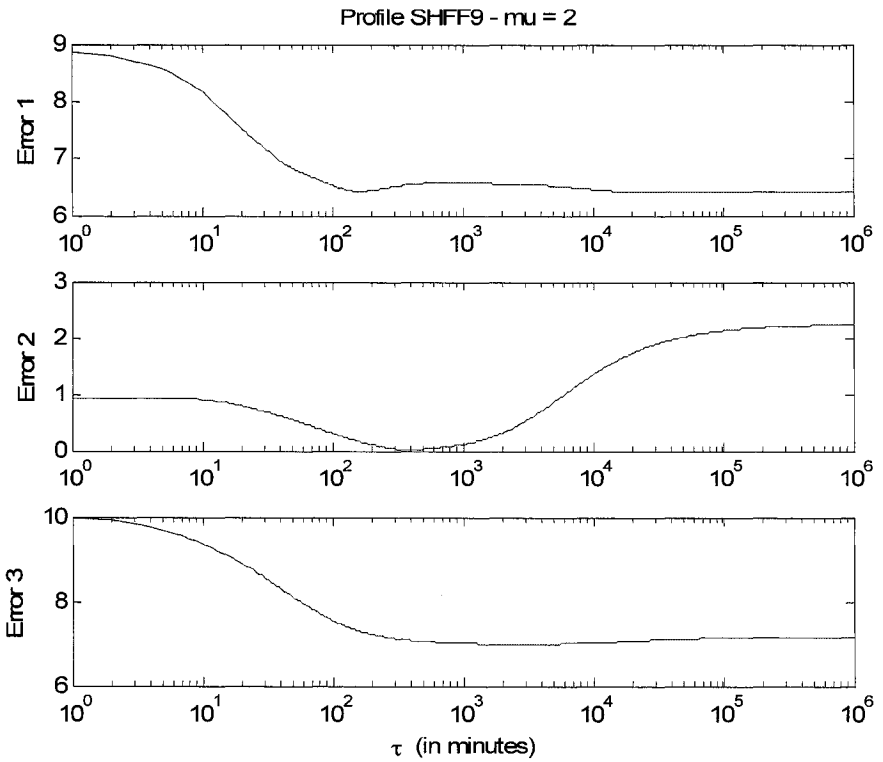


Figure 4.25: Error1, Error2, Error3 measured for profile SHFF9 $\mu = 2$

Generally, errors measured for the varying mean (VM) profiles were smallest for short τ values (< 100 minutes). Examples are shown in figure 4.26 and figure 4.27 for profiles VM5 and VM8 respectively. Here short time constants are preferred because the desired mean is constantly changing so the algorithm needs to put more weight on current values to capture the behavior of the series.

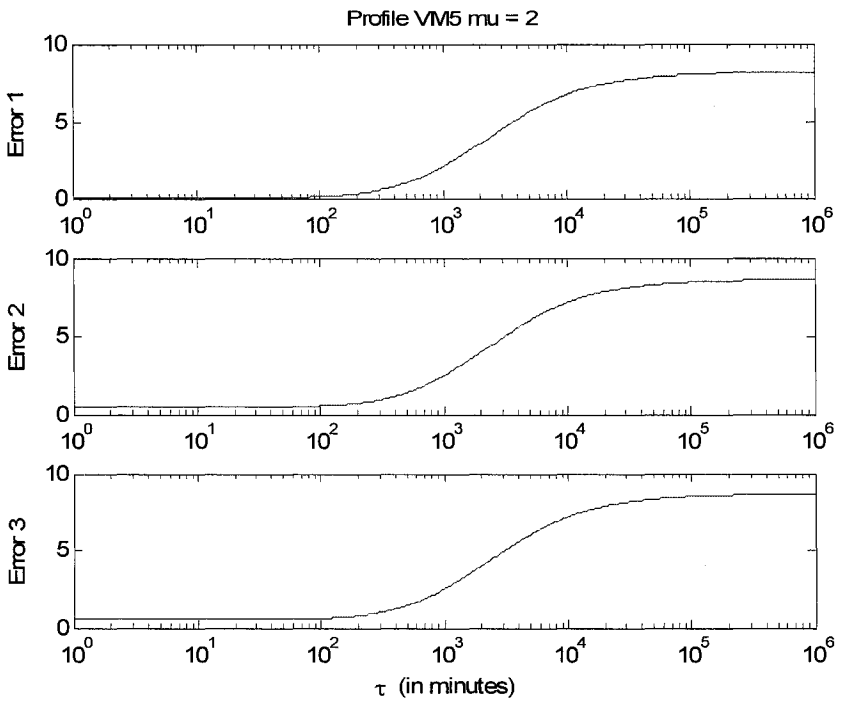


Figure 4.26: Error1, Error2, Error3 measured for profile VM5

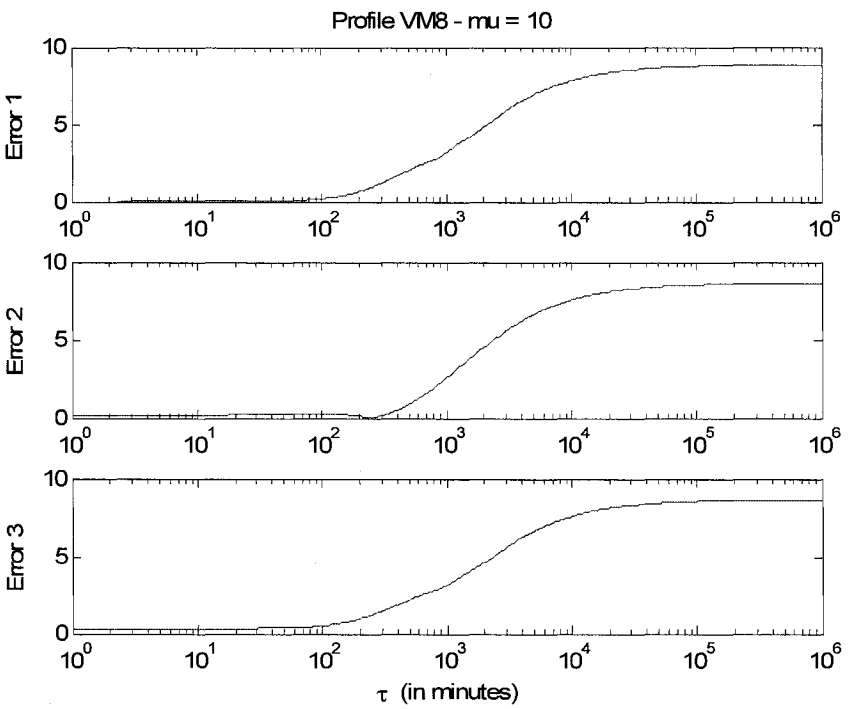


Figure 4.27: Error1, Error2, Error3 measured for profile VM8

To evaluate the performance of the adaptive algorithms when learning user preferences in extreme cases, a subset of profiles was selected from all those generated to cover collectively the whole τ range. The description of the profiles selected is summarized in table 4.3.

No.	Profile	Type	Number of Phases	Description	Optimum Time Constant τ (in minutes) for Error1
1	CP5	Changing Preference	40	Fairly Fluctuating μ = -9; μ = -10; μ = -4; μ =5	200 – 340
2	CP14	Changing Preference	20	Fairly Fluctuating μ = -10; μ =0; μ = -7; μ =7	500
3	CP15	Changing Preference	20	Fairly Fluctuating μ = -8; μ = 13; μ = -5; μ =7	1
4	CP18	Changing Preference	20	Fairly Fluctuating μ = -9; μ = 3; μ = -10; μ = 0	1897
5	CP19	Changing Preference	20	Fairly Fluctuating μ = -11; μ = 3; μ = 11; μ = 0	61 - 104
6	FF9	Fairly Fluctuating	60	Fairly fluctuating user μ = -3	10^3 - 10^6
7	FF11	Fairly Fluctuating	60	Fairly fluctuating user μ = -6	1000 - 3500
8	FF15	Fairly Fluctuating	60	Fairly fluctuating user μ = -8	184 - 1145
9	FF18	Fairly Fluctuating	60	Fairly fluctuating user μ = 6	569
10	FL2	Fluctuating	60	Fluctuating user μ =4	480 - 4200
11	FL15	Fluctuating	60	Fluctuating user μ = -7	426 - 1863
12	FL17	Fluctuating	60	Fluctuating user μ=10	1000
13	M10	Mixture of Behaviors	60	Fairly Fluctuating μ =-10; Fluctuating μ =0;	1658
14	M13	Mixture of Behaviors	60	Fluctuating μ =8; Fairly Fluctuating μ =0;	400
15	M15	Mixture of Behaviors	60	Fairly Constant μ =-5; Fairly Fluctuating μ = -10	246 - 1390
16	M25	Mixture of Behaviors	60	Fluctuating μ =-1; Fairly Fluctuating μ =6;	1280
17	M26	Mixture of Behaviors	60	Fluctuating μ =8; Fairly Fluctuating μ =-1	360 - 1000
18	M30	Mixture of Behaviors	60	Fairly Fluctuating μ =10; Fluctuating μ =2	390
19	M50	Mixture of Behaviors	60	Fluctuating μ =-9; Fairly Fluctuating μ =-6	2900
20	M51	Mixture of Behaviors	60	Fluc μ = 4; FF μ =-8	300 - 1000
21	VM8	Varying Mean	10	Decreasing values from 10 – 5	1 – 100
22	VM9	Varying Mean	10	Decreasing values from 12 – 6	1 – 100
23	SHFL4	Short – Fluctuating	60	μ=7;	400 - 3500
24	SHFL10	Short – Fluctuating	60	μ =1;	300 - 10^6
25	SHFC1	Short – Fairly Constant	60	μ = -7;	260

Table 4.3: Subset of profiles with optimum τ covering entire range of possible time constants

The previous simulations and observations allow us to identify the best time constant τ for every user. Clearly the optimum τ is different from one user to the other (see Table 4.3).

The adaptive algorithms need to start their learning with an initial time constant value. To determine what the initial conditions should be, the optimum time constant for learning all the profiles in table 4.3 was found. The optimum time constant was used in learning each profile and the average errors across profiles was determined. The average errors over all the profiles for a given time constant is plotted in figure 4.28. On average approximately 300 minutes would be optimum for minimizing error1 and error3, 30 – 400 minutes for error 2 and 469 minutes for error4. Also it could be observed that a time constant of 17 hours (1024 minutes) performs very close to the optimum and is a reasonable choice by some hearing aid manufacturers (see Section 2.5) for a single fixed time constant to use in learning user behaviors. The broad dip in figure 4.28 is expected because some users have fairly constant behaviors while others have fluctuating behaviors, the time constant of 17 hours is a meeting point for learning both these kinds of profiles.

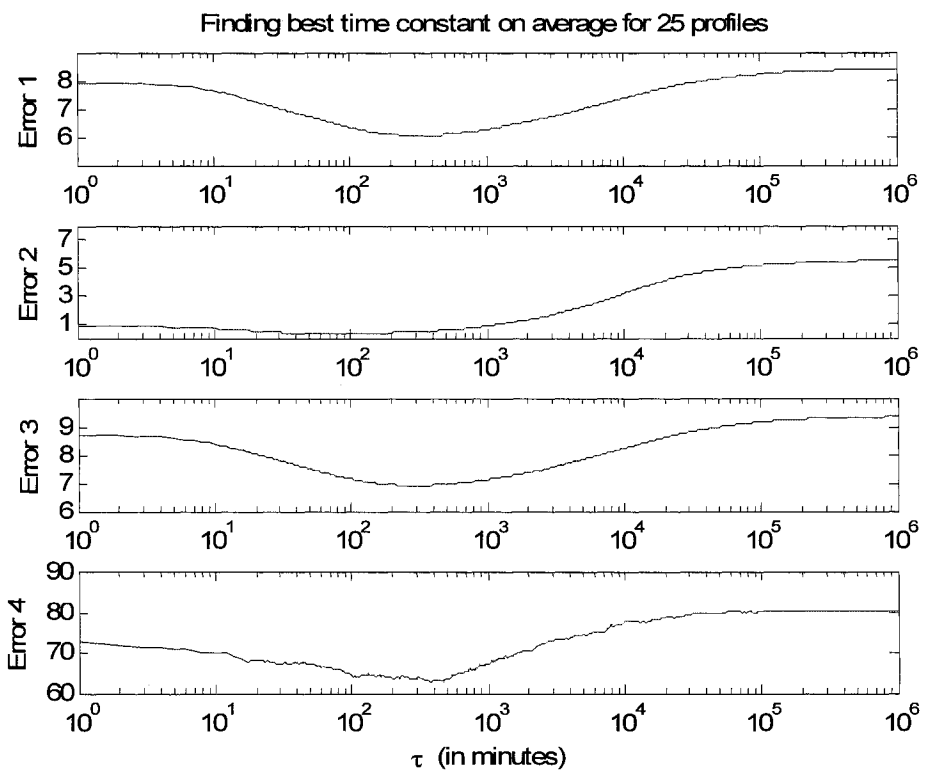


Figure 4.28: Finding optimum time constant on average for the 25 chosen profiles

4.4 Summary

Several error measures were devised to evaluate the performance of learning algorithms for tracking user preferences. From the analysis, it can be concluded that the optimum time constant depends on the user profile. While there is a "range" of optimum values, the range varies from one profile to the other. This demonstrates the need for adaptation of the time constant to match the user profile.

In assessing the errors proposed for measuring performance it was observed that error 1 (difference between the learned and first initial setting) and error 3 (RMS error) behave very similarly and therefore it is only necessary to consider one or the other. Error 2 (average error) is the most sensitive to the choice of a time constant over all the possible ranges for the different profiles. Error 4 (percentage of time within error margin) was still very large even at the optimum time constant. However, since error 4 it is a function of error 1 if the algorithm is able to minimize the latter error then error 4 would in turn be reduced as well. Therefore it was decided that error4 would not be investigated further. For the rest of this thesis, the algorithms will be evaluated based on error 1 and error 2.

Averaging results over the 25 profiles showed that the optimum time constant τ for error 1 was approximately 300 minutes and the range 30 – 400 minutes for error 2. The suitability of using a fixed time constant of 17 hours by some hearing aid manufacturers (as discussed in Section 2.5) is also confirmed. This choice seems higher than the optimum range found over the 25 profiles by a factor of approximately 4. However, the difference in error value is small enough that it is a reasonable choice for a fixed exponential smoothing algorithm and would suffice in minimizing the errors proposed.

Next, we will assess the performance of adaptive methods in comparison to fixed methods. Of particular interest is whether these adaptive methods can converge to the optimum time constants found manually in this chapter.

Chapter 5 Evaluation of Adaptive Algorithms

In the previous chapter, it was established that there is no single exponential time constant that would optimally learn the various types of user behaviors possible. In this chapter, we explore using the adaptive exponential smoothing algorithms discussed in Chapter 3 for learning user behavior. The algorithms are compared to one another and to the fixed adaptive algorithm based on the error1 and error2 measures discussed in Chapter 4. To suit the intended application in this work, the adaptive algorithms were modified in the manner that their time constant is updated. Also, in cases where the algorithm used other parameters, they were fine-tuned manually to find the best values for this application. The performance of these algorithms for an individual profile basis is first examined. Then the analysis is extended to investigate how the algorithms perform on average over many profiles. The time constant used in the adaptive methods was initialized to 17 hours (1024 minutes) as it was concluded in the analysis in Chapter 4 that this was a reasonable value for a fixed τ and because it is used in some hearing aids as discussed in Section 2.5 of this thesis.

5.1 Modification to the Adaptive Methods

The adaptive methods described in Chapter 3 were modified to update the time constant at the end of a phase instead of after every observed sample (i.e. at each time t) as was proposed by the original authors. This is to simulate the hearing aid adjusting the optimum τ after the user leaves an environment. Updating the time constant after each volume setting adjustment could also be too much processing for the hearing aid. Also, since the aim is to reduce user intervention and suggest a good volume setting when an environment is presented, the algorithms were updated to adjust τ based on error1, i.e. in a direction that minimizes this error. The methods evaluated in this chapter are those proposed by Chow [Section 3.2.1], Trigg [Section 3.2.3] and Taylor [Section 3.2.4]. The strategy proposed by Pantazopoulos [Section 3.2.2] was not considered because it led to unstable estimates.

5.1.1 Modification and fine-tuning of Adaptive Control of the Exponential Smoothing Constant

In this section three time constants τ, τ_U and τ_L are introduced, where the later two are defined in equations (5.1) and (5.2):

$$\tau_U = \tau + d \tag{5.1}$$

$$\tau_L = \tau - d \tag{5.2}$$

d is the step size in minutes. The corresponding smoothing constants α, α_U ,α_L are derived from τ, τ_U and τ_L using equation (3.1b). As discussed in Section 3.2.1, learning takes place with the three smoothing constants α, α_U ,α_L. Within a phase, fixed exponential smoothing takes place using the three constants as in equations (3.5) to (3.7). The last value calculated is taken as the learned/suggested value T_{i-1} for the next time the environment is presented i.e. phase i (see figure 4.6). At the start of the next phase, error1 is calculated based on the learned values obtained from the three constants as defined in equations (5.3a), (5.3b) and (5.3c):

$$error1\_\alpha_i = \left| T_{\alpha,i-1} - \underline{x}(1) \right| \tag{5.3a}$$

$$error1\_\alpha_{Ui} = \left| T_{\alpha_U,i-1} - \underline{x}(1) \right| \tag{5.3b}$$

$$error1\_\alpha_{l,i} = \left| T_{\alpha_L,i-1} - \underline{x}(1) \right| \tag{5.3c}$$

The corresponding constant τ that gives the minimum error1 value is then used for the main learning in this current phase and the secondary constants are updated according to equations (5.1) and (5.2) in Section 3.2.1. The learned value $T_{\alpha,i-1}$ from the main constant is also the one used in the calculation for $error2_i$ for phase i according to equation (5.4):

$$error2_i = \frac{1}{N} \sum_{j=1}^{N} (T_{\alpha,i-1} - \underline{x}_i[j]) \tag{5.4}$$

Fine-Tuning of parameter d

In this method, the parameter d is the step-size by which the time constant is decreased or increased during adaptation as defined in equations (5.1) and (5.2). To find the best value of d to use for an individual profile, it was varied from 0 – 550 minutes. The average error was calculated according to equation (4.5) using each d in the range. The value that gave the minimum error1 for that profile was deemed as the best one to use for learning the behavior for that user profile. In figure 5.1 error1 measured for profiles CP19, M50 and FF15 using various step-sizes is shown in the top panel and error2 measurements are shown in the bottom panel. It can be observed that d = 300 is the optimum for profile CP19 giving an error1 of 7.2. It should however be noted that the other d values performed very close to this optimum one, giving error1 between 7 and 8. For profile M50 the response was similar giving a very flat response. Many d-values in this case gave similar results. The best d value for profile FF15 is 410 minutes giving an error1 of 4.8. From these results it could be concluded that picking the right choice of d is not as critical as the error is fairly insensitive to d. In this work, wherever it was possible to pick an optimum d (e.g. for profile CP19 or FF15) it was used in the analysis. Otherwise, when no clear optimum could be seen (e.g. for profile M50) then an arbitrary value of d in the range was used. In the second panel of figure 5.1 where the results for error2 are shown, the best d value to minimize error2 when learning profile M50 is 500, CP19 is 90, 410 or 510 and finally d = 500 is best for profile FF15. For this method, the best value to use in the algorithm is dependent on the error we are minimizing and on the profile behavior. Also, in some cases this method is able to give a bias error of 0 (e.g. for profile CP19), something that is very desirable.

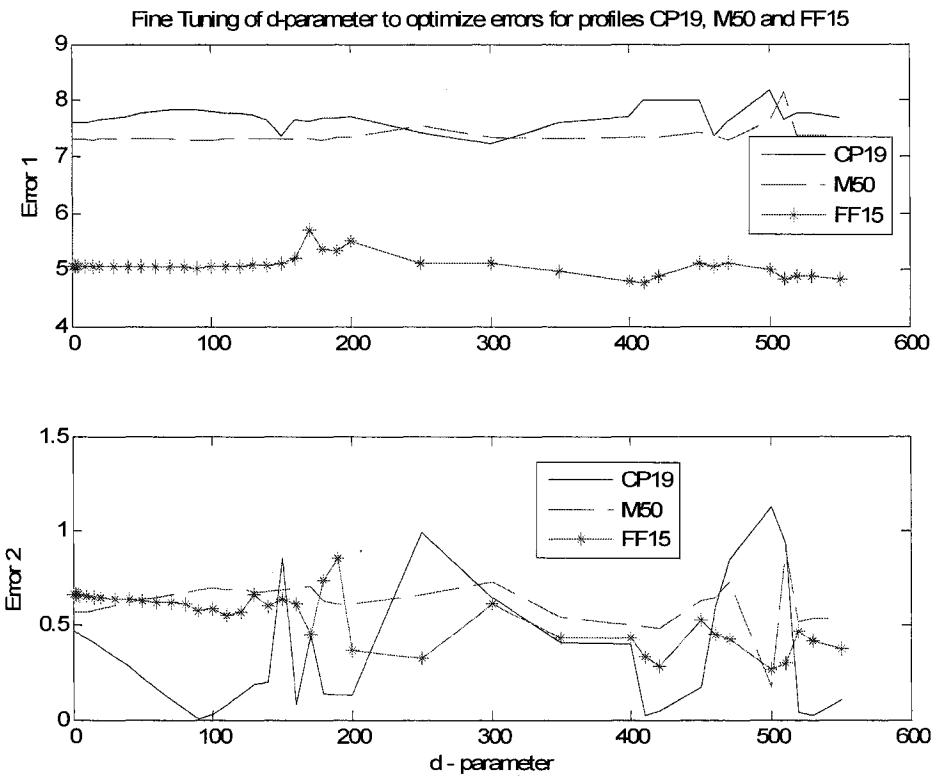


Figure 5.1: Fine-tuning of parameter d for individual profiles

To determine the best step-size d to use for learning many profiles, a given d in the range 0 - 550 is used in the learning of each profile and the error according to equation (4.5) is calculated for each profile and the average of all these errors was taken as the performance measurement for the d-value. The value that gave the best average error over all the profiles was taken as the best one. Figure 5.2 shows how various values for d performed when used in learning the behavior averaged over profiles CP19, M50 and FF15. It is clear that d has little impact on error1 and that d = 300 is a good choice over the three profiles. This is also observed in error 2 measurements where all d values gave errors less than 1.

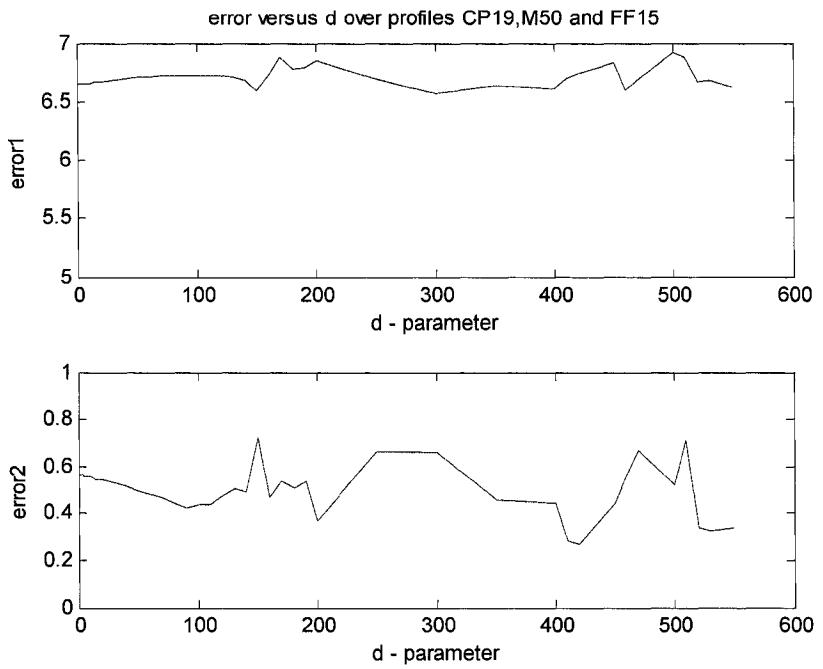


Figure 5.2: Fine-tuning d parameter over 3 profiles

5.1.2 Modification to Smooth Transition Exponential Smoothing

Here learning takes place with one smoothing constant α according to equation (3.1a). At the end of a phase, the last computed value is used as the learned value T_{i-1} in the next phase. The error1 value is then calculated as the transition variable V(t) as shown in equation (5.5) which is used in adapting the time constant as shown in equation (5.6):

$$V_i = \left| T_{\alpha,i-1} - \underline{x}_i[1] \right| \qquad (5.5)$$

$$\alpha_t = \frac{1}{1 + \exp(\beta + \gamma V_t)} \qquad (5.6)$$

Fine-tuning of parameters β and γ

The fine-tuning of the parameters for this method was more challenging because there are two parameters. The process to find the optimum values was to keep one of the parameters fixed while the other was varied. The fine-tuning was again done on both an individual profile basis and over many profiles. For an individual profile, β was varied from -2 to 2 and

γ was varied from -1 to 1. For a given profile, β was fixed to a value in the range [-2 2] and the error was computed based on equation (4.5) for all γ values. The best combination of β and γ giving the minimum error is identified. Figure 5.3 shows the results from tuning the parameters for profile M50. The plot shows the output for β fixed at 1.0 and 1.5 and γ varied in each case. For each β, error1 values obtained for variable γ is plotted. The results from using a small β were very erratic. The best combination giving the lowest error was when β = 1.5 and γ = 2.7.

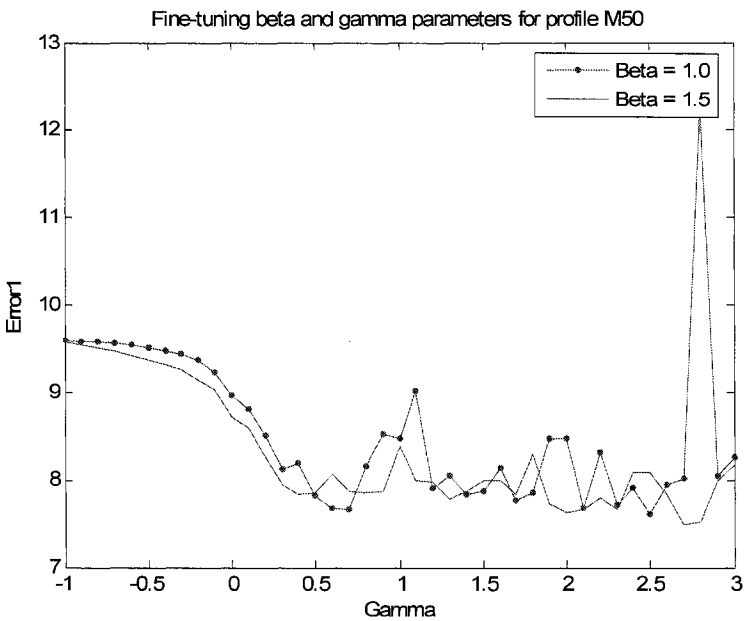


Figure 5.3: Fine-tuning Taylor parameters for profile M50

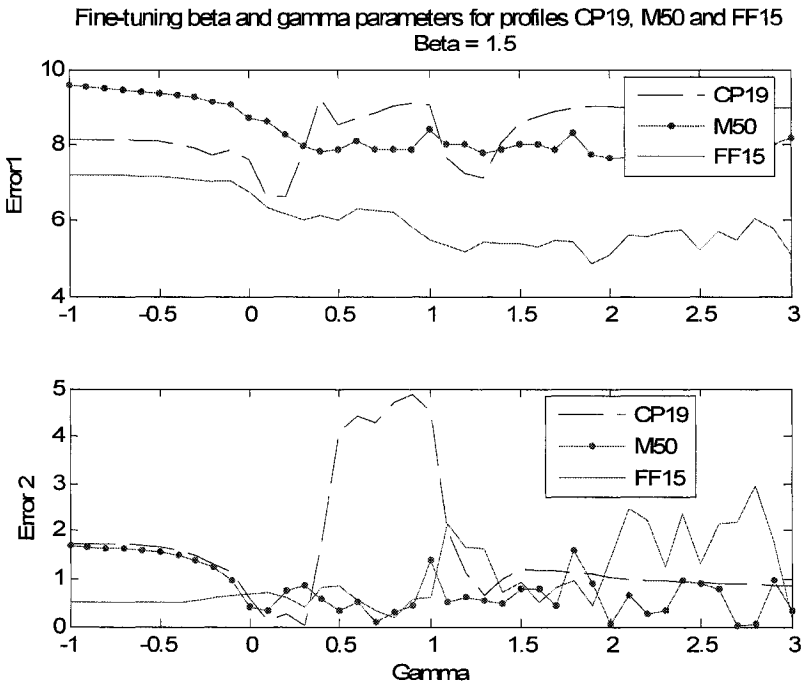


Figure 5.4: Fine-tuning of parameters β and γ for individual profiles

Figure 5.4 shows the fine-tuning for two other profiles CP19 and FF15 in addition to M50. In the top and bottom panels, errors versus γ were plotted for β fixed at 1.5. The Taylor method is seen to be more sensitive to the choice of its parameters compared to the Chow method. Profile CP19 has a minimum error1 at $\gamma = 0.2$ and error 2 is minimum at $\gamma = 0.3$. Profile FF15 has a minimum error1 at $\gamma = 1.9$ and error2 at $\gamma = 0.8$. Again this method was able to give zero error2 for some profiles, e.g. CP19 and M50.

To determine the best values of β, γ to use for learning many profiles, we need to average results over many profiles. Different combinations of β and γ were used for learning many profiles, error 1 and 2 were calculated for each profile and the average error over all the chosen profiles was taken. Figure 5.5 shows the fine-tuning of the parameters for learning three profiles CP19, M50 and FF15. The parameter β was fixed at -2. -1.5 and 1.5 with γ varying from -1 to 1. The result from the tuning is plotted in figure 5.5. The minimum error was found when $\beta = -1.5$ and $\gamma = 1.9$ or $\beta = 1.5$ and $\gamma = 1.3$.

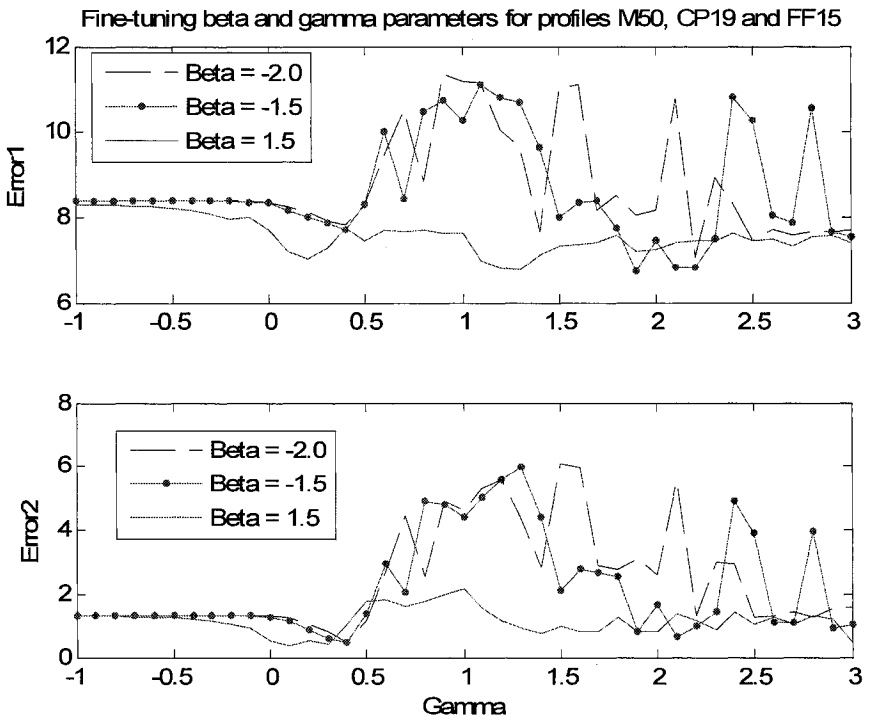


Figure 5.5: Fine-tuning of β and γ for profiles CP19, M50 and FF15

5.1.3 Modification to Exponential Smoothing with an Adaptive Response Rate

Learning takes place with one time constant α which is updated at the end of a phase instead of after every volume setting observed. Since the goal is to minimize error1 as defined in equation (4.5), this value was used in the calculation of the smoothed variables A_i and M_i as defined in equations (5.7) and (5.8). The corresponding tracking signal is defined in equation (5.9).

$$A_i = \phi \; error1_i + (1-\phi)A_{i-1} \qquad (5.7)$$

$$M_i = \phi \; |error1_i| + (1-\phi)M_{i-1} \qquad (5.8)$$

$$tracking\_signal = \left|\frac{A_i}{M_i}\right| \qquad (5.9)$$

Fine-tuning parameter ϕ

The parameter ϕ is the error smoothing constant and is used in the adaptation of the time constant α. To find the optimum ϕ value for an individual profile, ϕ was varied from 0 – 1 and the average error according to equation (4.5) using each ϕ in the range was calculated. The value that gave the minimum error for that profile was deemed as the best one to use for learning the behavior for that user profile. In the top panel of figure 5.6 the error1 measured for profiles M50, CP19 and FF15 using various ϕ values is shown. The best value for minimizing this error for profile M50 is 0.45, 0.1 for profile FF15 and 0.25 for profile CP19. In general higher values of ϕ only increased the error. For error2, ϕ = 0.05, 0.1 and 0.05 are best for minimizing this error in profiles FF15, CP19 and M50 respectively.

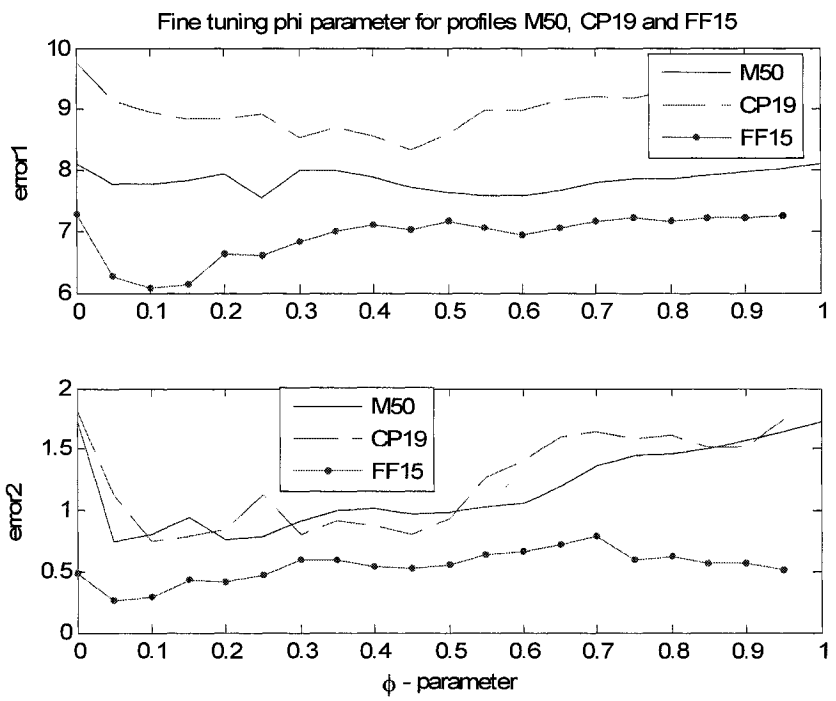


Figure 5.6: Error measures as a function of phi for individual profiles M50, CP19, FF15

To determine the best ϕ to use for learning many profile, a given ϕ in the range 0 -1 was used in learning of each profile and the error according to equation (4.5) is calculated for each profile and the average of all these errors is taken as the performance measurement for that ϕ. The value that gave the minimum average error over all the profiles was seen as the

best one. Figure 5.7 shows how various values of ϕ performed when used in averaging over profiles CP19, M50 and FF15. It is seen that $\phi \sim 0.1, 0.15$ gave the least error.

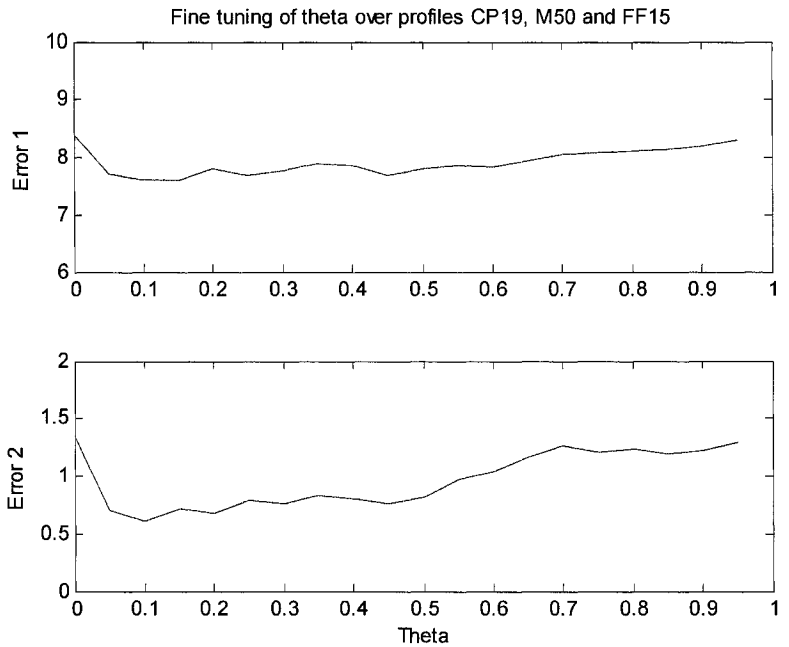


Figure 5.7: Fine-tuning theta parameter over 3 profiles

Using the processes outlined for finding the best parameter values to use in the adaptive algorithms, the best parameters on average for minimizing error 1 and 2 for the 25 profiles selected in table 4.3 of Chapter 4 are summarized in table 5.1.

		Parameter value minimizing Error 1	Parameter value minimizing Error 2
Chow	d =	400	450
Trigg & Leach	φ =	0.05	0.05
Taylor	β =	2.0	2.0
	γ =	0.2	0

Table 5.1: Summary of best parameters for learning the 25 chosen profiles given in table 4.3

5.2 Comparison of Adaptive Algorithms over Individuals profiles

In Chapter 4, the optimum time constants and their corresponding error values have been identified by direct search. In Section 5.1, the values of the control parameters in the adaptive learning algorithms were fine-tuned. We are now ready to verify if the adaptive methods would be able to perform as well as the optimum for individual profiles. The time constant value for each method was initialized to 1024 minutes and the best parameters identified in table 5.1 of Section 5.1 were used. The performance of the three adaptive was evaluated when learning individual profiles. The outcome of the methods was also compared against using fixed time constant values in the exponential smoothing equation.

The results from processing three profiles CP19, M50 and FF15 will be discussed first. The "fine-tuned" parameters for these profiles are summarized in table 5.2. These three specific profiles are used in this section because the results obtained from their analysis captures the summary of how the adaptive algorithms learn different user behaviors. Also, in order to compare to using a fixed time constant, the optimum time constant for the profile as identified in Chapter 4 is also given in table 5.2 along with the corresponding minimum value for error1. In addition to the best time constant, other fixed time constant values included for comparison are 1024 (17 hours), 10 and 10000 minutes to cover the full range of τ.

Profile	Chow: Best d	Trigg: Best ϕ	Taylor: Best β,γ	Optimum τ for error 1	Minimum Error 1
CP19	300	0.25	2.0,0.1	77	6.6
M50	180	0.45	1.5,2.7	3000	7.1
FF15	410	0.1	1.5,1.9	293	4.9

Table 5.2: Summary of tuned parameters

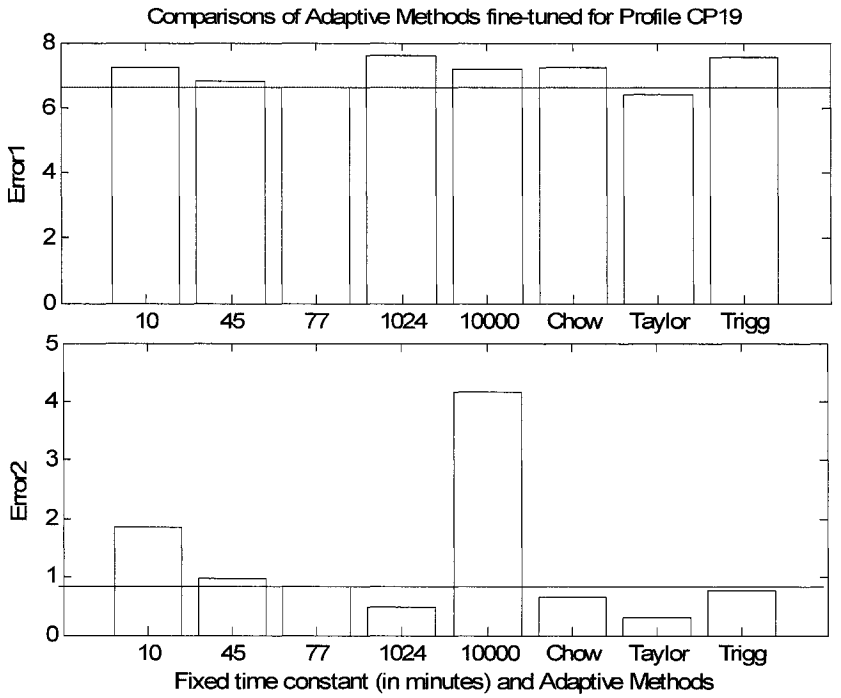


Figure 5.8: Comparison of adaptive methods for Profile CP19

For all adaptive algorithms, the initial value used for τ is 1024 minutes. In figure 5.8 the error1 was measured after learning the behavior of profile CP19 with fixed time constants of 10, 45, 77 (which is the optimum for this profile), 1024 and 10000 minutes as well as the three adaptive methods. The horizontal line across is the minimum error produced by the optimum fixed time constant 77 minutes. It can be observed that the method by Taylor behaved slightly better than the optimum for error1. However, for this profile the other two methods did not perform as well as the optimum time constant. Also, the method by Taylor was able to minimize error 2 the best followed by the fixed time constant of 1024 minutes.

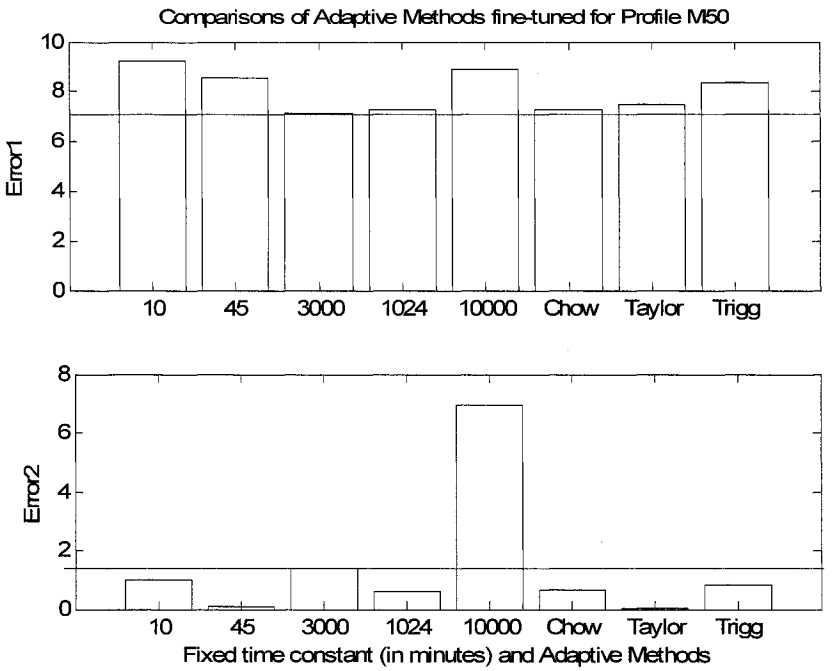


Figure 5.9: Comparison of adaptive methods (Error 1) for Profile M50

In figure 5.9, the learning of profile M50 is shown. The Chow and Taylor methods performed closest to the optimum time constant of 3000 minutes for error1. Like in the previous profile, the method by Trigg and Leach did not do a good job in minimizing error1. It is also worthwhile to note that a fixed time constant of 1024 minutes behaves close to the optimum. In the bottom panel it is shown that the three adaptive methods did better than the optimum of 3000 minutes for minimizing error2.

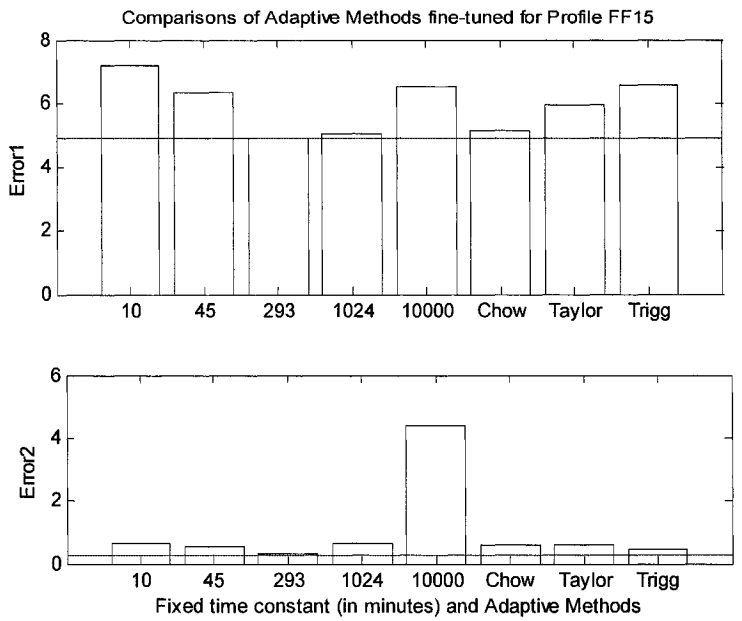


Figure 5.10: Comparison of adaptive methods for Profile FF15

The results from learning the FF15 profile is shown in figure 5.10. The Chow algorithm was the closest to the optimum.

All other profiles listed in table 4.3 where processed in the same manner. The Chow algorithm was found to consistently perform best giving an error closest to that produced by the optimum time constant for the specific profile. The outcome is summarized in table 5.3.

Method	Number of profiles the method performed closest to the optimum fixed constant
Chow	20 out of 25
Taylor	3 out of 25
Trigg & Leach	2 out of 25

Table 5.3: Summary of performance by adaptive methods over the 25 chosen profiles

The simulations show that of the three adaptive algorithms (with their parameters fine-tuned for best performance), the Chow method performs the best and is able to learn hearing aid volume control settings as well as the optimum time constant for a given individual.

5.3 Comparison of Adaptive Algorithms over all profiles

In practice it would be time consuming and difficult to tune the parameters in the adaptive algorithms. In practice, an average value over all profiles would probably be used. In Chapter 4, an analysis over 25 varying profiles showed the average optimum time constant to be 313 minutes. The results from that analysis are shown in figure 5.11 for errors 1 and 2.

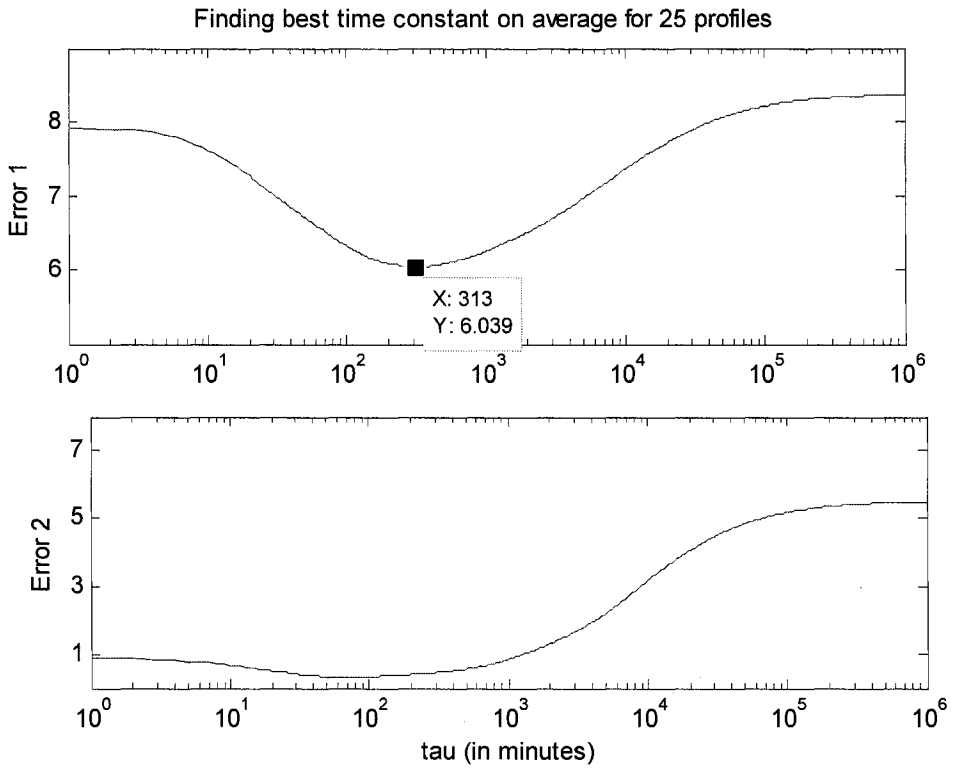


Figure 5.11: Summary Plot over the 25 chosen profiles finding optimum fixed time constant

Using the average parameters in table 5.1 that were found to be best for learning the 25 profiles, the profiles chosen at the end of Chapter 4 were processed to see how robust the parameters are. It is expected that the adaptive algorithms would perform at least as well as the average optimum time constant. In this case the adaptive algorithms were initialized to 313 minutes in line with the results obtained in figure 5.11. The errors obtained from learning the 25 profiles using the adaptive methods and fixed time constants of 313 and 1024 minutes are shown in figure 5.12. Overall the Chow method performed just as well as the optimum average of 313 minutes followed by the Taylor method and then the method by

Trigg and Leach. The performance when using a fixed time constant of 1024 minutes is also quite reasonable, again validating its use by some manufacturers (Section 2.5).

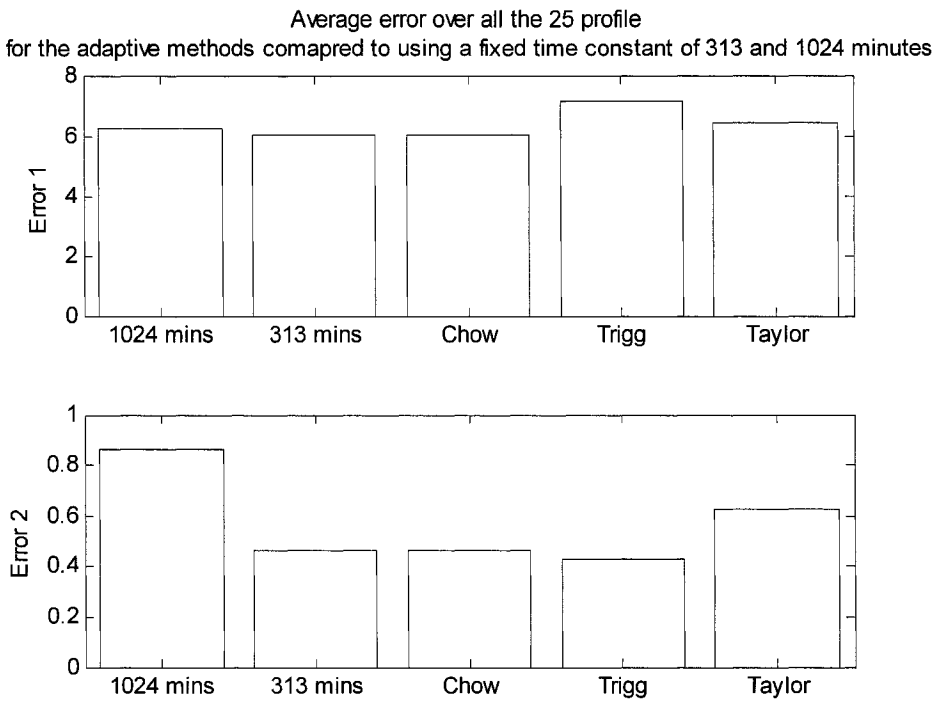


Figure 5.12: Average Errors over all the 25 profiles

Table 5.4 shows the breakdown for the error1 value obtained using the adaptive methods for each of the 25 profiles to see how the average parameters work for individual profiles. The difference between the error from the adaptive method and using a fixed time constant of 313 minutes was calculated. At first glance it can be observed that the Chow method gave the lowest error values based on the number of negative signs for the calculated difference values e.g. CP5, CP18, FF11 etc.. Also Chow performed just as well as the optimum (313minutes) giving identical results for example in CP19, VM8 andVM9. The Trigg and Leach method appears to have larger differences e.g. as high as 4.01 for profile FL15. The sums and standard deviations of these errors for each method are summarized in table 5.5. It is observed that the method by Chow performs a little better than the optimum (sum of errors is negative). Also from the standard deviation we can conclude that this method performs fairly well over all profiles. The method by Trigg and Leach is the furthest from the optimum with a total difference of 27.63. Compared to a using a fixed time constant of

313 minutes, using 1024 minutes did not do better than the Chow method but it did have a sum (in table 5.5) lower than the result from the method by Taylor. However, the standard deviation of the difference obtained from using this fixed time constant over all profiles suggests that there are some profiles where it does not perform well like VM8 and VM9 which need really short time constant because the behavior is changing fast.

The standard deviation of the difference also shows that the Taylor method behaves about the same over all the profiles unlike the Trigg and Leach method whose performance was quite variable over the profiles.

	Error1					Difference between method and fixed time constant of 313 minutes (delta)			
Profile	Fixed -1024 mins	Fixed - 313 mins	Chow	Tay	Trigg	Fixed - 1024 mins	Chow	Tay	Trigg
CP5	6.00	**5.56**	5.49	6.00	6.64	0.44	-0.07	0.44	1.08
CP14	8.13	**7.77**	7.80	8.70	10.29	0.37	0.04	0.93	2.52
CP15	7.89	**7.11**	7.11	7.74	6.67	0.77	0.00	0.63	-0.44
CP18	5.89	**6.33**	5.98	6.25	6.30	-0.44	-0.35	-0.07	-0.03
CP19	7.59	**7.48**	7.55	7.09	7.76	0.12	0.08	-0.39	0.29
FF9	5.84	**5.81**	5.82	6.54	7.71	0.03	0.01	0.73	1.90
FF11	5.47	**5.65**	5.52	6.20	6.15	-0.18	-0.12	0.56	0.51
FF15	5.05	**4.91**	4.97	5.71	6.25	0.14	0.06	0.80	1.35
FF18	5.10	**5.16**	5.06	5.89	7.73	-0.06	-0.10	0.73	2.57
FL2	9.16	**9.30**	9.08	9.65	10.62	-0.14	-0.22	0.35	1.32
FL15	8.02	**8.07**	7.96	8.31	12.08	-0.04	-0.11	0.24	4.01
FL17	7.53	**7.75**	7.70	8.33	8.66	-0.22	-0.05	0.58	0.91
M10	6.37	**6.58**	6.58	6.82	7.90	-0.22	-0.01	0.23	1.31
M13	7.63	**7.37**	7.34	7.80	9.15	0.27	-0.02	0.43	1.78
M15	2.77	**2.76**	2.73	3.11	3.20	0.01	-0.03	0.35	0.44
M25	7.07	**7.27**	7.25	7.43	7.76	-0.21	-0.03	0.16	0.48
M26	6.12	**6.10**	6.15	7.10	8.37	0.02	0.05	1.00	2.27
M30	6.66	**6.46**	6.51	7.56	8.73	0.20	0.05	1.11	2.27
M50	7.31	**7.77**	7.67	8.03	9.15	-0.46	-0.10	0.25	1.38
M51	6.46	**6.36**	6.45	6.14	7.12	0.10	0.09	-0.22	0.76
VM8	3.18	**1.24**	1.24	1.02	0.82	1.94	0.00	-0.21	-0.41
VM9	4.80	**2.00**	2.00	2.12	1.11	2.80	0.00	0.12	-0.89
SHFL4	5.74	**5.85**	5.86	5.88	6.70	-0.12	0.00	0.03	0.85
SHFL10	7.79	**8.00**	7.99	8.25	8.87	-0.21	-0.01	0.25	0.87
SHFF1	2.58	**2.32**	2.32	2.64	2.86	0.25	0.00	0.32	0.54

Table 5.4: Error 1 for all three adaptive algorithms for each of the 25 chosen profiles in table 4.3

	Fixed - 1024 mins	Chow	Tay	Trigg
sum	5.17	-0.84	9.35	27.63
stdev	0.72	0.10	0.39	1.10

Table 5.5: Summary of average value of error 1 for the 25 chosen profiles in table 4.3

Table 5.6 shows the breakdown for the error 2 values obtained using the adaptive methods for each of the 25 profiles. The difference between the error from the adaptive method and using a fixed time constant of 313 minutes is also shown.

	Error2					Difference between method and fixed time constant of 313 minutes (delta)			
Profile	Fixed - 1024 mins	Fixed - 313 mins	Chow	Tay	Trigg	Fixed - 1024 mins	Chow	Tay	Trigg
CP5	1.11	**0.68**	0.66	0.63	0.12	0.43	-0.02	-0.05	-0.56
CP14	1.21	**0.86**	0.55	1.61	0.72	0.34	-0.31	0.75	-0.14
CP15	1.91	**1.60**	1.60	1.99	0.47	0.31	0.00	0.39	-1.13
CP18	0.48	**0.24**	0.38	0.14	0.66	0.24	0.14	-0.10	0.42
CP19	0.48	**0.83**	0.90	0.21	0.75	-0.35	0.08	-0.62	-0.07
FF9	0.14	**0.24**	0.17	0.32	0.01	-0.10	-0.07	0.08	-0.23
FF11	0.74	**0.40**	0.52	0.00	0.11	0.34	0.12	-0.40	-0.29
FF15	0.66	**0.31**	0.26	0.32	0.25	0.35	-0.05	0.01	-0.05
FF18	0.82	**0.51**	0.54	0.32	0.54	0.31	0.03	-0.19	0.03
FL2	0.81	**0.51**	0.47	0.62	0.26	0.30	-0.04	0.11	-0.25
FL15	0.44	**0.23**	0.10	0.91	0.92	0.20	-0.14	0.67	0.69
FL17	0.65	**0.08**	0.26	0.11	0.01	0.57	0.18	0.03	-0.07
M10	0.06	**0.03**	0.08	0.55	0.54	0.03	0.05	0.51	0.50
M13	0.14	**0.13**	0.08	0.40	0.17	0.01	-0.05	0.27	0.04
M15	0.83	**0.36**	0.45	0.29	0.07	0.47	0.09	-0.07	-0.29
M25	1.00	**0.65**	0.68	0.14	0.78	0.35	0.03	-0.51	0.13
M26	0.50	**0.26**	0.32	0.34	0.28	0.24	0.07	0.08	0.02
M30	0.22	**0.56**	0.22	1.59	0.97	-0.34	-0.34	1.03	0.40
M50	0.56	**0.17**	0.35	0.79	1.13	0.39	0.18	0.62	0.95
M51	0.93	**0.67**	0.82	0.58	0.08	0.25	0.14	-0.10	-0.60
VM8	2.63	**0.20**	0.20	0.75	0.61	2.43	0.00	0.56	0.41
VM9	3.88	**1.05**	1.05	1.75	0.76	2.84	0.00	0.70	-0.29
SHFL4	0.13	**0.34**	0.23	0.52	0.10	-0.21	-0.11	0.19	-0.24
SHFL10	0.35	**0.42**	0.56	0.49	0.46	-0.07	0.14	0.07	0.04
SHFF1	0.86	**0.21**	0.21	0.30	0.08	0.65	0.00	0.09	-0.13

Table 5.6: Error 2 for all three adaptive algorithms for each of the 25 chosen profiles in table 4.3

For minimizing this error, the method by Trigg performed best of all the adaptive methods being able to lower this error better than the optimum 14 out of 25 profiles. This is expected because this method aims to reduce biased estimates. The other 2 adaptive methods performed similarly, being better than the optimum for 9 and 8 out of the 25 profiles for Chow and Taylor respectively. In the results summarized in table 5.7, using a fixed time constant of 1024 minutes performed poorly having the highest sum and standard deviation because this time constant is too long to capture changes in the observed volume control settings. Overall the results of the sum and standard deviation of the difference for Chow are favorable.

	Fixed - 1024 mins	Chow	Tay	Trigg
Sum	9.99	0.11	4.14	-0.72
STD	0.72	0.13	0.41	0.44

Table 5.7: Summary of average value of error 2 for the 25 chosen profiles in table 4.3

5.4 Summary

In this chapter, we examined the performance of three learning algorithms with adaptive time constants. We started by tuning the parameters for these algorithms. The adaptive methods proposed by Chow and Trigg and Leach are easy to tune. Also the parameter d in the Chow method is very robust and does not affect the outcome of the algorithm in this application. When the parameters of the adaptive methods were tuned to each individual profile, the method by Chow was able to perform as best as the optimum time constant for most profiles. When average parameters are used in the algorithms, the Chow method performed best of the three for minimizing error1. The method by Trigg performed best for minimizing the bias error 2 followed by the Chow method. The Taylor method is somewhere in between these two methods. Because fine-tuning its parameters poses a challenge and because its performance is surpassed by the other approaches, this method is not recommended.

The simulations carried out show that while a fixed time constant of 1024 minutes for all users provides a reasonable choice for the time constant used to learn the user preferences, it is not the optimum value. We also show that without having prior knowledge of a person's behavior, using an adaptive algorithm with good initialization, optimum hearing aid volume control preferences can be learned.

Chapter 6 Conclusions and Future Directions

Hearing aid users have often complained about the frequency at which they need to adjust their hearing aid settings. This issue is due in part to the clinical environments in which hearing aid gains are prescribed. It is difficult to simulate and optimize the prescribed gains for all possible real-world acoustic environments that could be presented to the user as they go about their daily activities. Research has also shown that different environments require different hearing parameters [4]. Therefore, it would suffice to suggest that fine-tuning should be specific to a given environment and be an on-going process in the real world based on feedback from the hearing aid wearer.

Hence, the idea of self-learning hearing aids was proposed. Such hearing aids can be trained by the user to automatically learn settings they prefer in different environments. Learning invariably involves tracking user behavior for different hearing aid settings. Currently a fixed time constant is often used for learning in hearing aids currently on the market.

6.1 Conclusions

One of the questions posed at the beginning of this work was whether a single fixed time constant could be adequate in learning hearing aid user volume control preferences for all users. Since users have different behaviors it was hypothesized that this might not be the case. From the simulations performed in this work it has been shown that there is not a single optimum time constant that would optimize learning for all users. The optimum range of values for volume setting is a function of the user behavior. This finding prompted the investigation of whether adapting the time constant to the user would give better performances for individual users.

The analysis carried out in Section 4.3.2 validates that the use of a time constant of 17 hours [Section 2.5] in the fixed exponential smoothing algorithm as used in some hearing aids in the market is a reasonable choice.

Three adaptive exponential smoothing methods were presented and evaluated using a bank of generated user profiles. After tuning the parameters for these methods, it was shown that the Chow method [38] performed the best over a wide range of individual profiles. Compared to using a fixed time constant of 17 hours, the improvement for these methods is not significant on average over all profiles. There are however some specific cases where the adaptive methods performed significantly better than a fixed time constant. An example is in the varying mean profiles. The method by Trigg and Leach does a good job of making sure the estimates are not biased and are adapting to changes in the observations. In some cases the adaptive methods are able to minimize the difference between the learned value and the first initial setting, achieving performance as good as the optimum time constant. The Chow method in particular was shown to work well for all types of behaviors.

Based on the results reported in this thesis we recommend the Chow method for learning user volume control preferences in hearing aids as it is robust enough to learn different user behaviors, its parameters are simple to tune and it is a fairly simple algorithm to implement.

6.2 Future Directions

This thesis focused on learning the user preferences with the assumption that environment classification had been done accurately and learning is based solely on the user's behavior in the environment. An extension to this project would be to map the acoustic features of the environment to the past behaviors and incorporate that into the learning process. This should allow for improved learning because the system can now learn the correlation between the actual input signal and a change in the setting. This will also allow the use of signal processing methods like Kalman filtering and Bayesian learning since there is access to the input signal.

This thesis focused only on "level changes" and not "spurious spikes" in the data. It would be interesting to expand the simulation of user behavior to include spikes so as to capture cases where the user is fiddling with the volume control. Some algorithms like the one by

Trigg and Leach can be tuned to be insensitive to spikes and may prove to be useful in this respect.

The data used in this work was modeled by assuming each user has a desired mean and different standard deviations to depict different user behaviors. These behaviors were also combined to give other secondary behaviors. The challenge would be to collect real volume control settings from hearing aids in use. This is a big undertaking and would take a substantial time investment to collect a significant amount of data. In the mean time, an intermediate step in the development stage would be to use some already established objective measures like the Speech Intelligibility Index to simulate the expected user behavior under the assumption that the user will adjust the settings to enhance speech intelligibility or some other objective measure.

In addition, this work is part of a larger intelligent hearing aid project where the device should be able to identify what environment the user is in, load the appropriate signal processing algorithms and learn the control preferences. The long term plan is to integrate the work done in this thesis with that of another M.ASc student Luc Lamarche [45] who has been working on dynamic environment classification. The environments classes can split and merge according to what the user is exposed to during the course of their everyday life.

References

[1] I. Blood, "The hearing aid effect: Challenges for counseling," *Journal of Rehabilitation*, vol. 63, pp. 59-63, 1997

[2] M.B. Jennings, "Hearing Rehabilitation for Older Adults: An Update on Hearing Aids, Hearing Assistive Technologies, and Rehabilitation Services," *Geriatrics Aging*, vol. 9, no. 10, pp 708-711, 2006

[3] S. Kochkin, "MarkeTrak V: "Why my hearing aids are in the drawer": The consumers' perspective, " *The Hearing Journal*, vol. 53, no. 2, pp. 34-42, February 2000

[4] J.M. Kates, "Classification of background noises for hearing-aid applications," *Journal of Acoustical Society of America,* vol. 97, no. 1, pp. 461- 470, 1995

[5] H. Dillon, *Hearing Aids.* New York: Thieme Medical Publishers, 2001

[6] P. Nordqvist and A. Leijon "An efficient robust sound classification algorithm for hearing aids," *Journal of Acoustical Society of America* vol. 115, no.6, pp. 3033-3041, 2004

[7] H. Dillon, J. A. Zakis, H. McDermott, G. Keidser, W. Dreschler, E. Convery, "The trainable hearing aid: What will it do for clients and clinicians?," *Hearing Journal*, vol. 59, pp. 30-36, April 2006

[8] R. Meredith, "Volume Control Preferences with WRDC Digital Hearing Aids," *The Hearing Review*, Vol. 13, No. 7, pp. 24-28, July 2006

[9] Siemens, "DataLearning & Learning VC : Learning Preferred Gain Settings in Everyday Life," <http://mysiemens.siemens-hearing.com/admin/documents/08139.pdf>

[10] Internet source < *http://cobweb.ecn.purdue.edu/~ee649/notes/figures/ear.gif*> (March 2008)

[11] J. O. Pickles, *An introduction to the physiology of hearing*, Toronto: Academic Press, 1988

[12] K. Lysons, *Understanding Hearing Loss*, Pennsylvania : Jessica Kingsley Publishers, 1996

[13] R.J. Roeser, K.A. Buckley and G.S. Stickney, "Pure Tone Tests, Audiology: Diagnosis." Ed. R.J. Roeser, M. Valente, H. Hosford-Dunn. New York: Thieme Medical Publishers, pp. 227-251, 2000.

[14] R. Gao, Y. Liu, S. Basseas, and L.H. Tsoukalas, "Next generation hearing aid devices," in *Proc. 11th Int. Conf. Tools Artificial Intelligence* Chicago, IL, pp. 327-331, 1999.

[15] T.H. Venema, *Compression for Clinicians*: Canada, Thomson Delmar Learning, 2006

[16] R.E. Sandlin, Hearing Aid Amplification: Singular Publishing Group, 2000

[17] A. Chankawee and N. Tangsangiumvisai, "On the improvement of acoustic feedback cancellation in hearing-aid devices," *The 47th International Midwest Symposium on Circuits and Systems (MWSCAS'04, Hiroshima, Japan),* July 2004

[18] Internet Source: <http://www.ee.ucla.edu/~spapl/rod/aeptf.htm> March 2008

[19] Y. Park, I. Kim, S. Lee, "An Efficient Adaptive Feedback Cancellation for Hearing Aids," *Proceedings of the 25th Annual International Conference of the IEEE EMBS*, pp. 17-21, September 2003

[20] D. Chanzan, Y. Medan and U. Shvadron, "Noise Cancellation for Hearing Aids," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 36, no. 11, pp. 977-980, November 1988

[21] M.G. Siqueira and A. Alwan, "Bias analysis in continuous adaptation systems for hearing aids," *IEEE International Conference on Acoustics, Speech, and Signal Processing Proceedings (ICASSP '99),* vol. 2, pp. 925-928, 1999

[22] V. Hamacher, J. Chalupper, J. Eggers, E. Fischer, U. Kornagel, H. Puder, and U. Rass., "Signal processing in high-end hearing aids: state-of-the-art, challenges and future trends," *EURASIP J. Appl. Signal Process.* (18), pp. 2915-2929, 2005

[23] M.T. Maltby "Principles of Hearing Aid Audiology (Second Edition)," London, Wiley, 2004

[24] Phonak, "Claro AutoSelect: Sound classification for an intelligent automatic mulit-program management system," http://www.phonak.com, March 2008

[25] A. Ypma, B. de Vries & J. Geurts, "Robust Volume Control Personalisation from on-Line Preference Feedback," *IEEE Proc. Signal Processing Society Workshop on Machine Learning for Signal Processing*, Sept. 2006, pp. 441-446, 2006

[26] J.M. Kates, "On the feasibility of using neural nets to derive hearing-aid prescriptive procedures," *J. Acoust. Soc. Amer.*, vol. 98, No. 1, pp. 172-180, July 1995

[27] O. Arnsten, H. Koren, and T. Strom, "Hearing-aid pre-selection through a neural network," *Scand. Audiol.*, vol. 25, pp. 259-262, 1996

[28] C. Schweitzer, "Perhaps Not by Prescription – But by Perception," The Hearing Review, vol. 3, pp. 58-62, January 1999

[29] B. Prasad "Learning the users' preferences in e-commerce: A weight-adjustment approach," *International Journal of Knowledge-based and Intelligent Engineering Systems,* 8, pp. 205-211, 2004

[30] S.Y Jung, J. Hong, T. Kim, "A Statistical Model for User Preference," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no.6, pp. 834-843, June 2005

[31] D.A. Fabry, "DataLogging: A clinical tool for meeting individual patient needs," The Hearing Review, January 2005

[32] S. Munkstedt, *Subjectively preferred sound representation for different listening situations*, Master Thesis in Hearing Technology, March 2006

[33] S. Ekern, "Adaptive Exponential Smoothing Revisited," *Journal of the Operational Research Society*, vol. 32, pp. 775-782, 1981

[34] R. D. Snyder, "Progressive Tuning of Simple Exponential Smoothing Forecasts," *Journal of the Operational Research Society,* vol. 39, pp. 393-400, 1988

[35] M. Dobber, R. van der Mei and G. Koole, "A prediction method for job runtimes on shared processors: Survey, statistical analysis and new avenues," *An International Journal of Performance Evaluation*, vol. 64, pp. 755-781, 2007

[36] J. Bezarra, R. Braquehais, F. Roberto, J. Silva, M. Fernandez, T. de Araujo and C. Junior " NAES: A Natural Adaptive Exponential Smoothing for Channel Prediction in WLANs," *Wireless Pervasive Computing, 2007. ISWPC apos; 07. 2nd International Symposium on* vol. 5, pp. 476-480, February 2007

[37] B.L. Bowerman and R.T. O'Connell, *Time Series and Forecasting*, North Scituate, Massachusetts: Duxbury Press, 1987

[38] W.M. Chow, "Adaptive Control of the exponential smoothing constant," *Journal of Industrial Engineering*, 16, pp. 314-317, 1965

[39] S. N. Pantazopoulos and C. P. Pappis, "Theory and Methodology: A New adaptive method for extrapolative forecasting algorithms," *European Journal of Operational Research,* 94, pp. 106-111, 1996

[40] D. W. Trigg and A. G. Leach, "Exponential Smoothing with an Adaptive Response Rate," *Operational Research Quarterly*, vol. 18 no. 1, pp. 53-59 1967.

[41] C.W.J. Granger and T. Terasvirta, *Modelling non-linear economic relationships*, UK: Oxford University, 1993

[42] W.A. Barnett, A.P. Kirman and M. Salmon, *Nonlinear Dynamics and Economics: Proceedings of the Tenth International Symposium in Economic Theory and Econometrics (International Symposia in Economic Theory and Econometrics)*, New York, NY: Cambridge University Press 1996

[43] J.W. Taylor, "Smooth Transition Exponential Smoothing," *Journal of Forecasting*, vol. 23, pp.385-4004, 2004

[44] M. Recht, "The Decibel as a Musical Unit," *Music Educators Journal*, vol. 50, pp. 102-105, 1964

[45] L. Larmarche, *Adaptive Environmental Classification System for Hearing Aids*, Master of Applied Science Thesis, 2008

[46] S. Chalupper and H. Heuermann, "Learning Optimal Gain in Real World Environments," *International Hearing Aid Research Conference* 2006

[47] S. Kochin, "Marketrak VI: Isolating the Impact of the Volume Control on Customer Satisfaction," *The Hearing Review*, vol. 10, pp 26-35, 2003

[48] Internet source < *http://www.sfu.ca/sonic-studio/handbook/Graphics/Audiogram.gif*> (March 2008)

Appendix I

Relationship between τ and α

A 1st order LTI system with a constant (leakey integrator) is characterized by the equation

$$\frac{dV}{dt} = -\lambda V + k$$

Where λ is the exponential decay constant, V = V(t) and

$$\lambda = \frac{1}{\tau}$$

The general solution to the differential equation is:

$$V(t) = V_o(1 - e^{-t/\tau})$$

For an input which is a step response, $V(t) = V_{\max}(1 - e^{-t/\tau})$

When t = τ $V(t) = V_{\max}(1 - e^{-1})$ = $V(t) = 0.63V_{\max}$

In discrete time, the smoothing constant $\alpha = 1 - e^{-t/\tau}$

Algorithm 1.0: Adaptive Control of the Exponential Smoothing Constant

Input: Time series y = [y_1, y_2, y_3 ... ... y_n]

Data: Set value for d and Initialize values for α, $\hat{y}_t$

Output: Forecasting

For t = 1 to length(y)

Step 1: Compute the secondary smoothing constants

$$\alpha_u = \alpha + d$$

$$\alpha_L = \alpha - d$$

Step 2: Compute forecasts at time t+1, using the three constants

$$\hat{y}_{t+1} = \alpha y_t + (1-\alpha)\hat{y}_t$$

$$\hat{y}_{t+1,U} = \alpha_U y_t + (1-\alpha_u)\hat{y}_{t,U}$$

$$\hat{y}_{t+1,L} = \alpha_L y_t + (1-\alpha_L)\hat{y}_{t,L}$$

Step 3: Compute forecast errors resulting from all three constants:

$$e(t+1) = \hat{y}_{t+1} - y_{t+1}$$

$$e(t+1)_U = \hat{y}_{+1t,U} - y_{t+1}$$

$$e(t+1)_L = \hat{y}_{t+1,L} - y_{t+1}$$

Step 4: Find the minimum of e(t), $e(t)_U$ and $e(t)_L$
If e(t) is the lowest then leave α unchanged
ElseIf $e(t)_U$ is the lowest then

$$\alpha = \alpha_u$$
$$\alpha_u = \alpha + d$$
$$\alpha_L = \alpha - d$$

ElseIf $e(t)_L$ is the lowest then

$$\alpha = \alpha_L$$
$$\alpha_u = \alpha + d$$
$$\alpha_L = \alpha - d$$

End.

Algorithm 2.0: A new adaptive method for extrapolative forecasting algorithms

Input: Time series $y = [y_1, y_2, y_3 \ldots\ldots\ldots y_n]$
Output: Forecasting
For $t > 1$ and $t = 1$ to length(y)

Step 1: Calculate the forecast error from the previous period $e_t = \hat{y}_t - y_t$
Step 2: Calculate optimum smoothing constant for the previous period t

$$\alpha_t' = \left| \frac{(y_t - \hat{y}_{t-1})}{(y_{t-1} - \hat{y}_{t-1})} \right|$$

Step 3: If $e_t = 0$ then

$$\alpha_{t+1} = \alpha_t \text{ (Do nothing)}$$

ElseIf $\alpha_t' > 1$ then

$$\alpha_{t+1} = 1$$

Else $\alpha_{t+1} = \alpha_t'$

Step 4: Calculate $\hat{y}_{t+1} = \alpha_t y_t + (1 - \alpha_t)\hat{y}_t$

End

Algorithm 3.0: Exponential Smoothing with an Adaptive Response Rate

Input: Time series $y = [y_1, y_2, y_3 \ldots\ldots\ldots y_n]$

Data: Initialize α, ϕ *and* $\hat{y}_t$

Output: Forecasting

For t = 1 to length(y)

Step 1: Calculate estimate for next time period $\hat{y}_{t+1} = \alpha_t y_t + (1-\alpha_t)\hat{y}_t$

Step 2: Calculate the forecast error $e_t = \hat{y}_t - y_t$

Step 3: Calculate the smoothed forecast error $A_t = \phi e_t + (1-\phi)A_{t-1}$

Step 4: Calculate the smoothed absolute forecast error $M_t = \phi|e_t| + (1-\phi)M_{t-1}$

Step 5: Calculate the update something constant $\alpha_t = \left|\frac{A_t}{M_t}\right|$

End

Algorithm 4.0: Smooth Transition Exponential Smoothing

Input: Time series $y = [y_1, y_2, y_3 \ldots\ldots\ldots y_n]$

Data: Initialize β *and* γ

Output: Forecasting

For t = 1 to length(y)Step 1: Calculate forecast for the next period t+1

$\hat{y}_{t+1} = \alpha_t y_t + (1-\alpha_t)\hat{y}_t$

Step 2: Calculate forecast error based on estimate made for y_t *by* $\hat{y}_t$ $e_t = \hat{y}_t - y_t$

Step 3: Calculate transition variable $V_t = e_t^2$

Step 4: Calculate new value for smoothing constant $\alpha_t = \frac{1}{1+\exp(\beta+\gamma V_t)}$

End