PHYSIOLOGICAL ASSESSMENT OF HEARING AID COMPRESSION SCHEMES

PHYSIOLOGICAL ASSESSMENT OF HEARING AID COMPRESSION SCHEMES

By

Benedict K. H. Leung, B. Eng.

A Research Report
Submitted to the School of Graduate Studies
In Partial Fulfillment of the Requirements
For the Degree
Master of Engineering

McMaster University

Master of Engineering (2008)	McMaster University
(Electrical and Biomedical Engineering)	Hamilton, Ontario

Title:	Physiological Assessment of Hearing Aid Compression Schemes
Author:	Benedict K. H. Leung (McMaster University)
Supervisor:	Dr. Ian C. Bruce
Number of Pages:	vii, 56

Abstract

Nonlinear amplification schemes for hearing aids have been developed to deal primarily with the problem of loudness recruitment. The most commonly used form of nonlinear amplification is wide-dynamic-range compression (WDRC). Unfortunately, finding WDRC characteristics that satisfactorily deal with loudness recruitment while maintaining good speech intelligibility has proven difficult. An alternative nonlinear scheme, Advanced Dynamic Range Optimization (ADRO), has been shown in several studies to provide better speech intelligibility and listening comfort than fast-acting WDRC. ADRO uses a set of fuzzy-logic rules to make gain changes to optimize audibility, comfort, protection against loud sound, and noise attenuation. The "hearing protection" gain rule acts instantaneously, whereas the audibility and comfort rules adjust the gain slowly, such that ADRO provides linear amplification most of the time.

The goal of this study is to examine the physiological basis for the relative performance of linear amplification, WDRC, and ADRO. Sentences from the TIMIT Speech Database were processed by each algorithm. In the case of WDRC, both single-channel and multi-channel schemes with fast and slow dynamics were tested. Speech signals were presented at 52, 62, 74, and 82 dB SPL (sound pressure level) with various noise levels and types, to simulate real-life environments. The simulations first use an auditory-periphery model to generate a "neurogram" of the auditory nerve's representation of the test speech material. The spectral and temporal modulations in the neurogram are then analyzed by a model of cortical speech processing. The effects of the background noise, the presentation level, the hearing loss and the amplification scheme are evaluated by comparing the cortical model response for a given condition (the "test" response) to the cortical model response to the same TIMIT sentence presented in quiet at 65 dB SPL to the normal-hearing model (the "template" response). From the difference between the test and template responses, a spectro-temporal modulation index (STMI) value is calculated. High STMI values predict good speech intelligibility, while low values predict poor intelligibility. Results show that ADRO is better at restoring the neural representation of speech than the other algorithms tested, even when the WDRC algorithms utilize slow time constants. In the case of no background noise, all the algorithms perform similarly well. However, when background noise is added, STMI values for higher SPLs drop notably for all the algorithms except for ADRO, which sustains a stable value throughout the range of SPLs test.

Acknowledgement

The author thanks Dynamic Hearing for providing a software simulation of the ADRO algorithm, and the National Centre for Audiology at the University of Western Ontario for providing the DSL m[i/o] V5.0a prescription software.

The author also wants to thanks his supervisor, Dr. Ian C. Bruce for his invaluable guidance and support throughout the study period.

Contents

List of Figures

1 Introduction

Hearing loss due to sensorineural impairment is associated with deficits in aspects of hearing including audibility, dynamic range, frequency resolution, and temporal resolution. These deficits can to some extent be overcome by the use of a hearing aid, but no existing amplification scheme fully compensates for all aspects of hearing impairment. Consequently, it would be helpful to understand the physiological basis for different amplification schemes' ability and inability to counteract auditory deficits in different acoustic environments.

The decrease of audibility due to hearing loss is usually quantified via a pure-tone audiogram, in which the increases in the intensities (in dB) required to detect tones at different frequencies are given relative to the averages for normal hearing listeners. There are several different prescription formulae for determining appropriate amplification gain-frequency responses bases on the audiogram [Dillon 2001]. However, while audibility thresholds increase as a result of hearing impairment, the sound intensity at which sounds become uncomfortably loud typically does not change, leading to a reduction in the dynamic range of sound intensities for a hearing impaired listener. This has led to the introduction of amplitude compression schemes in hearing aids. Rationale for applying compression include avoidance of discomfort, distortion and damage; reduction of inter-syllabic intensity differences; long-term dynamic range reduction; comfort increase; loudness normalization; noise reduction; and increased intelligibility (For more information, refer to Dillon 2001). Unfortunately, a set of compression characteristics that achieves all these aims simultaneously has not been found. The aim of this study is to utilize computational models of speech processing in the ear and brain to better understand how compression algorithms affect the neural representation of speech.

2 Background

Hearing is one of the most important senses in our bodies and helps us perceive the world around us and to communicate using speech. Sound consists of pressure waves that travel through the air at about 340 m/s, and sound pressure is usually given in decibels (dB). The following equation is used to convert sound pressure into dB:

$$SPL = 20 \log_{10} (P_1 / P_0) \text{ (dB SPL)}, \tag{1}$$

where SPL = Sound Pressure Level, P_1 = sound pressure (Pascal), P_0 = reference sound pressure = 20 μPa.

2.1 Anatomy and Physiology of the Auditory Periphery

2.1.1 The Ear

The mammalian auditory system consists of three main parts, the outer ear, middle ear, and the inner ear. The real signal transduction starts in the beginning of the middle ear. The middle ear consists of two main parts: the tympanic membrane and the ossicular system. The tympanic membrane responds to the air pressure fluctuations generated by sound. The membrane is attached to the handle of the malleus – the first part of the ossicular system. The ossicles also include the incus and the stapes, which are attached together by small ligaments allowing the three bones to move together. The end of the stapes is the oval window of the cochlea; thus this middle ear system conducts the vibrational energy to the cochlea. The basilar membrane (BM) and the organ of Corti lie within the cochlea. Inside the organ of Corti, there are the hair cells and the tectorial membrane. Figure 2-1 shows a diagram of the middle and inner ear.

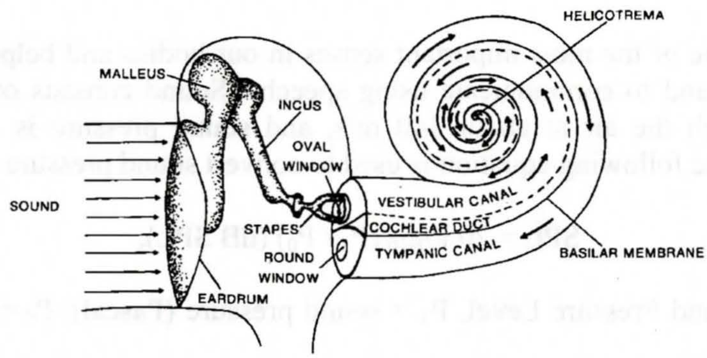


Figure 2-1: The middle and inner ear of the whole hearing system from the eardrum to the cochlea. The arrows indicate signal flow. The middle ear acts as an air to liquid signal transduction system. [Warren 1999]

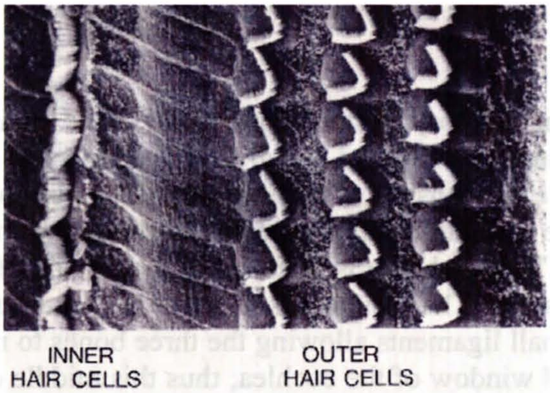
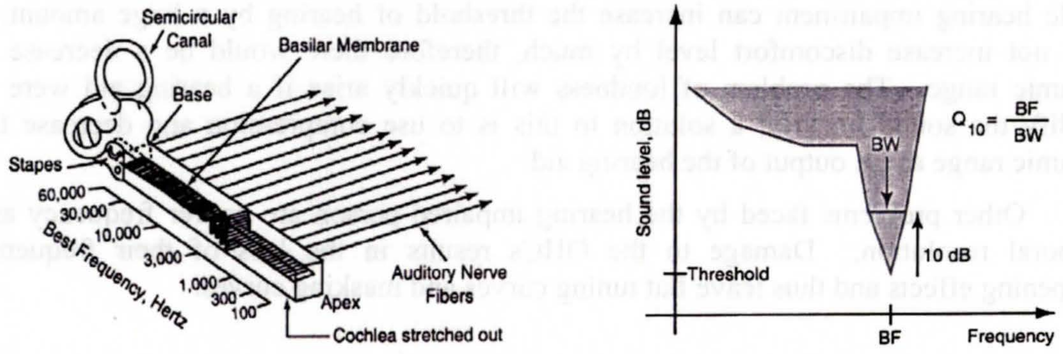


Figure 2-2: Hair cell arrangement in the organ of Corti along the length of the basilar membrane. [Warren 1999]

Pressure waves will cause the BM to vibrate and subsequently the stereocilia of the hair cells (Figure 2-2) in of the organ of Corti to shear against the tectorial membrane. This will depolarize the hair cells and may generate impulses in the auditory nerve. Most of the afferent nerve fibers that send information to the auditory centers come from the inner hair cells (IHC); they line up in a single file down the BM inside the organ of Corti. In contrast, the outer hair cells (OHC) only have very few afferent fibers, but they have efferent fibers and thus receive control from the brain. They are arranged three-in-a-row all the way along the BM. OHCs act as the feedback control for the IHC, amplifying and sharpening the IHC's response. This phenomenon is called the cochlear amplifier [Robles 2001].

2.1.2 Tonotopic Representation

If the cochlear turns are stretched out, we can see that the responses of the BM is similar to a Fourier representation of sound. The BM is frequency tuned and the scale is Best Frequency (BF) or Characteristic Frequency (CF) (Figure 2-3).



A) B)

Figure 2-3: A: tonotopic representation of the stretched out basilar membrane. B: The tuning curve [Sachs 2002]

For a single place on the BM, a curve can be drawn to show the sensitivity in terms of frequency, the tuning curve. Any input above the curve will cause an increased firing rate of the neuron. This curve is usually characterized by BF, Threshold and Q_{10dB}. The Q_{10} value is a scale for sharpness of tuning, the higher the better; the higher the BF, the higher the Q_{10} value [Sachs 2002].

2.1.3 Hearing Loss

There are two main types of hearing impairment: conductive hearing loss and sensorineural hearing loss. Conductive hearing loss is the decrease in effectiveness of sound conduction through the ear canal, tympanic membrane, and ossicles. Sensorineural hearing loss is the more common form of hearing impairment and any damage of auditory neural units like the hair cells in the inner ear will cause such impairment [Dillon 2001]. This study only discusses the latter form.

The first and most obvious symptom of hearing loss is the decrease in audibility. For people with mild or moderate loss, they can hear some sounds more easily than others. For example, softer phonemes like some consonants will not be heard, whereas vowels are more easily heard. Even if some words are audible, they may be hard to understand by the hearing impaired, because higher formants (intense region by frequency) in the upper frequency regions will unlikely be heard. Higher frequency components of vowels are weaker than lower frequency components. This is worsened by the fact that sensorineural hearing loss often has a greater impairment at higher frequencies [Goldstein 2002]. As a result, vowels with the same first formants and different second formants would sound nearly identical for the hearing impaired person, e.g. /oo/ and /ee/ [Dillon 2001].

Another problem faced by the hearing impaired person is the decreased dynamic range, which is the level difference between discomfort and the threshold of audibility. While hearing impairment can increase the threshold of hearing by a large amount, it does not increase discomfort level by much, therefore there would be a decrease in dynamic range. The problem of loudness will quickly arise if a hearing aid were to amplify the sound linearly; a solution to this is to use compression and decrease the dynamic range at the output of the hearing aid.

Other problems faced by the hearing impaired person are poorer frequency and temporal resolution. Damage to the OHCs results in the loss of their frequency sharpening effects and thus leave flat tuning curves and masking curves.

2.2 Hearing Aid Algorithms

2.2.1 Compression

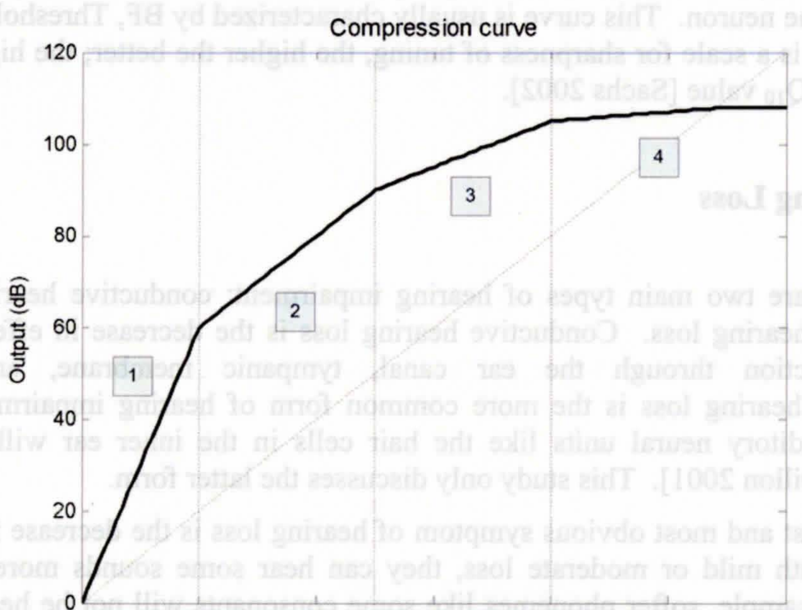


Figure 2-4: A sample WDRC compression curve with 1) expander 2) linear amplification 3) compression 4) limiting

As mentioned before, a hearing impaired person will have their hearing dynamic range decreased. Therefore, a compressed range of input SPLs will allow them to hear the range of input without being too soft or too loud. One of the most common ways to compress the range is to use a Wide Dynamic Range Compressor (WDRC). This compressor gradually decreases gain as the input level increases.

A compressor is inherently a dynamic device; it detects the incoming signal level and changes gain accordingly. The rate of this change determines the speed at which the compressor reacts. There are two time constants for each compressor: the attack time and the release time. Attack time is the time constant for the compressor to react to an increase in signal level. Release time is the time constant for the compressor to react to a decrease in signal level. Presently, there is still no algorithm or scheme that prescribes attack or release times. There are many options to choose from with different rationales, and this study used two of them – a fast pair of time constants for phonemic compression, and a slow pair of time constants for longer term dynamic range reduction.

Figure 2-4 shows a sample WDRC compression curve with its four different sections. The first section is the expander (also called noise gater), used in place of linear gain to reduce noise caused by amplifying very quite sounds by a set amount of decibels. Expanders act on inputs lower than ~20 dB SPL because they are used only to reduce unwanted amplification and noise. The second section is the linear gain region, minimizing changes in sound quality. Linear gain is applied up to the compression threshold of about 50 dB. (This level is variable and it depends on the prescription and algorithm.) Section three is the compression region. The range is from the compression threshold to the start of the next region, which is limiting. The compression ratio of this region varies between different prescriptions, but the range would be around 1.5:1 to 3:1. The last section is the compression limiting region, and this exists to prevent the output from being too large, such that it would further damage hearing or distort the loudspeaker (or *receiver*) of the hearing aid. In limiting, the compression ratio increases greatly (e.g., 10:1) to reduce a loud signal's dynamic range. If the input level reaches even higher, the compressor may even have a flat compression curve (infinite compression ratio). Hearing prescriptions may even skip the high compression ratio limit and use only the infinite compression limit.

2.2.2 Real Ear Gains

Hearing aid prescription formulas set the desired amplification values in terms of either Real-Ear Aided Gain (REAG) or Real-Ear Insertion Gain (REIG or just IG). REAG is the actual level difference in dB SPL between the area just beside the ear drum and a reference point outside the auricle (or free field). A hearing aid will need to amplify this amount to achieve the desired gain from the prescription. Included in this gain is the Real-Ear Unaided Gain (REUG) or the natural frequency specific gain applied to the signal by the ear canal. The act of inserting the hearing aid into the ear will negate the natural amplification; therefore, this natural gain was added to the REAG. REIG is the difference between REAG and REUG.

$$\text{REAG} - \text{REUG} = \text{REIG} \qquad (2)$$

Another common gain measure used by hearing aid manufactures and fitters are gain for 2cc-coupler units in dB SPL. It is convenient to test the hearing aid hardware using a 2 cm cubed testing box; therefore the 2cc-coupler units were used. To convert 2cc-coupler unit to REAG, Real Ear to Coupler Difference (RECD) values would be added to the 2cc-coupler gains.

2.2.3 Linear Aids

2.2.3.1 NAL-RP

One of the popular linear prescriptions used today is NAL-RP and it has been revised a number of times since it was first published in 1976. NAL stands for National Acoustic Laboratories of Australia, R stands for revised, and P stands for profound. The aim of the procedure is to maximize speech intelligibility at the listening level preferred by the aid wearer, and it was assumed that intelligibility was maximized when all frequencies of speech are perceived to have the same loudness, referred to as *loudness equalization*. The advantage of having loudness equalization is the user does not have to vary the volume just because a particular frequency region gets too loud or too soft. In other words, loudness equalization will maximize audibility in a user's comfortable hearing range.

The original NAL formula was derived empirically. First, the preferred insertion gain at 1 kHz was set to 0.46 times the hearing loss in dB at 1 kHz (a minor variation from the half gain rule); then to get the preferred gain at other frequencies, the gains were adjusted to follow the long-term average speech spectrum; finally, a 60-phon equal loudness curve (equal loudness perceived relative to 60 dB SPL at 1 kHz) for normal hearing was used as a reference. Putting the above steps together generates the original NAL formula. The subsequent revisions were to compensate for steeply sloping hearing loss, and profound hearing loss. The NAL-RP's prescription is in insertion gain, and its formulae are shown in Figure 2-5. A sample of its prescription along with DSL is shown in Figure 2-6.

$$H_{3FA} = (H_{500} + H_{1k} + H_{2k})/3$$

$$X = 0.15 * H_{3FA} \qquad \text{For } H_{3FA} < 60$$

$$X = 0.15 * H_{3FA} + 0.2 * (H_{3FA} - 60) \qquad \text{For } H_{3FA} > 60$$

$$IG_i = X + 0.31 * H_i + k_i + PC$$

Freq (Hz)	250	500	1k	2k	3k	4k	6k
k_i (dB)	-17	-8	1	-1	-2	-2	-2

(PC)	Frequency (Hz)						
H_{2kHz}	250	500	1k	2k	3k	4k	6k
≤ 90	0	0	0	0	0	0	0
95	4	3	0	-2	-2	-2	-2
100	6	4	0	-3	-3	-3	-3
105	8	5	0	-5	-5	-5	-5
110	11	7	0	-6	-6	-6	-6
115	13	8	0	-8	-8	-8	-8
120	15	9	0	-9	-9	-9	-9

Figure 2-5: NAL-RP formulae [Dillon 2001].

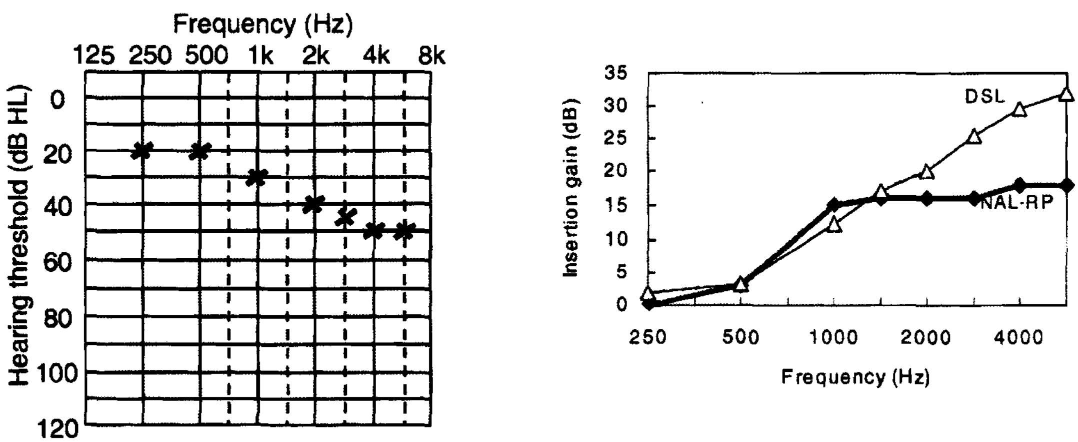


Figure 2-6: Sample hearing loss profile (left), and its prescription in DSL and NAL-RP. Prescription for DSL was set to REIG for comparison [Dillon 2001].

2.2.3.2 DSL

DSL stands for Desired Sensation Level, and its aim is to provide the aid user with an audible and comfortable signal in each frequency region. The DSL prescription uses REAG instead of REIG, and instead of attempting to make speech equally loud in each frequency, DSL attempts to make it comfortably loud. DSL targets are given in terms of sound levels at the ear drum. Dillon [2001] provided equivalent REAGs as a lookup table (see Table 2-7).

In a similar manner to NAL-RP, DSL was derived empirically. The desired sensation levels were found experimentally for profound losses. For mild to severe losses, the desired sensation levels were set below the estimated most comfortable level by one standard deviation. For normal hearing, the desired sensation levels were set to be the same as the level experienced by people with normal hearing. Using the desired sensation level sets, hearing threshold, and short term maximum speech levels in the field for speech at 90 dB SPL, REAG can be calculated.

Table 2-7: DSL lookup table of REAG [Dillon 2001].

	Frequency								
dB HL	**250**	**500**	**750**	**1000**	**1500**	**2000**	**3000**	**4000**	**6000**
0	0	2	3	3	5	12	16	14	8
5	3	4	5	5	8	15	18	17	11
10	5	6	7	8	10	17	20	19	14
15	7	8	10	10	13	19	23	21	17
20	9	11	12	13	15	22	25	24	20
25	12	13	14	15	18	24	28	27	23
30	14	15	17	18	20	27	30	29	26
35	17	18	19	21	23	30	33	32	29
40	20	20	22	24	26	33	36	35	32
45	22	23	25	27	29	36	39	38	36
50	25	26	28	30	32	39	42	41	39
55	29	29	31	33	35	42	45	45	43
60	32	32	34	36	38	46	48	48	46
65	36	35	37	40	42	49	52	51	50
70	39	38	40	43	45	52	55	55	54
75	43	42	43	46	48	56	59	58	58
80	47	45	47	50	52	59	62	62	61
85	51	48	50	53	55	63	66	65	65
90	55	52	54	57	59	66	69	69	69
95	59	55	57	60	62	70	73	73	
100	62	59	61	64	66	73	76	76	
105		62	64	68	70	77	80	80	
110		66	68	71	73	80	83	84	

2.2.4 Non-linear Aids

Non-linear aids differ from linear aids by having multiple gain-frequency responses for different input levels; the prescription can also be viewed as specifying individual input-output curves for each frequency band. The reason for having two separate representations of the prescription is because the hearing aid gain changes relative to frequency and input level. It is convenient to observe the compression curves from a set of input-output curves, and filter characteristics are more easily read from a set of gain-frequency responses.

2.2.4.1 NAL-NL1

NL1 is NAL's non-linear version 1 and its underlying rationale changed from attempting to restore loudness in each frequency to maximizing speech intelligibility, and it attempts to give the aid wearer the loudness perception of speech similar to a normal hearing person. To achieve this, NAL-NL1 uses two theoretical models. The first model uses a modification of the Speech Intelligibility Index to incorporate hearing loss effects and hearing in high SPL fields. The second model uses a method for calculating loudness, and it was adjusted so that it will take into account the effects of sensorineural hearing loss. Both of the models require only the hearing threshold and the speech spectrum at the output of the hearing aid.

For each input level and frequency, the gain was systematically varied until the calculated speech intelligibility was maximized. However, the constraint was that the loudness would not exceed the perceived loudness for a normal hearing person. After many simulations, an equation was fitted to the complete set of optimized gains. The equation consolidates all the optimizations and can be applied to any audiogram, and it is integrated in a computer program called NAL-nonlinear [Dillon 2001]. Software version 1.40 with formula version 1.20 was used in this study. Although NL1 uses a separate goal and rationale in calculating its gain targets, the resultant gain-frequency curves for input levels around 65-70 dB SPL were very close to the original NAL-RP formula. Therefore, loudness normalization (NAL-RP's main goal) was a consequence of maximizing intelligibility. It was obvious at National Research Laboratory that the NAL-RP gain-frequency curve was very close to an optimum prescription to increase intelligibility, therefore, half of the difference between NAL-NL1's and NAL-RP's gain was added to NL1 to get the final targets [Byrne 2001].

2.2.4.2 DSL m[i/o]

DSL m[i/o] stands for Desired Sensation Level Multistage Input Output Algorithm. It received a major revision recently from DSL [i/o] v4.1 to DSL m[i/o] v5.0 [Scollie et al., 2005], and another minor one (v5.0a) to update WDRC targets for severe to profound hearing loss [Child Amplification Laboratory 2007]. The goals for DSL v5.0 as listed in Scollie and colleagues' [2005] description of the new algorithm are as follows:

- Avoidance of loudness discomfort during hearing instrument use
- Hearing instrument prescription that ensures audibility of important acoustic cues in conversational speech as much as possible
- Support for hearing instrument fitting in early hearing detection and intervention programs
- Prescription of hearing instrument compression that is appropriate for the degree and configuration of the hearing loss, but that attempts to make a wide range of speech inputs available to the listener
- Adaptation for the different listening needs of listeners with congenital versus acquired hearing loss
- Accommodation for the different listening requirements within quiet and noisy listening environments

DSL m[i/o] has four stages of processing: expansion, linear gain, compression, and output limiting; hence the name multistage. In the expansion stage, DSL m[i/o] sets an expansion threshold (ET) to be about 10 dB lower than soft speech. It is likely sound below ET would be background noise. The linear stage spans in the input range between the ET and the compression threshold (CT). The compression or WDRC stage then calculates its gain at every one-third octave frequency for a 60 dB SPL input. A slope equal to a prescribed compression ratio is plotted to intersect both the CT and the desired output at 60 dB SPL point for each one-third octave frequency. The span of the compression stage is from whichever is lower of 30 db SPL or the CT up to the Broadband Output Limiting Threshold (BOLT). BOLT is the maximum output level for each one-third octave frequency band, and it is one of the factors DSL m[i/o] use in the limiting stage. The other two are the Upper Limits of Comfort (ULC), and OSPL90 (Output Sound Pressure Level Limiting when the input is 90 dB SPL).

2.2.4.3 ADRO

A relatively new algorithm designed specifically for digital hearing aids is Adaptive Dynamic Range Optimization (ADRO™), which was developed by The University of Melbourne and the company Dynamic Hearing based in Australia. It is a digital multi-channel FFT-based amplification strategy that uses statistical analysis of the output signal of each individual frequency channel. A set of fuzzy logic rules control the gain of the output.

ADRO's main rationale is to improve audibility, comfort and intelligibility to the user [Blamey 2005, Blamey 2006]. ADRO attempts to optimize the dynamic range of the amplified sound so that the hearing aid conveys the most information at a sound pressure level most suitable to the listener. Therefore, the sound needs to be audible and comfortable to the listener – neither too soft nor too loud. A problem arises for prescribing ranges of output to the listeners because the notions of 'too soft' or 'too loud' are subjective and prescriptions needs to be tailor-made for each individual. Another aspect of sound that ADRO addresses is sound quality, since ADRO acts as a linear aid over the short term, as long as the output levels do not go outside audible or comfortable levels. Consequently ADRO gives good sound fidelity similar to linear aids.

Being linear most of the time is one of the main differences between ADRO and other nonlinear amplifiers. The algorithm analyzes the incoming sounds and adjusts the gain so that information rich signals will be passed to the user, and independent gain adjustments are made for each frequency channel (up to 64). For a 32-channel ADRO aid, the channels are equally spaced with a bandwidth of 250 Hz spanning from 250 to 8000 Hz. Figure 2-8 shows the sound processing stages of ADRO. The initial reason for having so many channels was that it is possible to implement it with an available digital signal processor that is highly optimized for calculating discrete Fourier transforms. It was later found that it generated some interesting and worthwhile benefits, although one of the drawbacks was a relatively long group delay for a high number of channels [Blamey 2006]. Dynamic Hearing has subsequently reduced the group delay below 2ms by replacing the FFT with an adaptive FIR filter with many taps [Dickson and Steele, 2006]. The FFT-based version of ADRO was used in the study reported here. One of the advantages of having so many channels is that it increases the flexibility to shape the maximum gain, maximum output levels, comfort targets, and audibility targets at many frequencies to fit well to different audiograms. Another advantage is it has inherent ability to suppress narrow band noise; for example if low-intensity noise is filling up a channel, ADRO will reduce the gain of that particular channel reducing the noise.

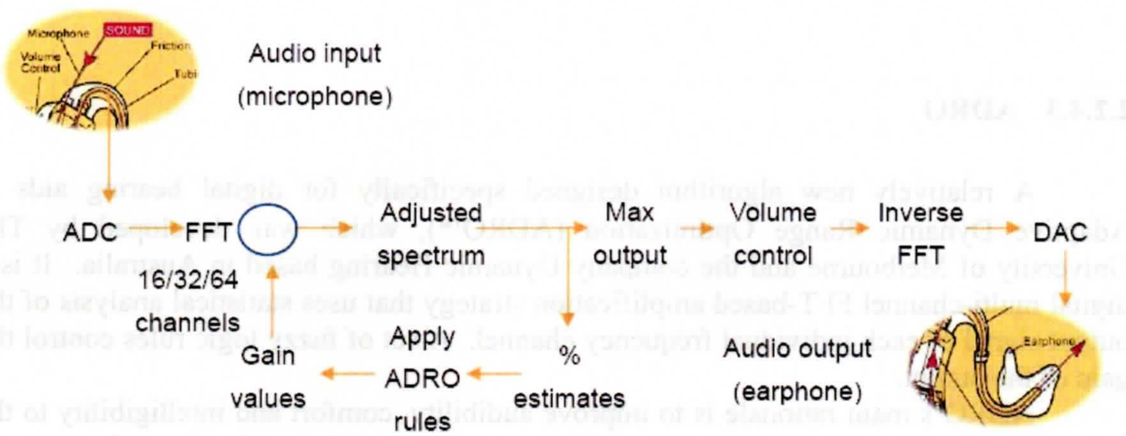


Figure 2-8: Sound processing stages for a typical FFT-based implementation of ADRO in a hearing aid. ADC is analog to digital converter; DAC is digital to analog converter; FFT is fast Fourier transform [Taken from Blamey 2006].

Instead of using a fixed compression function to control the gain, ADRO uses a set of 4 rules to reach its output goals. Statistical analysis of output sound intensity is done in each channel [Blamey 2005, Blamey 2006]:

1. The "comfort rule" ensures that sustained sounds are not uncomfortably loud, and the summed output level of all the channels does not exceed the user's comfort level. It reduces gain in a channel if the output level exceeds the "comfort target" (comfortable level) more than 10% of the time.
2. The "audibility rule" ensures that sustained sounds are not too soft. It increases the gain in a channel if the output level is below the "audibility target" (an acoustic level about 20 dB below the comfort level) more than 30% of the time.
3. The "hearing protection rule" prevents damage and uncomfortably loud sounds, a fast-acting maximum output level limiter for each channel.
4. The "background noise rule" prevents low-level background noise from high-level amplification. It limits the maximum gain in each channel.

In summary, these 4 rules keep the output within a comfortable, audible zone for the user in each frequency channel. It uses fuzzy logic in which it increases gain if too soft, reduces gain if too loud; if the output is within the comfort/audible or optimized zone, it does nothing and acts as a linear amplifier [Blamey 2005, Blamey 2006].

These rules will change the gain in a fashion similar to that of a relatively slow-acting compressor. The rationale behind a slow-acting compressor is to decrease the longer-term dynamic range without changing the intensity relationships between syllables spaced closely together in time. (In contrast, fast-acting compressors decrease inter-syllabic level differences.) There is a compromise between fast and slow time constants; fast time constants lead to reduced sound quality because in reducing inter-syllabic difference, the compressor adds distortions to sound; on the other hand, slow time constants can lead to loudness discomfort for sudden loud sounds. ADRO avoids the compromise between fast and slow time constants by using the 4 rules described above;

in other words, it has no single set of attack and release times. The hearing protection rule or the maximum output level rule operates instantaneously to limit sudden loud sounds, and operates in individual channels. The audibility rule and the comfort rule operate more slowly to maintain the sound within the optimum range of hearing in each frequency channel. The rate of change of these two gain rules operate at 3 to 6 dB/s, and from Dynamic Hearing's experience, more people prefer a slower adaptation rate [Blamey 2006].

The four rules of ADRO processing effectively optimize the output sound levels for a listener and increase the intelligibility of speech. Since ADRO acts linearly most of the time, the sound quality is good as well. However, since ADRO does not converge to a single I/O curve after coming back from the audibility or comfort range, a single reference loudness may be perceived differently because the I/O function is not static. By looking at a sample I/O function measured by giving ADRO a test pure-tone input (Figure 2-9), one can see that a single input level can be perceived to have different loudness at different times or different input levels can be perceived as being the same loudness at different times. This characteristic of ADRO does not affect speech intelligibility but it can change sound loudness perception. One algorithm that responds very similarly to ADRO is an automatic volume control, but ADRO is more sophisticated with multi-channel fuzzy logic processing.

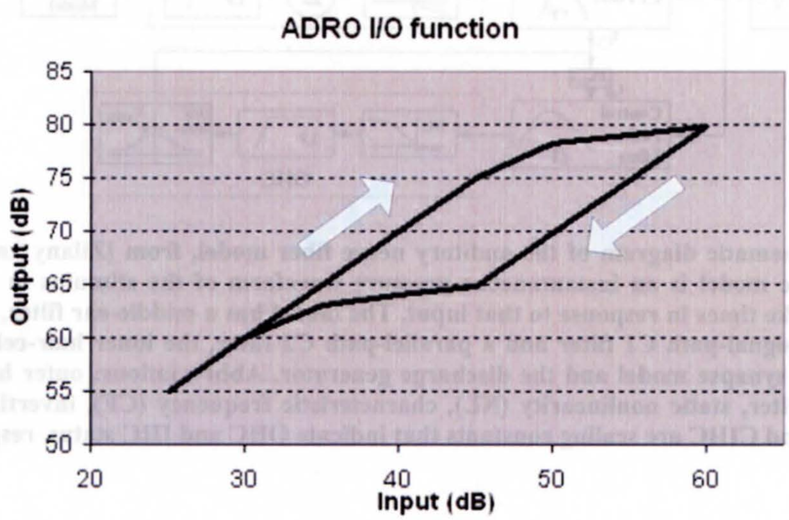


Figure 2-9: The input/output function for ADRO measured with a pure-tone in a hearing aid test box is a closed loop [Blamey 2005].

2.3 Model of the Auditory-periphery

The auditory-periphery model used in this study was recently developed by Zilany and Bruce [2006] (Figure 2-10). This model can accurately represent realistic responses in auditory nerve (AN) fibers in cats. It is a phenomenological model of the auditory-periphery that can simulate a wide range of characteristic frequencies (CFs) and intensities spanning the dynamic range of hearing. It includes a description of the main functional components from the middle ear to the auditory nerve.

The sound inputs are to be converted into units of Pa (Pascal) and are sampled at 500 kHz, and the output of the model is a two dimensional time-frequency plot called a "neurogram"; the pseudo colours represent simultaneous outputs (discharge rates averaged over 8 ms) from 128 AN fibers with CFs spaced logarithmically from 0.18 to 7.04 kHz. The model can incorporate OHC and IHC impairment to produce a range of hearing loss profiles. For this study, the hearing threshold shift at each CF, 2/3 was created by OHC and 1/3 by IHC impairment.

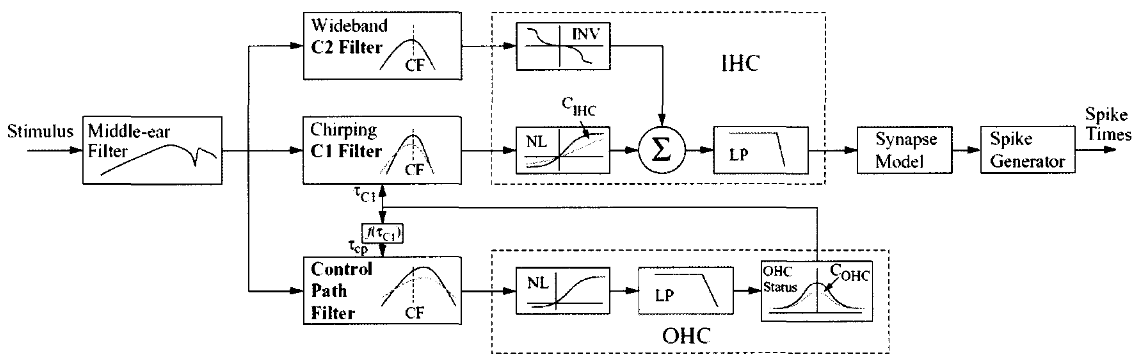


Figure 2-10: Schematic diagram of the auditory nerve fiber model, from [Zilany and Bruce 2006]. The input to the model is an instantaneous pressure waveform of the stimulus in Pascal and the output is the spike times in response to that input. The model has a middle-ear filter, a feed-forward control-path, a signal-path C1 filter and a parallel-path C2 filter, the inner hair-cell (IHC) section followed by the synapse model and the discharge generator. Abbreviations: outer hair cell (OHC), low-pass (LP) filter, static nonlinearity (NL), characteristic frequency (CF), inverting nonlinearity (INV). COHC and CIHC are scaling constants that indicate OHC and IHC status, respectively.

2.4 Model of Speech Processing in the Central Auditory System

The auditory-periphery output is analyzed by this stage to estimate the spectral and temporal modulation content of the AN neurogram. The primary auditory cortex receives and analyzes the dynamic acoustic spectrum of the input by using arrays of spectro-temporal response fields (STRFs) [Elhilali et al. 2003]. The STRFs act as modulation-selective filters of the input neurogram and summarize the way cells respond to a stimulus. In this study, we used a bank of modulation-selective filters that ranges from 2 – 32 Hz temporally, and 0.25 to 8 cycles per octave scales spectrally [Zilany and Bruce 2007].

2.5 Spectro-Temporal Modulation Index (STMI)

The spectro-temporal modulation index (STMI) was proposed by Elhilali et al. [2003] as a metric for predicting speech intelligibility based on the output of the cortical modulation filter bank. To calculate the STMI, a cortical response is compared to a template (expected) cortical response, as illustrated in Figure 2-11. The closer a cortical response is to the template, the higher the value of the STMI; the more the cortical response deviates from what is expected, the lower the STMI. In this study, the template is the unprocessed sentence presented at 65 dB SPL (conversational speech level) in quiet to the normal model of the auditory periphery [Zilany and Bruce 2007].

To compute the STMI, the template and the simulation neurogram (2-D: time and frequency) are passed through the filter banks and the output is 4-dimensional (4-D: time, frequency, rate, scale) complex-valued representation. The template T and test N cortical responses will then be passed through equation (2) to compute the STMI – a modified version of Elhilali and colleagues' equation (1) [Zilany and Bruce 2007]:

$$\mathrm{STMI} = \sqrt{1 - \frac{\|T - N\|^2}{\|T\|^2}}, \tag{2}$$

where $\|\cdot\|$ signifies the 2-norm.

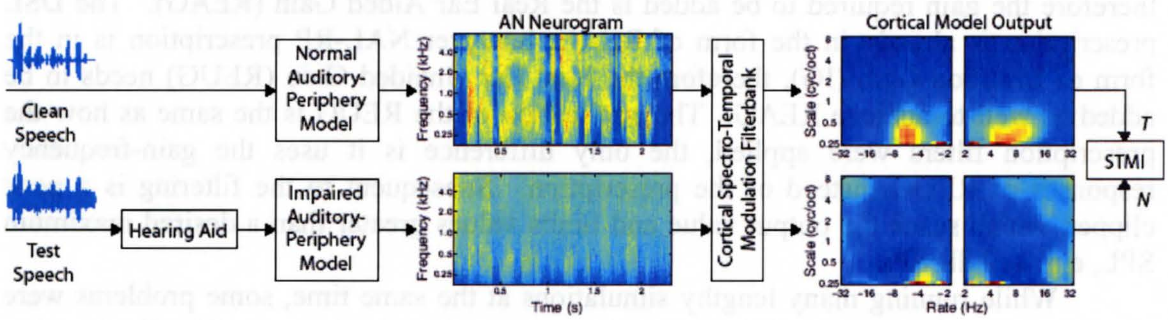


Figure 2-11: Schematic of the STMI computation. The clean and test speech signals are given as inputs to the auditory model. The right panel shows the cortical output of both clean and test inputs. These cortical patterns are then used to compute the template-based STMI.

3 Hearing Aids Simulation Design

3.1 Linear Aids

As mentioned before, DSL and NAL-RP prescriptions are used in simulating linear aids. Since linear aids will not have compression and response times like attack and release times, implementation of these aids is relatively straightforward. Amplification is done by passing the sound signal through a filter with a gain frequency response matching the gain-frequency curve of the prescription. The prescriptions are calculated for a given audiogram using the NAL-RP formula or the DSL lookup table from Dillon [2001]. Two different filter implementations were utilized; the first filter was FIR filter based and the second was FFT based.

At the early stages in the project, it was decided that the simulations would not be run in real time; each sound track can be filtered by passing the full length of data into a MATLAB function. The convenience of using the built-in MATLAB filter function made the implementation easier.

MATLAB has a function called **fir2.m**. It is a frequency sampling-based finite impulse response filter design function for arbitrary shaped frequency responses. By giving the function the frequency points, the desired gain at those frequencies, and the filter order, the function will generate coefficients for a filter that has the shape close to that of the desired responds. The coefficients can then be passed to another MATLAB function called **filter.m** to implement the filtering.

The final output of the filter requires being in pressure levels at the ear drum, therefore the gain required to be added is the Real Ear Aided Gain (REAG). The DSL prescription is already in the form of REAG, however NAL-RP prescription is in the form of Insertion Gain (IG), therefore the Real Ear Unaided Gain (REUG) needs to be added as well to achieve REAG. The application of the REUG is the same as how the prescription filters were applied, the only difference is it uses the gain-frequency responses of REUG instead of the prescription. Subsequent to the filtering is a peak clipper, which scans the output value and limits values greater than a desired maximum SPL, e.g. 110 dB SPL.

While running many lengthy simulations at the same time, some problems were encountered which prevented large numbers of parallel simulations. The Engineering department has a limited number of licenses for each special MATLAB tool box. The department only has 50 licenses for the Signal Processing Tool Box in which the filter function resides. This is one of the reasons to change to the FFT implementation. Another reason to change to the FFT implementation was that the filter function actually introduces some delay (filter order / 2 samples of delay). Although this can be quickly solved with the **filtfilt.m** function in MATLAB, it still does not address the licensing problem. Also, the gain needs to be reduced to its root value because **filtfilt.m** applies the gain 2 times, backwards and forwards to achieve zero phase shift. Another incentive to switch to an FFT implementation is that the technique can be carried over to the non-linear algorithms.

Although the simulations do not need to run at real time, having an STFT implementation is more analogous to a real life digital hearing aid design. The FFT needs a set of sample points to calculate the transform so the sound signal needs to be sectioned into short windows of data; this task is done by the function **stratify.m**. The length of the FFT window was not crucial at this point of design because the frequency response is time invariant in the linear hearing aid design, therefore a lengthier time window was acceptable. The length of the window was set at 16 milliseconds with an overlap of 50% and Hanning window was used to rid the transform of oscillations providing a better representation of the signal. The rationales for using the Hanning window includes: Hanning window tapers the ends of the window to zero, the 50% overlapping point between adjacent windows are exactly at half amplitude, and it is relatively simple to implement. This window setup made sure that the amplitude of the signal was minimally distorted. Only the ends of the signal were slightly distorted because there are no overlaps of windows, therefore half window length of the beginning and end of the signal will have their amplitude modulated by Hanning window. This variation is not significant because all of the simulations did not have speech information in those areas, and the total duration of change is one window length (16 ms in this case).

After the sound signal was sectioned, each part was zero-padded to 32 milliseconds giving more points to shape the spectrum and the transfer function of the gain filter. The increased points allow the shape of the transfer function to match the prescribed gain-frequency curve more closely. The filter transfer function was interpolated to correspond with the frequency axis of the FFT coefficients, and then the FFT of the signal and the interpolated gains were multiplied together, effectively applying the gain needed to the FFT section. The inverse transform was done on the product to synthesize the time domain signal, however the extra zero padded length still needs to be truncated and return it to the original window length. Each piece of the synthesized data was then placed one after another with 50% overlap generating an uninterrupted sound data. This job was done by the function named **overlapandadd.m**.

3.2 Non-linear Aids

Early on in the project, the lab did not yet have the prescription calculating software packages for DSL m[i/o] and NAL-NL1, therefore the prescriptions from DSL and NAL-RP were used as a starting point for generating a non-linear aid design. The linear algorithm only provides one gain-frequency curve for a given hearing loss profile, so initial implementation used only that single gain-frequency curve. A problem with using the linear algorithm was that no compression curve was given and the input-output function is a straight line because it is linear, therefore the compression curve was arbitrarily set to match wide dynamic range compression (WDRC) aid characteristics.

In this simplified WDRC scheme, the single channel algorithm first compresses the input and then amplifies the signal with a filter similar to that of the corresponding linear aid. The compression curve is simply a straight line starting from the origin with a slope of 1 until the compression threshold of about 50 dB, the slope then drops to a flatter slope, for example ½, setting the compression ratio to 2:1. The simplified multi-channel scheme is simply the multi-channel version of the single channel with the addition of a filter bank that separate the input sound into an arbitrary number of channels and the single channel algorithm is used in each channel.

The filter bank **fbank.m** used in the design was custom made so that it could integrate better with the overall design and avoid the use of MATLAB's Signal processing toolbox. The filter bank is FFT based and it is a collection of high pass, low pass, and band pass filters, all shaped by the Hanning window. The digital filters were also a custom made function **fftfilter.m**. Initially, the filter bank was designed with the FIR filter in the signal processing toolbox, however it added problems of the FIR filter design in MATLAB and it did not give high enough separation for lower frequency channels when channel number increased to about 7 (channel distances are separated by octaves).

Given that non-linear hearing aids require non-linear amplification schemes, DSL m[i/o] and NAL-NL1 were included in the final project design. This change of algorithm demanded an overall change of both the single-channel and multi channel-compressors, because amplification is now dependent on both frequency and input power. While the linear algorithm has only one gain-frequency curve for all input levels, the non-linear algorithm has different curves for different input levels. The same is true for input-output curves as well; different frequency bands have different input-output curves. This complicates implementation and different algorithms are used to tackle the problem. Another problem is that the prescription from the fitting software does not give formulas or a lookup table, it only gives the final prescription for the particular patient information that it was given. Both DSL m[i/o] and NAL-NL1 provide enough information to generate multiple gain frequency curves and/or input-output curves. Also, both fitting software packages were able to give REAG prescriptions so no gain adjustments for REUG were needed.

While the fitting software does provide shape of the curves, maximum ranges, and limits, it does not provide the shape for low level inputs. A straight line can be extrapolated to get those values, but expanders were added to the low input level region of the compression curves instead of just extrapolating the values. Expansion is also known as noise-gating and it is used to reduce audibility of very low level sounds and noise generated by the amplifier trying to amplify those low level sounds [Dillon 2001]. The range expansion begins from the beginnings of detectable sound pressure levels (SPL) to around 25 dB SPL. This point was chosen because the main purpose of this section is to suppress noise generate by the linear gain for low level input. Minimal expansion is wanted in the more audible levels of sound to maintain sound quality.

3.2.1 Single Channel Compression

Many methods were attempted to realize the single channel compressor with the gain dependent on both frequency and input power. One of the first attempts was to trace frequency and input power to adjust the gain, but this idea was dropped because real hearing aids do not work this way; also it loses frequency specific gain. To get gain profiles to match the prescriptions in a single channel, filters similar to what the linear algorithm used were needed. The difference would be that there are different filters for different input levels, and to get a specific input level's curve, it was interpolated from the series of curves provided by the prescription.

The compressor was changed to a filter based design derived from the linear FFT algorithm. Sectioning for the STFT was the same in this design with the sound power calculator added. The average power value in dB was calculated for each window and used to select a specific filter for it. Since gain was changed for every window, the window width needed to be much shorter than before to allow gain changes between windows – it was set to be 2 ms.

For the compressor's response (attack and release) times, early in the project it was post-processed or computed after all the amplification filters were applied. This is easier to apply but it introduces some error because it scans the input and filters output for their power and compares them to simulate the response times. This method was replaced by a more realistic algorithm to apply the response times – instead of simulating the response time after amplification, a control block is placed between the window power calculator and the filter select so that the SPL value inputted to the filter select block lags behind the actual dB number in the power calculator by the response times. This effectively generates the response time effect at the output. The layout of the single channel compressor can be seen in Figure 3-1.

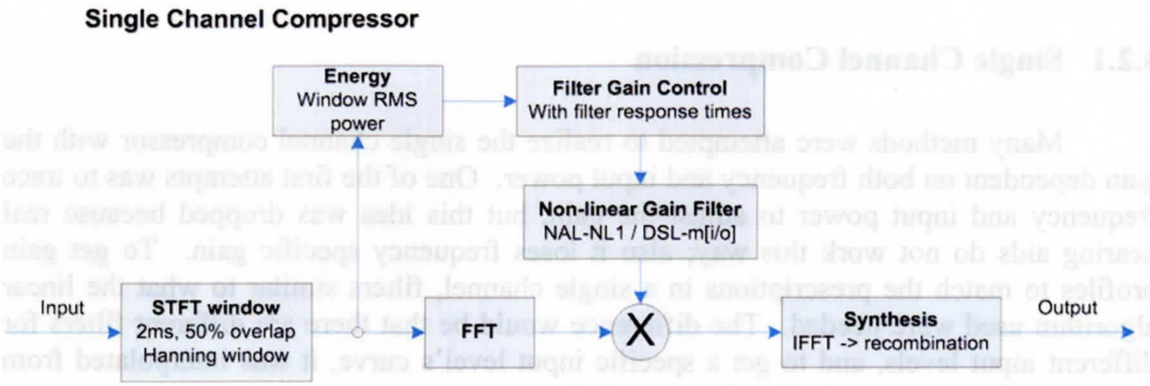


Figure 3-1: Single channel compressor layout; input is a pressure vector in units of Pascal, sectioned into overlapping Hanning windows for STFT calculations, the windows were sent to both an energy calculator and the FFT calculator. The calculated energy was then processed to add in respond times and sent to prescription block where the gain filter would be selected to apply to the transformed window. The process were repeated for each window and were synthesized and recombined to generate the output signal. Output signals were in units of Pascal and were real ear pressure signals.

3.2.2 Multi Channel Compression

In a similar manner to the design prior to the use of the non-linear prescription software, the input signal passes through a filter bank to separate the given number of channels the prescription requires and the crossover frequencies are given by the prescription. In each channel the signal then passes through a compressor based on an input-output curve determined by the prescription for each channel. In contrast to the single channel compressor, the multi-channel compressor is I/O curve based. For every input level, there would be an ideal output level according to the I/O curve, and the gain difference between the input and the output was the control variable to simulate attack and release times. To track the input level, a sliding window of 1 ms long was used to calculate the RMS value.

After amplifying the input in each channel, the input would then be summed together to obtain the output signal. The ideal simulation was if each channel was computed in parallel, but it was done one channel at a time. Since the project does not require the design to run in real time, the multi-channel compressor was left running in series. Output limiting for this multi-channel compressor was done in each channel because the limit was one of the characteristics of the compression curve used; a peak clipper was added to each channel and again after channel summation. The layout of the single channel compressor can be seen in Figure 3-2. A 4-channel compressor was used in this study.

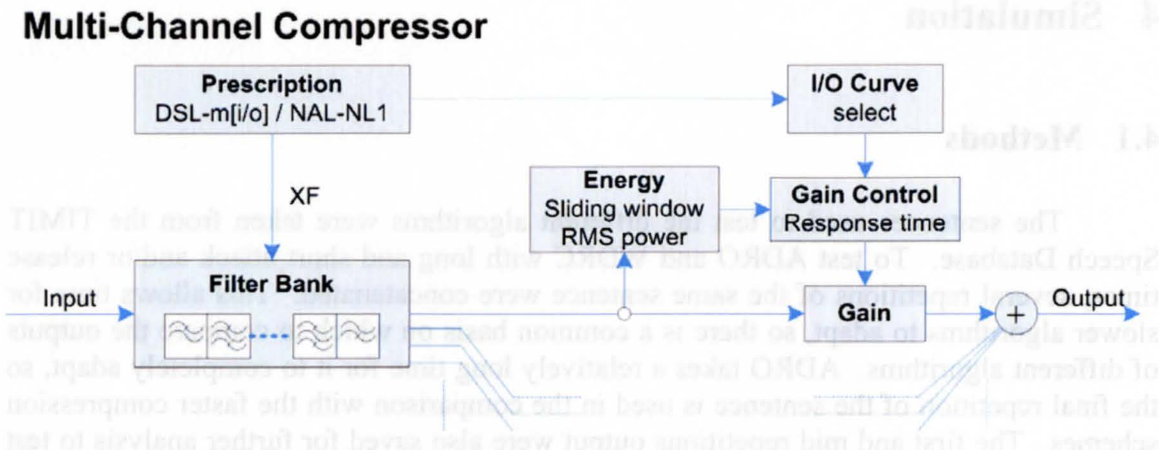


Figure 3-2: Multi-channel compressor layout; input is a pressure vector in units of Pascal, first filtered into several different bands by the filter bank. The spacing of bands was prescribed either by DSL m[i/o] or NAL-NL1; the prescription computes a specific I/O curve for each channel. For each channel the energy was tracked by a sliding window calculator and each value was compared with the I/O curve to get the ideal output level and the ideal gain to apply to get such level, the response time were added to the gain value by the gain control. The value was then applied to the signal sample by sample. At the end, the signals in each channel were summed to get the output signal.

3.2.3 ADRO

ADRO's design was provided by Dynamic Hearing as a MATLAB Simulink model. In addition to the Simulink model, a number of MATLAB script files and instructions were given to calibrate and to adjust the model. To set the model to a particular hearing loss profile, Dynamic hearing provided a Manufacturer's Toolkit to set the ADRO's target values to fit to the profile. Those target values include the comfort targets, audibility targets, maximum output level, and maximum gain. To run the simulations, all the target values for a particular loss profile are loaded into MATLAB's Workspace before running the model in Simulink. The model is calibrated so that the input and output signals are vectors describing sound pressure waveforms in units of Pascal, consistent with the input and output of the compression algorithms.

The ADRO model is designed to replicate the manufacturer or audiologist's testing of a real hearing aid system, such that the output of the simulation is in 2cc coupler units. In order convert back into real ear SPL, a frequency dependent gain filter is applied to the ADRO output to compensate for the Real Ear to Coupler Difference (RECD). The values of the RECD is given by the Manufacturer's Toolkit for a particular hearing aid type (in this case, the in-the-ear type), these values corresponding to the average values of RECD in Dillon's text [Dillon 2001].

4 Simulation

4.1 Methods

The sentences used to test the different algorithms were taken from the TIMIT Speech Database. To test ADRO and WDRC with long and short attack and/or release times, several repetitions of the same sentence were concatenated. This allows time for slower algorithms to adapt, so there is a common basis on which to compare the outputs of different algorithms. ADRO takes a relatively long time for it to completely adapt, so the final repetition of the sentence is used in the comparison with the faster compression schemes. The first and mid repetitions output were also saved for further analysis to test the effects of adaptation changes over time. The speech signals were modified to simulate different real life situations to test the effectiveness of the compression schemes. Different speech intensities and signal-to-noise ratios (SNRs) were evaluated for a range of background noises, such as speech-shaped Gaussian noise, and multi-talker babble (babble soundtrack taken from the Connect Speech Test (CST)). Once the test sound inputs were generated, the sentences were processed by software simulations of ADRO and single-band and multi-band compression schemes with a range of attack and release times. The compressed sounds were then passed through the impaired auditory models (with a variety of hearing loss profiles) and then to a speech intelligibility predictor.

In a similar manner to the hearing aids simulation design, the overall simulation design was modified several times before the current version was finalized. Initially, every simulation is very long (10+ seconds), requiring a tremendous amount of computation power and memory. This simulation will simply overload the computer even if simulations were run on the Blade server. Even if some of the nodes on the grid can run the full 10+ seconds of data, it will take 3 to 5 days just to complete. Therefore in order to allow even a random node to complete a job, the full length of data was sectioned into two or three parts. To compensate with the AN fiber adaptation, the sectioned parts (except the first) were prefixed with about 32 ms of the previous section. The sectioned neurograms were recombined with care to ensure the output would be contingent with a full length non-sectioned neurogram. However, it was finally decided to just model responses to a single repetition of the sentence at the output of the hearing aid simulator. In addition to the new method of computing the simulations, the performance bottle neck of the auditory periphery model was improved so the performance increased by about 5 times. This improvement was due to a reduction of sampling rate of one of the model's filters. It was using 500 kHz sampling rate and now it can use 100 kHz.

4.2 Simulation Runs

To start the simulations, the script **sim_ag.m** is called with a single string of input that defines all the variable parameters in the simulations such as sentence number, speaker ID, speech dB SPL, noise type, noise level, audiogram, and prescription information. An example of such input would be *sim_ag('SX255DR2MRHL0062BAL09Nxx')*. The first part 'SX255DR2MRHL0' is the reference info of the TIMIT sound file, *SX255* is the sentence reference, *DR2* is the speaker region reference, and *MRHL0* is the speaker reference. *062* is the input dB SPL, *BA* is the type of noise, *L* is the level of noise, *09* is the audiogram index, and Nxx is the prescription. If the simulation is for DSL m[i/o] or NAL-NL1, the prescription variable name would be different with the attack and release times added, for example, 'SX255DR2MRHL0062BAL09NSC_5_25'. *5* and *25* are attack and release times respectively in milliseconds. See Table 4-1 for the range of different input parameters. Once the sim_ag.m function is called, the script will then carry out all the necessary operations to get the final output. The one exception is ADRO simulations, because the ADRO simulation program is in Simulink and it needs to be run separately from the rest of the script. ADRO simulations will be discussed later. The input for sim_ag.m may seem long; however the input string is also the name of the file that the data would be saved in. This reduces the naming confusion and sets a standard for all simulation runs effectively creating a unique identification system for all simulations. The overall design can be found in Figure 4-4 and this first input step is represented by the top input block in the middle column, it is marked by the "start of simulation". The flanking columns are both pre-processing steps to get the final results which will be discussed later.

TIMIT ID	**dB SPL**	**Noise type**	**Noise level**	**Audio-gram**	**Prescription**	
SX255DR2MRHL0	052	xx	x	##	xxx	(none)
(Sentence id/	062	WG	L		Nxx	(NAL-RP)
Speaker region /	074	BA	M		NSC	(NAL-NL1 single-channel)
Speaker id)	082		H		NMC	(NAL-NL1 multi-channel)
					Dxx	(DSL)
					DSC	(DSL m[i/o] single-channel)
					DMC	(DSL m[i/o] multi-channel)
					AAA	(ADRO)

Table 4-1: Range of different input parameters for simulation. Attack and release times are separated by an underscore in font of the number, and it is only apply to DSL m[i/o] and NAL-NL1; single and multi channel.

If the input demanded an ADRO output, the simulation will not generate a sound signal; it would immediately load the pre-processed data from the other scripts and Simulink. It would input the sound file based on the specified parameter given, and then it would be adjusted to the desired SPL. The simulation uses the sound pressure levels of 52, 62, 74, and 82 to cover most SPL of everyday speech. The first three levels are the same levels used by Zilany and Bruce [2007] for their simulations; and 82 dB SPL was added to better represent each hearing aid's high SPL handling ability. The SPL values were calculated by using the full length of the sound signal as the RMS window, effectively taking the average SPL of the signal. To set the desired SPL, the sound signals were simply multiplied by a constant to get the desired level.

The next step is the addition of noise to the signal; this step would simply do nothing if no noise was required. The different noise setting in this study involves three different types: noiseless, white Gaussian, and babble. (More noise types can be added in the future.) The latter two would also be presented at three different levels: low, medium, and high; or 6 dB, 3 dB, and 0 dB SNR (signal-to-noise ratio). The noise types and noise level together add up to a total of seven simulation conditions. The noises were generated with the same length as the speech signal, and then the SPLs were adjusted relative to the speech signal according to the noise level. Then, the noise signal is simply added to the speech signal. After the noise adding steps, the signal is ready to be placed in one of the amplification schemes to simulate the output of a hearing aid. Below the amplification decision block are four paths: no amplification, NAL-RP, DSL, and non-linear aids. For no amplification, a Head Related Transfer Function (HRTF) filter block was added to the path to convert the free-field SPL to ear drum SPL [Wiener and Ross, 1946], the REUG. In this study, the HRTF only compensates for the REUG and it does not adjust for diffraction and reflection of sound. The REUG from HRTF, NL1, and DSL m[i/o] are shown in Figure 4-2.

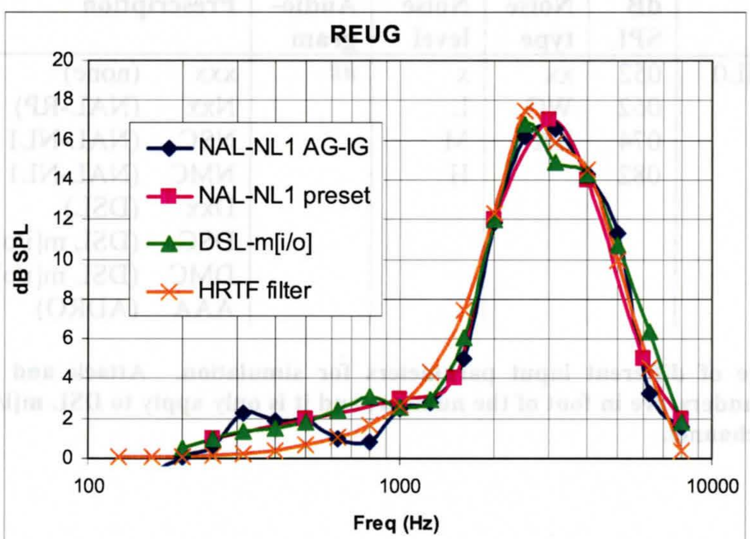


Figure 4-2: REUG Transfer function for prescriptions: NAL-NL1, DSL m[i/o], and HRTF.

The second path is the NAL-RP scheme; this path also includes the HRTF block because NAL-RP prescriptions are in the form of insertion gain. The third path is the DSL scheme; unlike NAL-RP, DSL prescriptions are already in the form of REAG, therefore the HRTF block is not needed to convert signal to eardrum level. The fourth path is for non-linear hearing aids other than ADRO, which has its own separate path outside of the amplification scheme decision block. Depending on what non-linear prescription was inputted at the start of the simulation, a specific fitting file would be loaded into the amplification script. There are two different scripts, one for single-channel, and the other for multi-channel. The details for those two scripts were described in the hearing aid simulation design section. Whether it is the NAL-NL1 prescription or the DSL m[i/o] prescription, the method for running them is identical in the simulations, the differences are in the data in the prescription. Also, the single-channel compressors are filter based, so inputs are multiple gain frequency curves, where as the multi-channel compressors are I/O curve based, so inputs are multiple I/O curves; the script to handle the different inputs are also different for single and multi channel compressors. The curves are manually taken from the prescription's fitting program, modified in Excel to fit the design. Figure 4-5 and Figure 4-6 are the modified curves used by the scripts to apply the amplification, and the prescription was for a hearing loss profile shown in Figure 4-3.

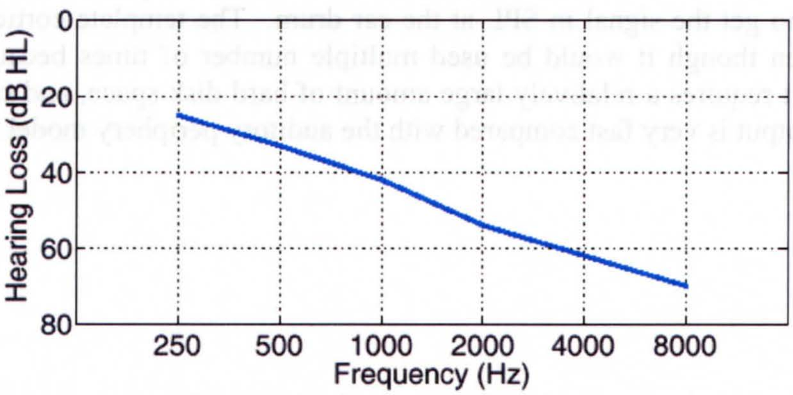


Figure 4-3: The hearing loss profile used for simulations. The threshold shift at each CF was created by 2/3 OHC impairment and 1/3 IHC impairment.

The modifications were done to fill the dynamic range from 0 dB SPL to 130 dB SPL because the fitting from the NAL-NL1 and DSL m[i/o] only has an input dynamic range of about 40–90 dB SPL. The curves beyond 90 dB SPL was extrapolated (linear extrapolation) all the way to its maximum output level or until the input range of 130 dB SPL, whichever came first. If the maximum output level is reached, the input-output curve would be flat all the way until the input range of 130 dB SPL. For curves below 40 dB SPL, the same linear extrapolations were done, and the range extends to 25 dB SPL. 25 dB SPL and down, the input-output curve will enter expansion (opposite of compression). Most people with normal hearing would consider such levels as nearly inaudible; also, speech information usually does not fall into this range of SPL. Depending on where the curve is at 25 dB SPL, the expansion curve will be a line from the 25 dB SPL point to 0 dB input and output. Expansion was used instead of a straight line from extrapolation mainly because of the noise reduction property of low level expansion. Expansion reduces a lot of "clicking" noise caused by sudden high level sounds within a period of quietness, because long period of quietness sets the amplifier gain very high and the gain will not reduce instantaneously because of the non-zero attack time.

At this point, the signal is in the form of Real Ear SPL or eardrum level, and it is at the input level to be entered into the auditory periphery model to continue the simulation. The ADRO pre-processed path converges at this point as well loading the pre-processed ADRO signal in real ear SPL. The periphery model is the performance bottle neck of the entire simulation processes, however during the course of this project, the periphery model received an update, which speeded up the periphery model process by about 5 times. The neurograms were computed at 128 different best frequencies at every 8 ms (bin widths are 16 ms with 50% overlap). The neurogram ("noisy/test" neurogram) is then passed into the cortical model by Elhilali et al. [2003] to do the spectro-temporal modulation analysis. The same thing would be done to the saved clean and noiseless neurogram template at 65 dB SPL ("clean/template" neurogram). The template neurogram was pre-computed as shown in Figure 4-4's top left side column; it is very similar to the main computation path, but with the decision blocks taken out and replaced by set blocks. It is noiseless at 65 dB SPL, and it was passed through the HRTF filter as well to get the signal in SPL at the ear drum. The template cortical output was not saved even though it would be used multiple number of times because saving the cortical output requires a relatively large amount of hard disk space, and computation of the cortical output is very fast compared with the auditory periphery model.

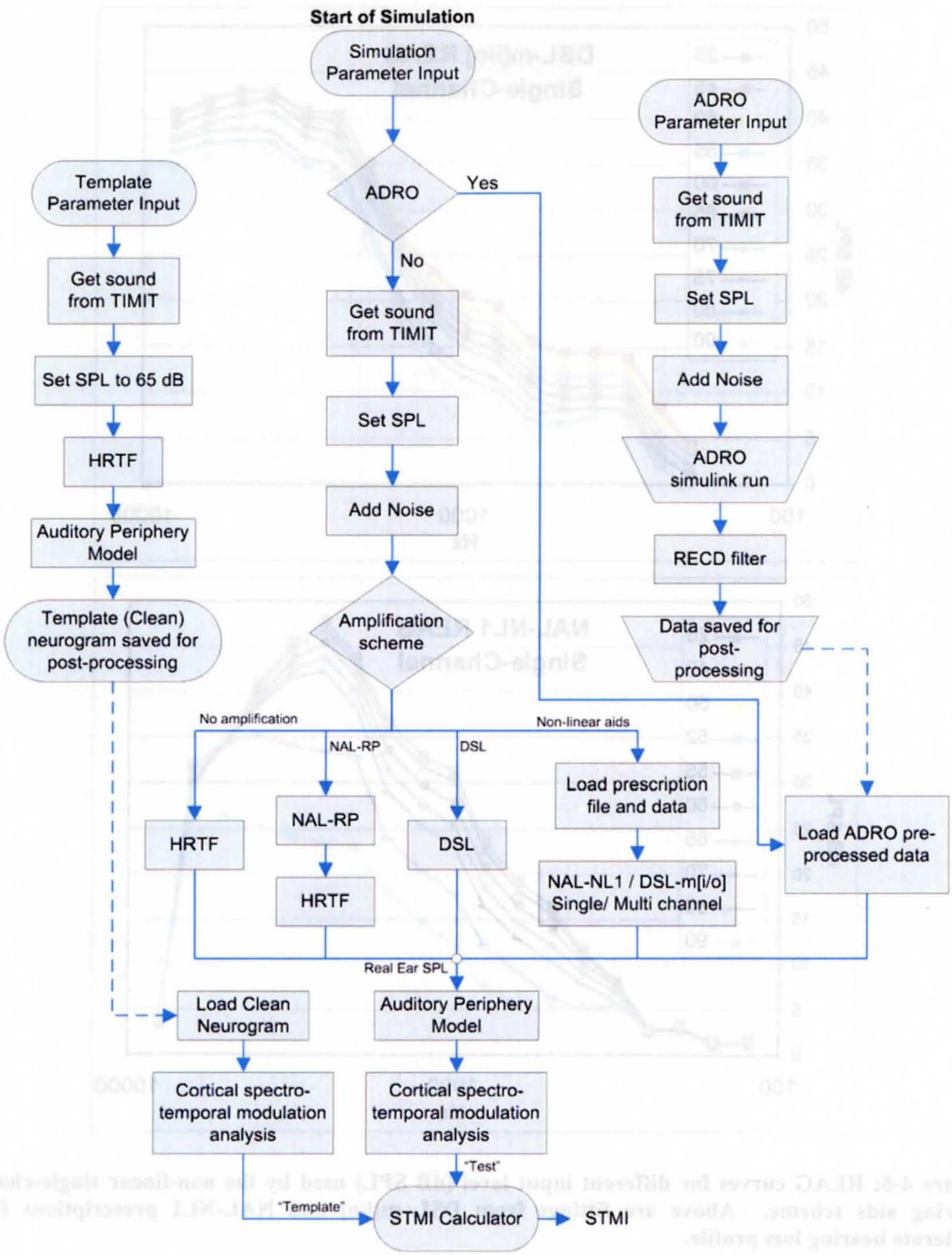
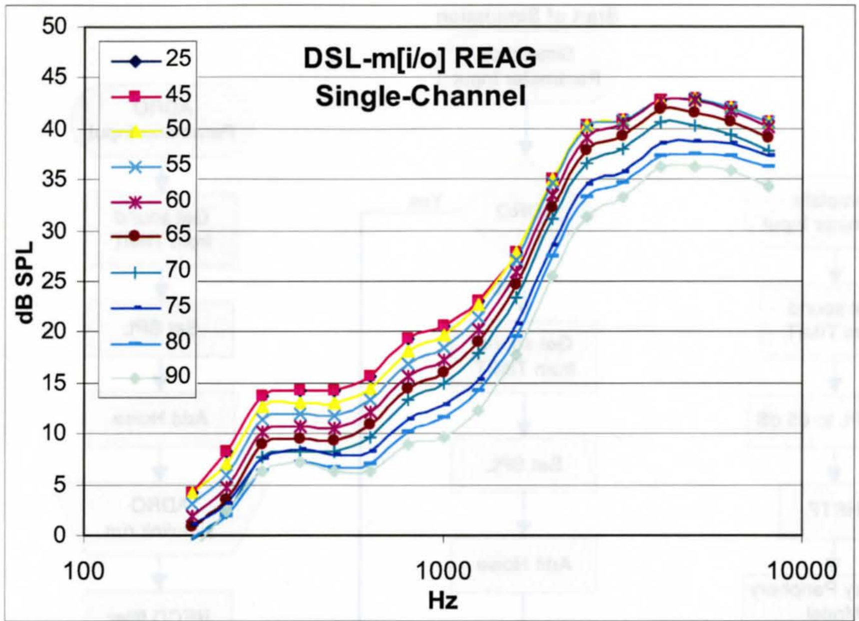


Figure 4-4: Overall follow diagram of the simulation design. The start of the simulation is labeled on top of the main starting input block. The left side column is the pre-processing that needs to be done to obtain the template neurogram. The right side column is the pro-processing for ADRO.



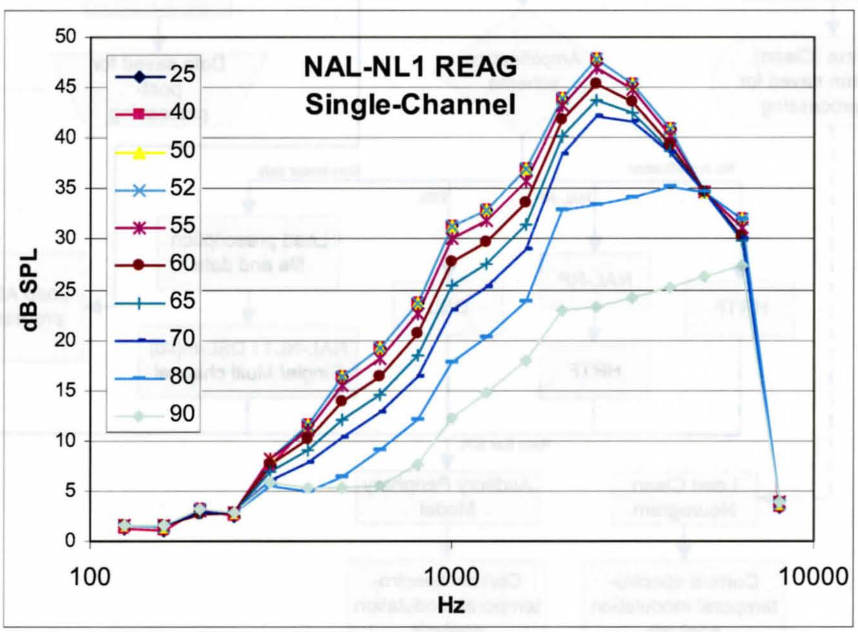


Figure 4-5: REAG curves for different input level (dB SPL) used by the non-linear single-channel hearing aids scheme. Above are fittings from DSL m[i/o] and NAL-NL1 prescriptions for a moderate hearing loss profile.

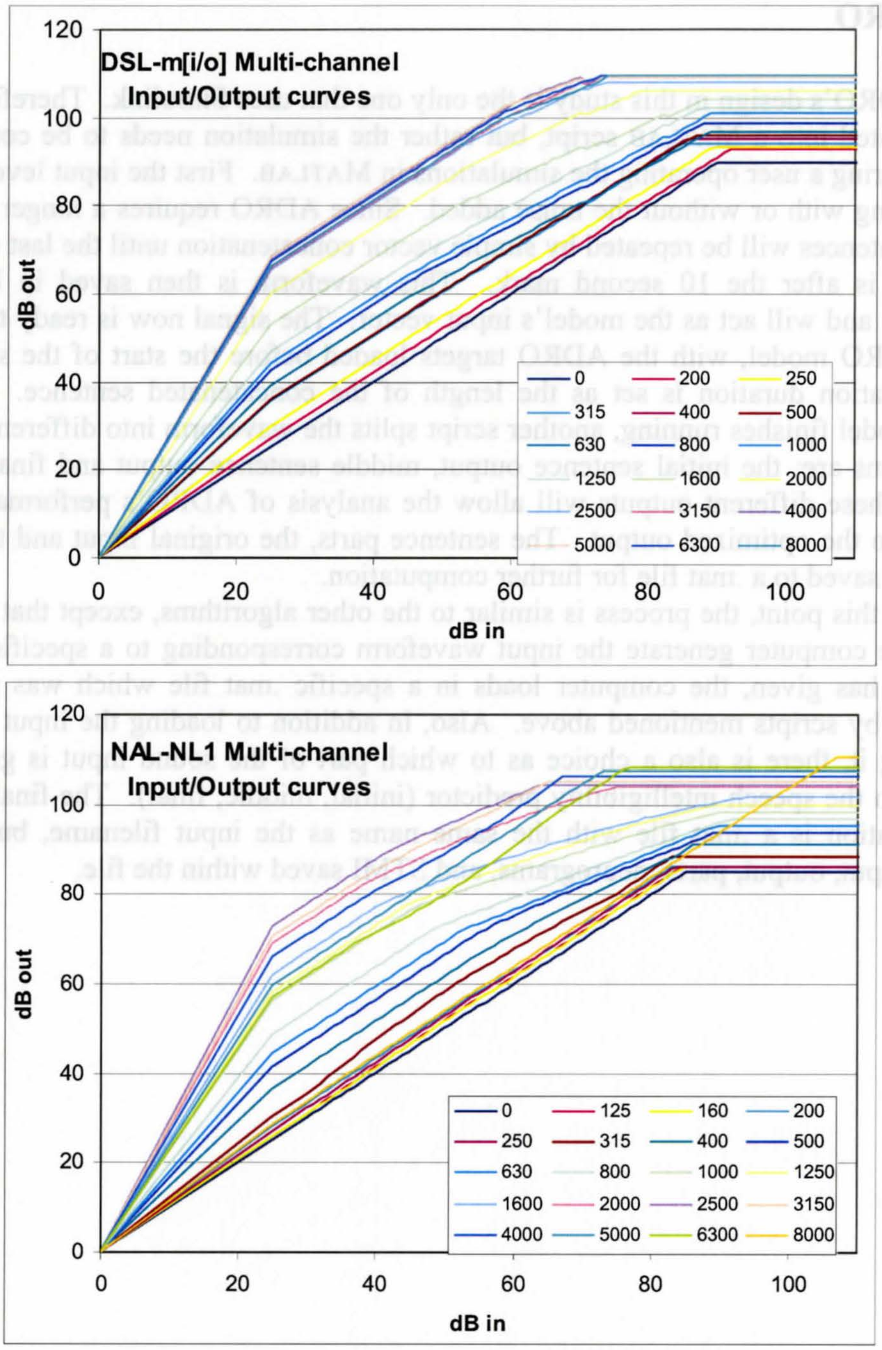


Figure 4-6: Input/output curves used by the non-linear multi-channel hearing aids scheme. Above are fittings from DSL m[i/o] and NAL-NL1 prescriptions for a moderate hearing loss profile.

4.3 ADRO

ADRO's design in this study is the only one that uses Simulink. Therefore, it was not integrated into a MATLAB script, but rather the simulation needs to be computed in parts requiring a user operating the simulations in MATLAB. First the input level needs to be set, along with or without the noise added. Since ADRO requires a longer input, the TIMIT sentences will be repeated by simple vector concatenation until the last sentence's start time is after the 10 second mark. This waveform is then saved in MATLAB's workspace and will act as the model's input vector. The signal now is ready to be input to the ADRO model, with the ADRO targets loaded before the start of the simulation. The simulation duration is set as the length of the concatenated sentence. Once the ADRO model finishes running, another script splits the waveform into different sections. The sections are: the initial sentence output, middle sentence output and final sentence output. These different outputs will allow the analysis of ADRO's performance while adapting to the optimized output. The sentence parts, the original input and the ADRO output are saved to a .mat file for further computation.

At this point, the process is similar to the other algorithms, except that instead of having the computer generate the input waveform corresponding to a specific filename the script has given, the computer loads in a specific .mat file which was previously generated by scripts mentioned above. Also, in addition to loading the input instead of generating it, there is also a choice as to which part of the sound input is going to be pasted into the speech intelligibility predictor (initial, middle, final). The final output of the simulation is a .mat file with the same name as the input filename, but with the original input, output, parts, neurograms, and STMI saved within the file.

5 Results and Discussions

Figure 5-1 shows the noiseless input of the TIMIT sound track that was used for the simulations; the figure also shows the spectrogram view and the corresponding neurogram output. The sound pressure level was set to 65 dB SPL, so the bottom plot of Figure 5-1 was the clean or template neurogram used to calculate the STMI values. An immediate observation is that the neurogram output mimics the spectrogram output very closely. The simulations were using a moderately-high hearing loss profile (Figure 4-3).

5.1 Noiseless Results

Figure 5-2 shows the STMI data for all the amplification schemes in relation to no amplification in a noiseless environment. The STMI charts' y-axis is the STMI value, and x-axis is the speech sound pressure level. A quick observation of Figure 5-2 A shows that every amplification scheme operates very well in noiseless situations. Perhaps the one that worked the best was ADRO, followed closely by NAL-NL1 fast and slow. On the other end of the spectrum, the ones that did the least well were the multi-channel schemes at lower sound pressure levels at seen in Figure 5-2 A and D. However, the lowest ~0.7 STMI value, was still higher than all of the noisy data shown later in this chapter.

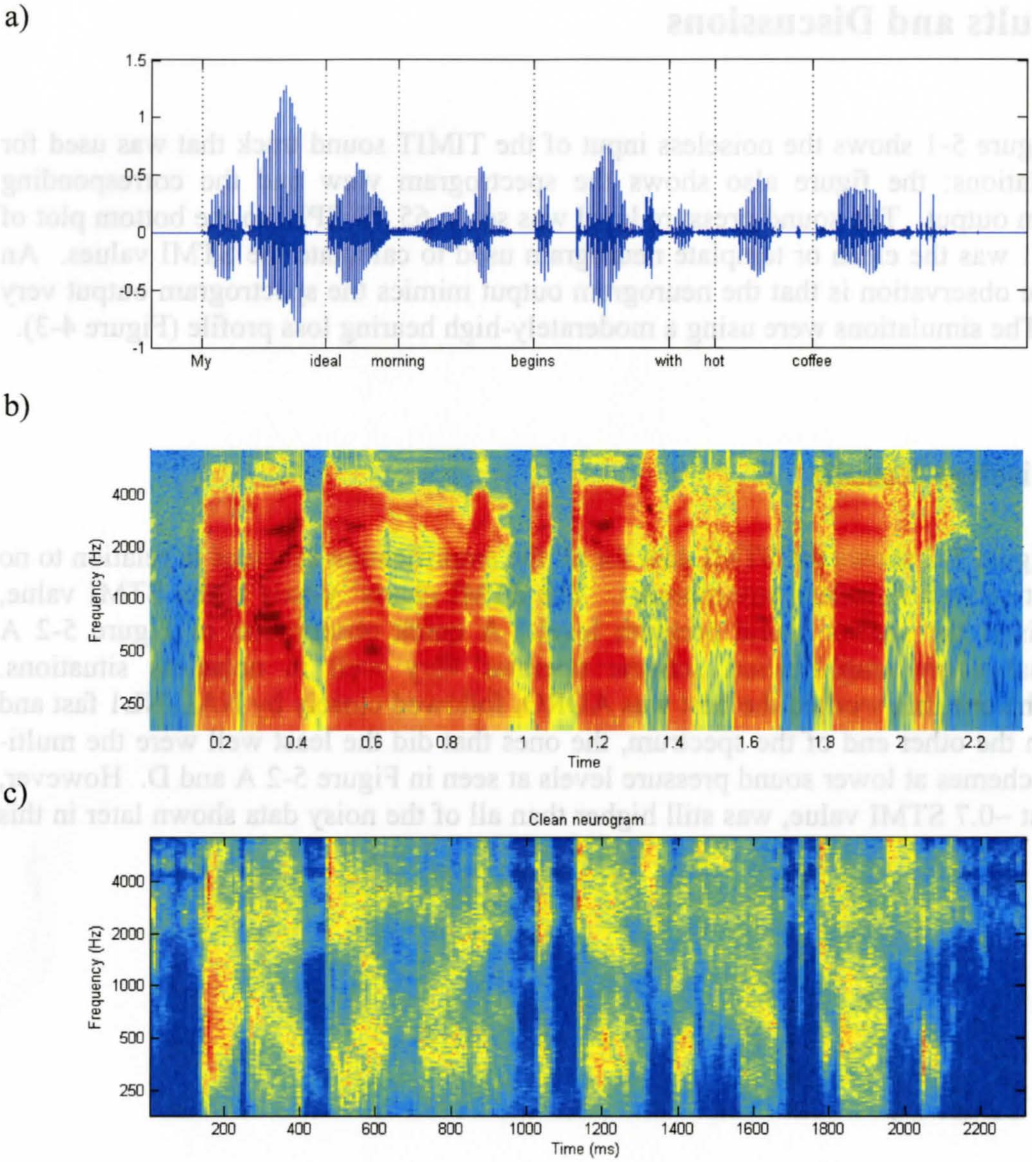


Figure 5-1: The input wave file to the intermediate output neurogram. The above plots are for normal hearing. a) Time based view of the input TIMIT file. b) Corresponding spectrogram. c) Corresponding neurogram.

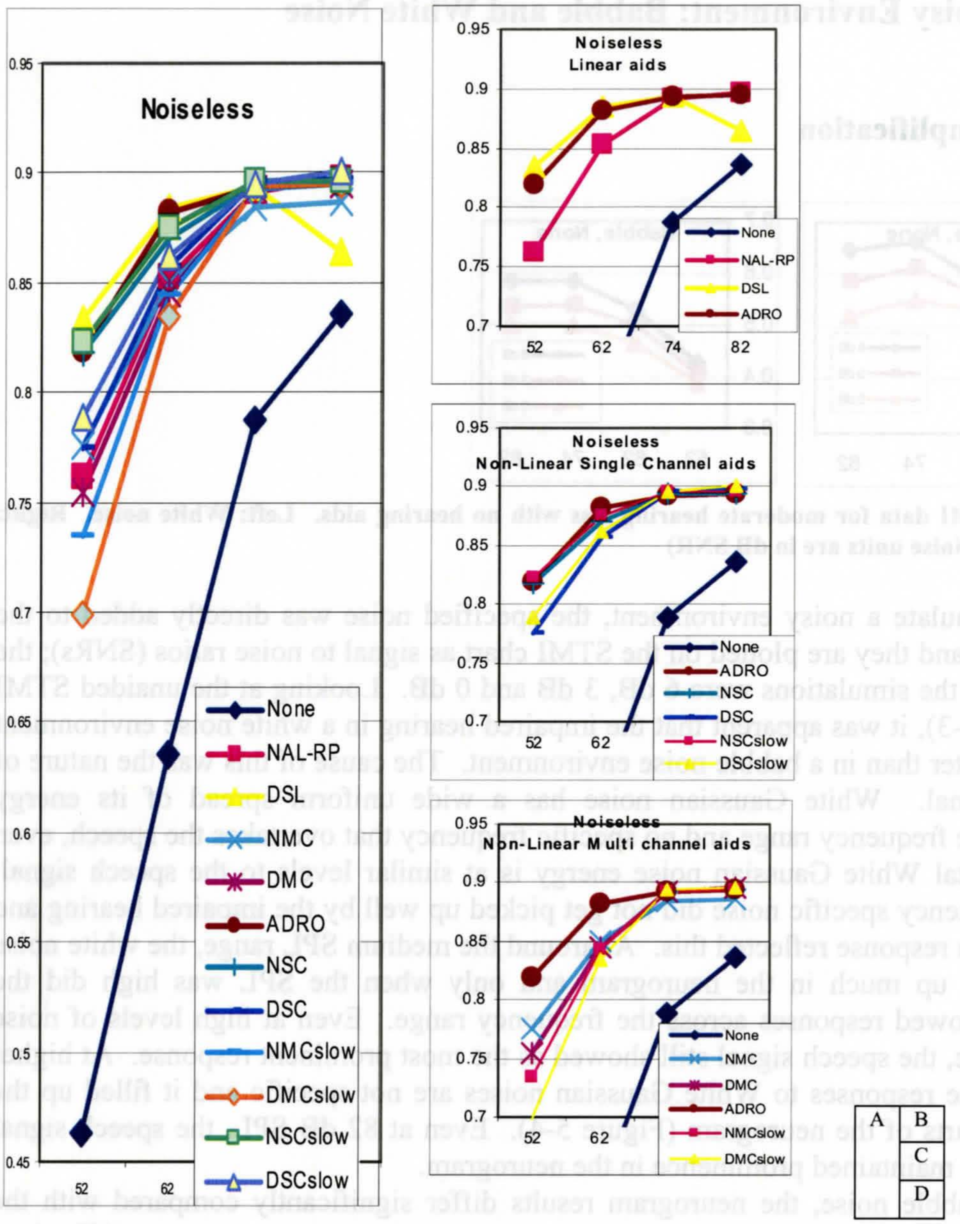


Figure 5-2: STMI values for noiseless simulations with different algorithm sets. A) A plot of all the amplification schemes. B) Linear aids. C) Non-linear single channel aids. D) Non-linear multi channel aids.

5.2 In a Noisy Environment: Babble and White Noise

5.2.1 No Amplification

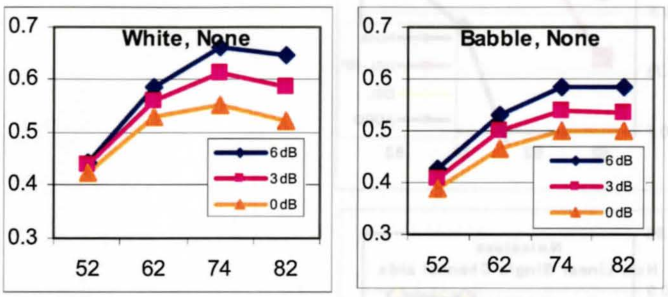


Figure 5-3: STMI data for moderate hearing loss with no hearing aids. Left: White noise. Right: Babble noise. (Noise units are in dB SNR)

To simulate a noisy environment, the specified noise was directly added to the speech signal and they are plotted on the STMI chart as signal to noise ratios (SNRs); the SNRs used in the simulations were 6 dB, 3 dB and 0 dB. Looking at the unaided STMI data (Figure 5-3), it was apparent that the impaired hearing in a white noise environment performed better than in a babble noise environment. The cause of this was the nature of the noise signal. White Gaussian noise has a wide uniform spread of its energy throughout the frequency range and no specific frequency that overtakes the speech, even though the total White Gaussian noise energy is at similar levels to the speech signal. The non-frequency specific noise did not get picked up well by the impaired hearing and the neurogram response reflected this. At around the medium SPL range, the white noise did not show up much in the neurogram and only when the SPL was high did the neurogram showed responses across the frequency range. Even at high levels of noise and SPL value, the speech signal still showed as the most prominent response. At higher SPL range, the responses to White Gaussian noises are not specific and it filled up the non-speech parts of the neurogram (Figure 5-4). Even at 82 dB SPL, the speech signal seems to have maintained prominence in the neurogram.

For babble noise, the neurogram results differ significantly compared with the white noise neurogram (Figure 5-5). From visual examination, it was more difficult to distinguish the speech response and the noise response because the babble noise's frequency ranges were similar with the speech signal and the babble noise has some speech quality to it; even at 6 dB SNR, the response of the babble noise was significant relative to the speech response. It modulates independently of the speech, thus lowering the STMI.

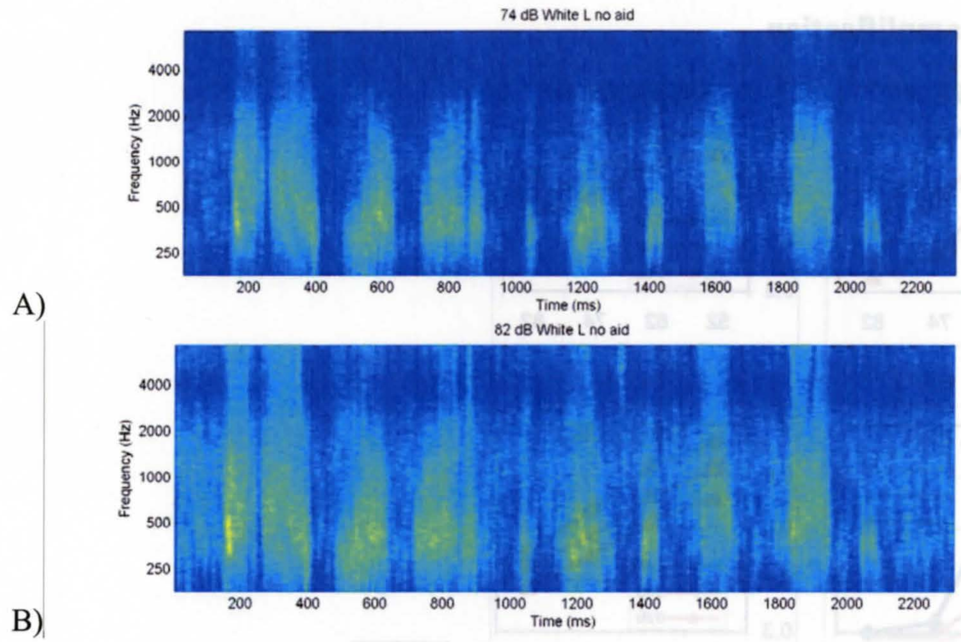


Figure 5-4: Neurogram of moderately high impaired ear in white noise (SNR = 6dB SPL) conditions at a) 74 dB SPL and b) 82 dB SPL.

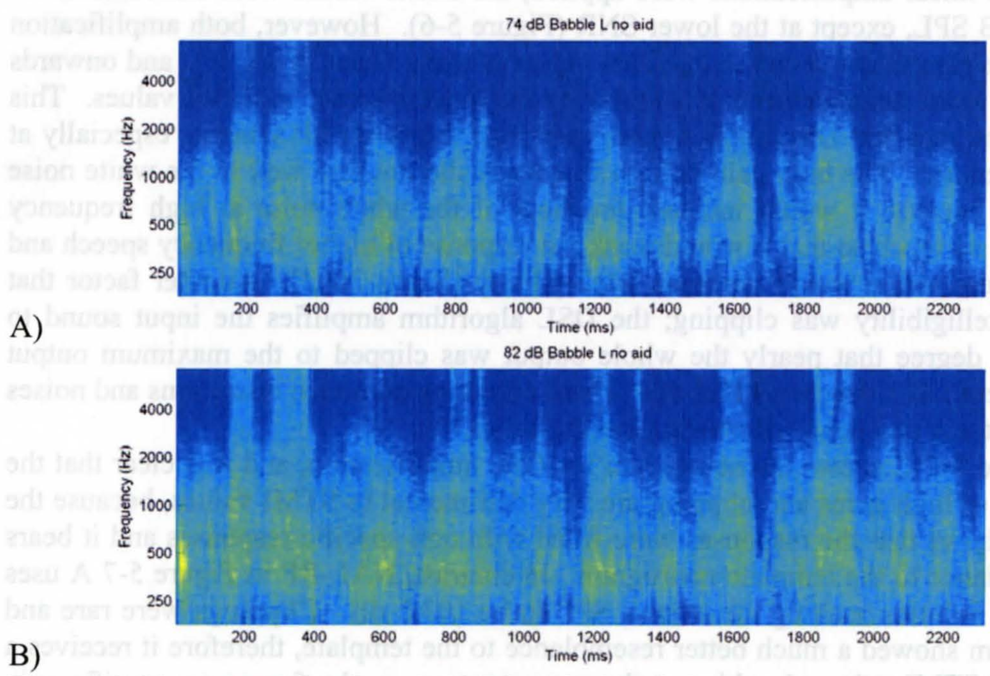


Figure 5-5: Neurogram of moderately high impaired ear in babble noise (SNR = 6dB SPL) conditions at a) 74 dB SPL and b) 82 dB SPL.

5.2.2 Linear amplification

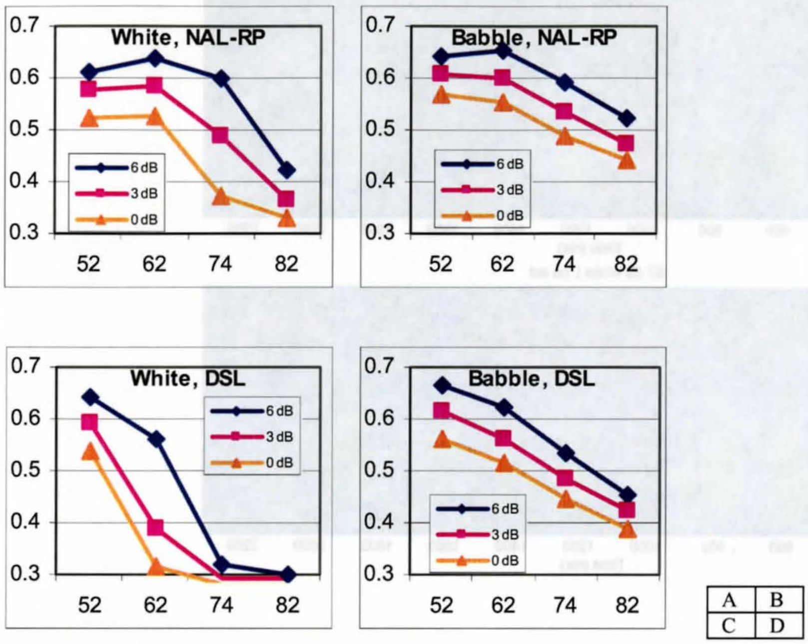
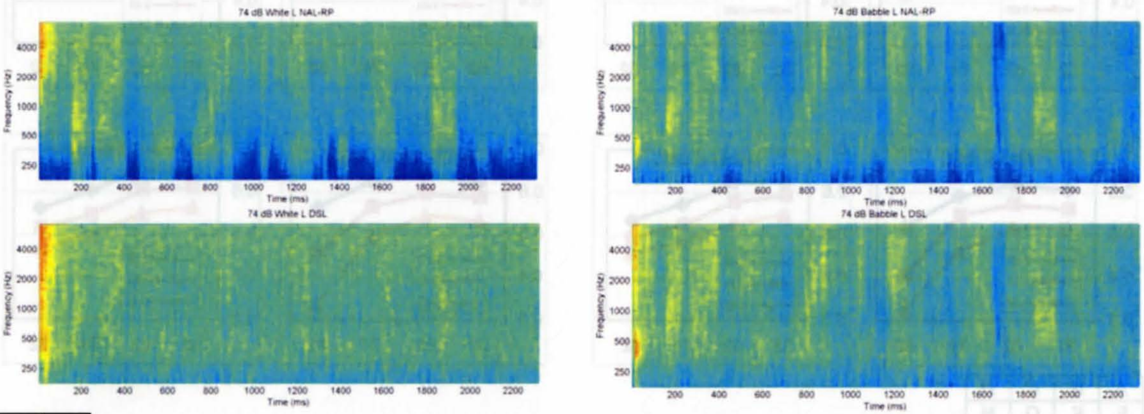


Figure 5-6: STMI values for linear amplification. A) NAL-RP with white and B) babble noise respectively. C) DSL with white and D) babble noise respectively. (Noise units are in dB SNR)

When linear amplifications were applied, the STMI values were increased at 52 dB and 62 dB SPL, except at the lower SNR (Figure 5-6). However, both amplification schemes have significant decrease in STMI values starting from 74 dB SPL and onwards with DSL in white noise environment suffering the most decrease in STMI values. This was due to the fact that DSL applies more gain than the NAL-RP scheme, especially at higher frequencies. The high gain at high frequency did not fare well in the white noise environment because it would increase loudness of the white noise at high frequency significantly, so much so that it would mask the response of higher frequency speech and mask formants, which would decrease intelligibility [Sachs 2002]. Another factor that decreases intelligibility was clipping; the DSL algorithm amplifies the input sound to such a large degree that nearly the whole output was clipped to the maximum output level; the overall SPL reached 110.3 dB. These clippings introduce distortions and noises in the sound track and hence decreasing intelligibility.

Figure 5-7 C shows the neurogram for DSL amplification, and it is clear that the combination of high gains and clipping are very detrimental to STMI values, because the neurogram shows that the responses were filled with non-specific responses and it bears little resemblance to the template neurogram. In contrast, NAL-RP in Figure 5-7 A uses a lesser gain solution making the overall SPL to be 103.2 dB. Clippings were rare and the neurogram showed a much better resemblance to the template, therefore it receives a much higher STMI value. Looking at the same neurogram, the frequency specific gain was clearly visible; at lower frequencies, NAL-RP did not provide much gain and the speech response was much stronger than the white noise at lower frequencies. At higher frequencies, the response was very similar to DSL but more details can be seen.

DSL fared a lot better in the babble noise case with significant improvement in STMI values compared with the white noise response. With fewer high frequency components from the babble noise, DSL was able to increase intelligibility with greater success, but NAL-RP seems to be able to do better in this situation. By looking at Figure 5-7 B and D, NAL-RP's neurogram showed a clearer representation of the speech signal. This is in part due to the lesser overall gain of the NAL-RP and also lesser gain at lower and higher frequencies. The overall SPL of NAL-RP's output was 99.2 dB, whereas DSL has an overall SPL of 103.5 dB.



A	B
C	D

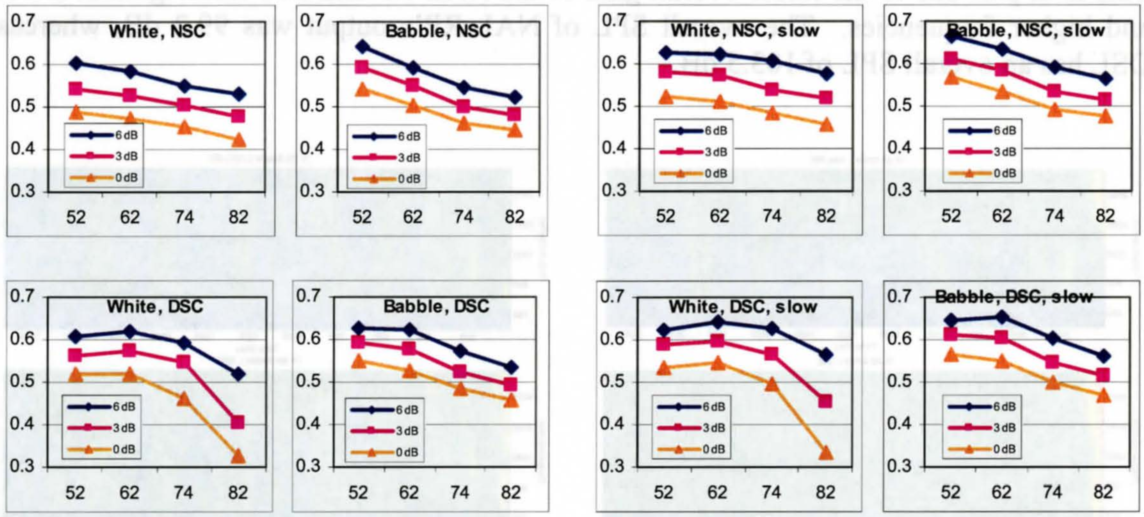
Figure 5-7: DSL and NAL-RP's neurogram responds at 74 dB SPL with white or babble noise. A) NAL-RP with white and B) babble noise. C) DSL with white and D) babble noise. SNR = 6 dB SPL

5.2.3 Non-Linear Amplification

5.2.3.1 Single Channel Compression

A non-linear amplification scheme should work better than its linear counterparts when the input sound pressure level gets high, because it compresses the dynamic range output to the ear and avoids the problem of being too loud and over saturating the amplifier, which will clip the sound. However, compression is a form of distortion in itself and therefore should decrease intelligibility in non-linear aids at lower input sound pressure levels. Looking at the general trends on all the graphs on Figure 5-8 and comparing the results of the non-linear aids with the linear aids suggests that linear aids increase intelligibility more at lower input sound pressure levels, and non-linear aids increase intelligibility more at higher input sound pressure levels. Other than having the

STMI between lower and higher level balancing out, the intelligibility difference between the noises were balancing as well. With the volume managed by compression and gain limiting, the intelligibility in white noise environment was again better than babble noise environment just like in the unaided situation. This again suggests that modulation in the babble noise signals were significant enough to lower intelligibility more than white noise.



A	B	C	D
E	F	G	H

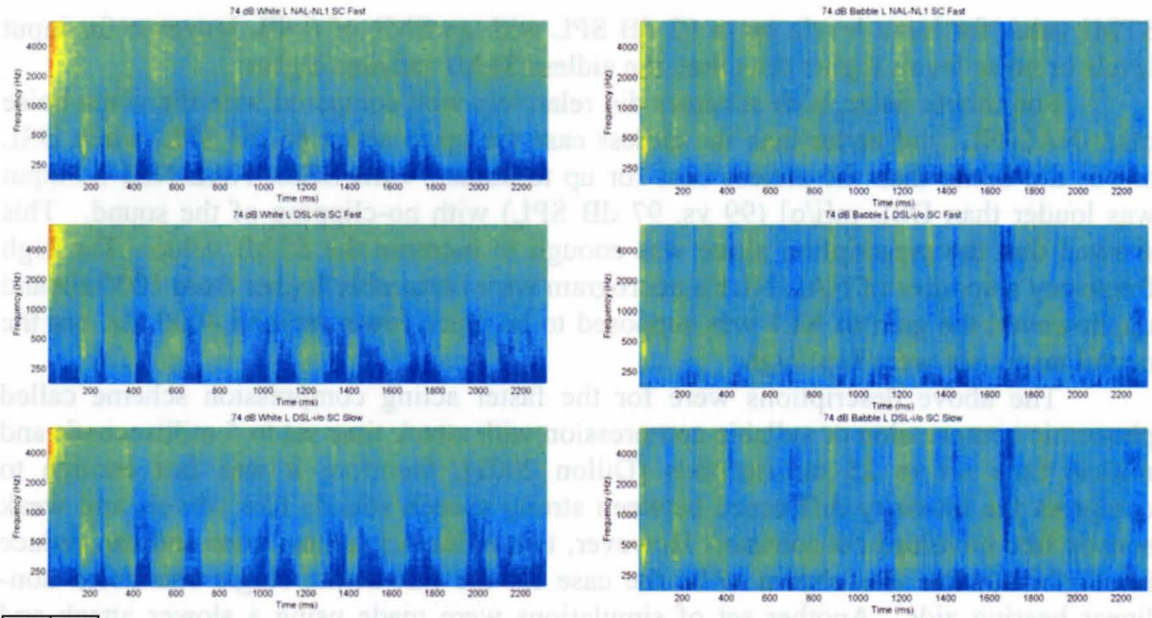
Figure 5-8: STMI values for single-channel amplification. A) NAL-NL1 with white and B) babble noise respectively. C) NAL-NL1 (with slow attack and release times) with white and D) babble noise respectively. E) DSL m[i/o] with white and F) babble noise respectively. G) DSL m[i/o] (with slow attack and release times) with white and H) babble noise respectively. (Noise units are in dB SNR)

There were fewer differences in STMI values between the two different amplification schemes than there were in the linear case, however, the DSL m[i/o] still suffered the drop in STMI value at 82 dB SPL; especially for the white noise, but it was much better than the linear DSL case. As for NAL-NL1 in the white noise environment, there was really no significant improvement from input sound pressure level of 62 dB and above, because STMI level of the aidless case at 62 dB SPL was already at about 0.6 and STMI value increases as input level increases. For NAL-NL1, it did not give any improvement in intelligibility even at 6 dB SNR at input sound pressure level of 62 dB, and STMI decreases as input level and noise level increase. The amplification of high frequency white noise was again the cause of the decrease of STMI value; Figure 5-9 A is the neurogram for NAL-NL1's output, the high frequency responses were significant compared with the lower frequency speech responses. DSL m[i/o] in Figure 5-9 C has a similar response with less gain in the high frequency band, therefore it received a higher

STMI value for input levels up to 62 dB SPL with an SNR of 6 dB, however for input levels or noise levels higher than that, the aidless STMI remains higher.

For babble noise, both schemes did relatively well compared with the white noise case: NAL-NL1 did better than the aidless case for up to about 66 dB SPL, while DSL m[i/o] did better than the aidless case for up to about 74 dB SPL. NAL-NL1's output was louder than DSL m[i/o] (99 vs. 97 dB SPL) with no-clipping of the sound. This showed that the prescription alone was enough to increase the STMI value. The high frequency responses of NAL-NL1's neurogram were noticeably higher from 1000 Hz and up, however, the gain of NL1 was supposed to be much lower beyond 4000 Hz, but the neural responses were kept high.

The above descriptions were for the faster acting compression scheme called phonemic compression or syllabic compression with attack time set to 5 milliseconds and release time set to 25 milliseconds [Dillon 2001], therefore it was fast enough to compress the intensity difference between strong speech sounds like vowels and weak sounds like unvoiced consonants. However, this reduces loudness cues and can reduce intelligibility; this was shown to be the case for the fast acting single channeled non-linear hearing aids. Another set of simulations were made using a slower attack and release times to see how it would affect intelligibly. Slower acting compressors are sometimes referred to as automatic volume control, as they decrease the long-term level differences without changing the inter-syllabic relationships and the mean level difference between the soft and the intense speech. Setting the attack and release times to 120 and 500 milliseconds, respectively, changed the fast acting compressor to a slow one, and the STMI results of the simulation done with these new settings are shown in Figure 5-8 CDG & H. The results showed that there were overall increases in STMI value in all input levels with all the variations of SNR; it almost seems like those STMI values were just shifted upwards. This again demonstrates that phonemic compression and its act of reducing inter-syllabic level differences reduces intelligibility. Its neurogram was shown in Figure 5-9 E and F to compare with its fast acting pair in Figure 5-9 C and D; the slow neurograms were almost identical to the ones of fast acting compressor with the exception that the responses were slightly clearer and more intense.



A	B
C	D
E	F

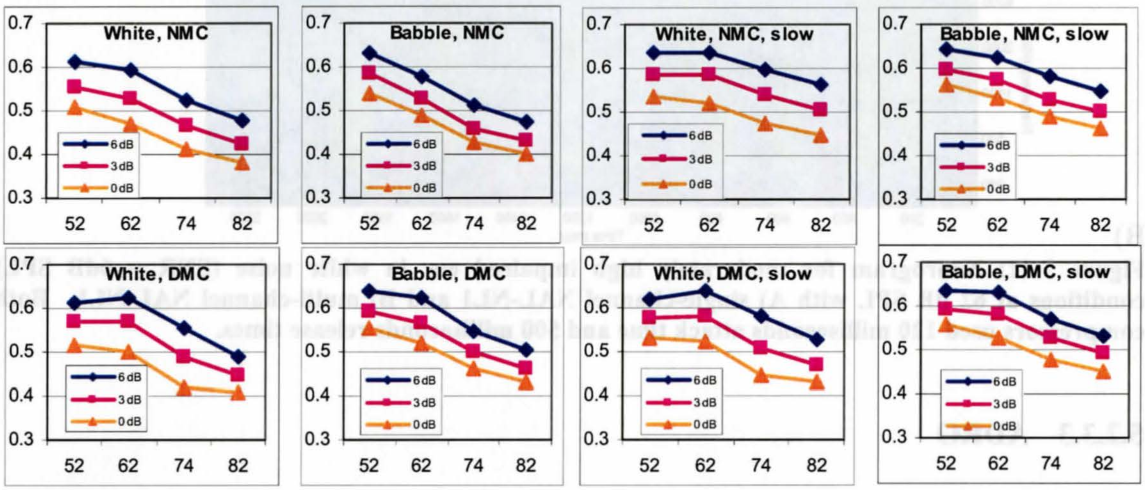
Figure 5-9: DSL m[i/o] and NAL-NL1 's neurogram responds at 74 dB SPL with white or babble noise. A) NAL-NL1 with white and B) babble noise. C) DSL m[i/o] with white and D) babble noise. E) DSL m[i/o] using slower attack and release times with white and F) babble noise. SNR = 6 dB SPL

5.2.3.2 Multi-Channel Compression

Figure 5-10 gives the result for multi-channel compression, and a quick comparison of the result here and the results of the single-channel compression shows that there is a similarity of shapes in the STMI curves, but with an earlier and steeper drop in STMI values for multi-band compression. The decrease in STMI value is may be due to the fact that the number of channels was not sufficient enough to shape gain curves to the hearing loss profile since there was only one compression curve per channel in the 4-channeled compression scheme. The implementation of more channels would enable the compression gain curves to fit to the hearing loss profile more precisely, giving more audibility to the user. However, increasing the number of channels might also increase the drawbacks of spectral flattening, and consequently intelligibility might decrease.

The STMI values were close between the single-channel and multi-channel compressors with the single-channel compressors having higher STMI values; values were closer if slower attack and release times were used. NAL-NL1 seems to receive a much bigger boost in STMI value than DSL m[i/o]. The boost made the STMI performance of the NAL-NL1 multi-channel aid near identical to the single-channel aid.

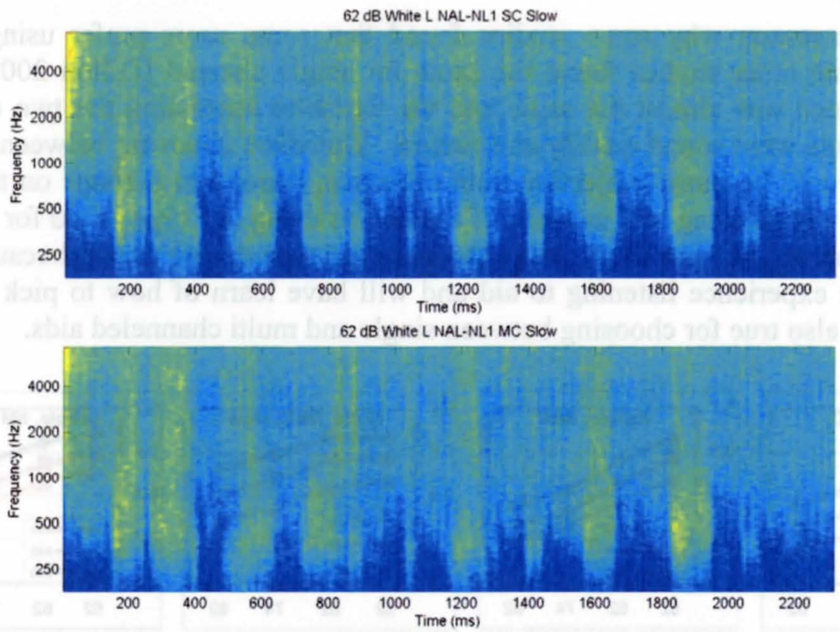
This maybe the reason why some studies found that some users prefer using multi-channel and some other studies found the same for single channel [Dillon 2001]. The STMI performance was almost the same and the variables separating the two different multi-channel aids were sound quality and timbre. Therefore choosing between the two different aids would be more subjective than objective, depend in perhaps on the users past experience with hearing aids as well. If a user was using one type of aid for a period of time, he or she would most likely prefer the same aid type during testing because he or she would have experience listening to aid and will have learn of how to pick up cues from it. This is also true for choosing between single and multi channeled aids.



A	B	C	D
E	F	G	H

Figure 5-10: STMI values for multi-channel amplification. A) NAL-NL1 with white and B) babble noise respectively. C) NAL-NL1 (with slow attack and release times) with white and D) babble noise respectively. E) DSL m[i/o] with white and F) babble noise respectively. G) DSL m[i/o] (with slow attack and release times) with white and H) babble noise respectively. (Noise units are in dB SNR)

Auditory cues such as spectral peaks and valleys were decreased by multi-channel compressors in an attempt to increase audibility of sound in each frequency channel since it amplifies weak sound more than stronger ones. Figure 5-11 showed that the multi-channel did flatten the spectral response. However, it also increased the response to the speech signal. Overall, the STMI values between the two neurogram were very similar: 0.6441 for single channel and 0.6401 for multi channel.



A) B)

Figure 5-11: Neurogram for moderately high impaired ear in white noise (SNR = 6dB SPL) conditions at 62 dB SPL with A) single-channel NAL-NL1 and B) multi-channel NAL-NL1. Both compressors used 120 milliseconds attack time and 500 milliseconds release times.

5.2.3.3 ADRO

ADRO's fuzzy logic based amplification scheme is very different to the other non-linear amplification schemes in this study, and it reaches a steady amplification level for the test signal the slowest. Therefore 10 seconds or more of the same sentence were introduced to the input of the amplifier before the start of the final sentence; the final sentence would then be the one to be tested by the intelligibility predictor. The result of delaying 10 seconds was to get the most ideal output by the ADRO amplification scheme, and the STMI values for this output are shown in Figure 5-12 A and B. The result showed that ADRO did very well in both low input levels and high input levels; also, the result has the flattest STMI curve in this study and has the best overall STMI. The chart's shape was very similar to the ones for the slow acting single channel NAL-NL1 amplifier with ADRO having a higher and flatter curve. Slow acting NAL-NL1's output was about 5 dB louder than ADRO across the SPL range for white noise, and about 3.5 dB SPL for babble noise. However, ADRO's STMI values were higher than NAL-NL1's; this showed that more audibility does not always mean more intelligibility. In Figure 5-13 A and C, ADRO's high frequency output was controlled well in the white noise environment compared with NAL-NL1. ADRO's multi-channel processing has a clear advantage in this environment. In Figure 5-13 B and D, the neurograms were very similar between ADRO and NAL-NL1, with NAL-NL1's having slight more response to the sentence. The STMI value were also very similar with ADRO being slightly higher.

If the two neurograms were observed closely, ADRO's neurogram has less response in the speechless sections and thus made the speech response slightly more focused and distinct. This may be the reason it has a slightly higher STMI value. Recall that ADRO acts like a linear amplifier as long as the output is in the comfort zone. Letting it reach a stable state by inputting the same sentence several times for more than 10 seconds would likely let ADRO set all gains so that the output would be in the comfort zone; in other words, the amplifier would be operating linearly within the final sentence, with the gains adjusted appropriately for this input. Linear amplification keeps a higher fidelity sound and therefore gets a boost in STMI value and intelligibility.

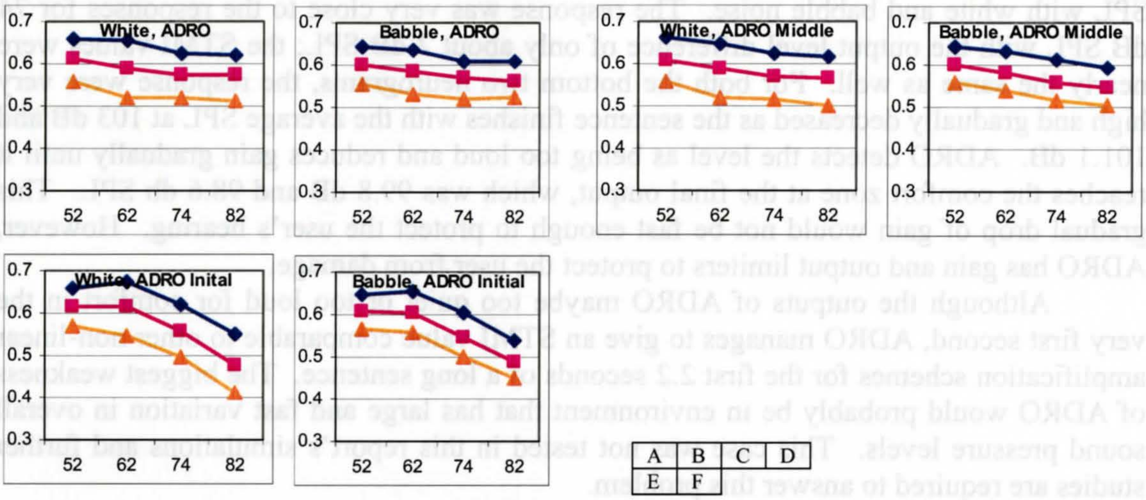


Figure 5-12: STMI values for ADRO in A) white noise and B) babble noise environment. C and D are STMI values while ADRO was adapting. E and F are STMI values when the first repetition of the sentence was introduced. (Noise units are in dB SNR)

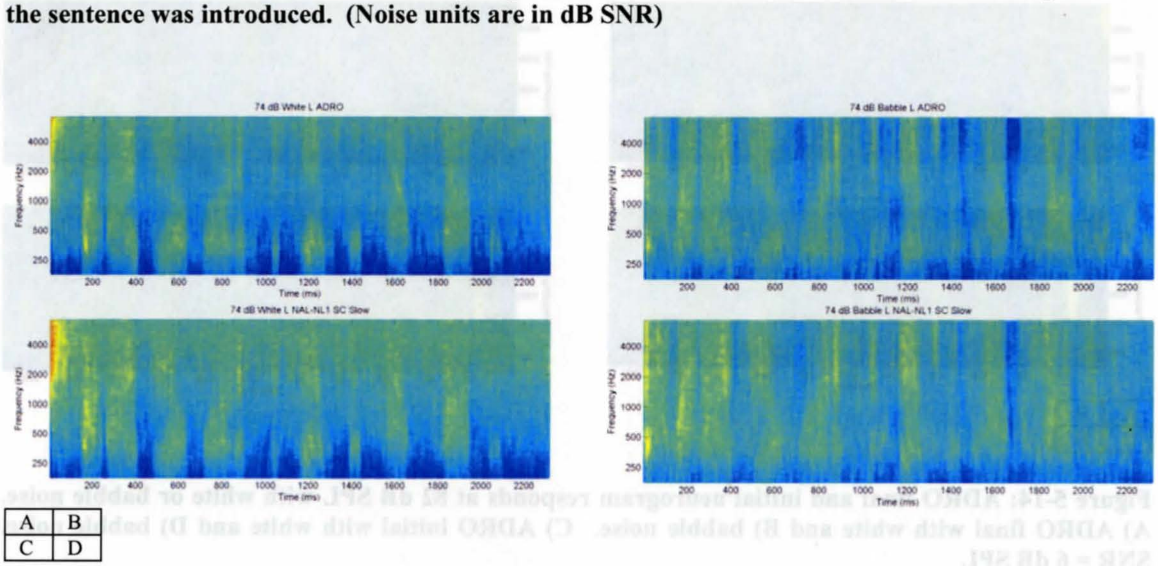


Figure 5-13: ADRO and NAL-NL1's neurogram responds at 74 dB SPL with white or babble noise. A) ADRO with white and B) babble noise. C) NL1 with white and D) babble noise. SNR = 6 dB SPL

Figure 5-12 C to F are ADRO's STMI values in its adapting stages. "ADRO initial" stands for the first sentence of the repetitions sent to ADRO's input, and "ADRO middle" stands for the middle repetition that was between the first and the final repetition that were sent to ADRO's input. Since the test sentence was about 2.2 seconds long, the middle sentence was the third repetition that plays between the 4.4 to 6.6 seconds mark. The STMI result of "ADRO middle" was almost the same as the final and the same was true for the neurogram. Therefore ADRO can nearly reach its ideal output within 5 seconds for this input and initial conditions. For "ADRO initial", the adaptation process could be observed in the neurogram. Figure 5-14 shows the comparison of "ADRO final" and "ADRO initial" neurograms; the top two neurograms were responses at 82 dB SPL with white and babble noise. The response was very close to the responses for 74 dB SPL with the output level difference of only about 2 dB SPL; the STMI values were nearly the same as well. For both the bottom two neurograms, the response were very high and gradually decreased as the sentence finishes with the average SPL at 103 dB and 101.1 dB. ADRO detects the level as being too loud and reduces gain gradually until it reaches the comfort zone at the final output, which was 99.8 dB and 98.6 db SPL. This gradual drop of gain would not be fast enough to protect the user's hearing. However, ADRO has gain and output limiters to protect the user from damage.

Although the outputs of ADRO maybe too quiet or too loud for comfort in the very first second, ADRO manages to give an STMI value comparable to other non-linear amplification schemes for the first 2.2 seconds of a long sentence. The biggest weakness of ADRO would probably be in environment that has large and fast variation in overall sound pressure levels. This case was not tested in this report's simulations and further studies are required to answer this problem.

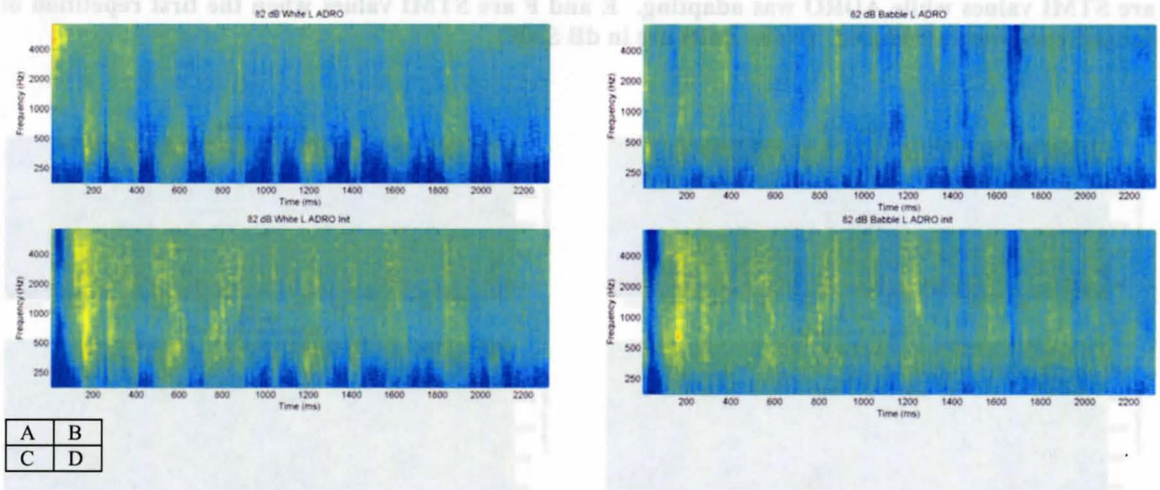


Figure 5-14: ADRO final and initial neurogram responds at 82 dB SPL with white or babble noise. A) ADRO final with white and B) babble noise. C) ADRO initial with white and D) babble noise. SNR = 6 dB SPL

Spectral flattening did not seem to have a noticeable detrimental effect on intelligibility in the test case for ADRO, even though it was running with 32 different channels; this was probably because ADRO operates linearly most of the time, and it does not have a predefined gain target for given a input level and frequency. It acts slowly compared to other schemes such that it would not change level variations within a phoneme. The gain differences between channels do change over time, however, the input speech spectrum changes enough over ADRO's adaptation time so that spectral flattening is minimized. Also, the benefits of increased audibility in each frequency probably outweighed the detrimental effects.

6 Conclusion

The general trend of all the hearing aid amplification schemes tested in this study was to provide very good gain to vastly increase intelligibility in noiseless situations. However, when noise was added, the different amplification strategies start to show different results. Most of the hearing aid algorithms tested in this study worked very well for low level inputs even in noisy situations. However, they all tend to decrease in effectiveness as the input level increases. The decrease in predicted intelligibility was large enough at high level inputs so that even taking the aid off completely could be better for intelligibility. This problem can probably be solved by having a manual volume control on the hearing aid itself. However, an increasing number of non-linear hearing aids in the market are being built without manual volume controls [Dillon 2001]. If the user does not mind changing the volume in loud situations, the user can simply get a linear aid because the linear aids work very well for low levels. In theory, non-linear aids would do even better than linear-aids in very low situations (30–40 dB SPL). This maybe true for some of the schemes tested in this study like NAL-NL1 and ADRO, because the slope of the STMI curve increases towards lower input levels, whereas linear aids and DSL m[i/o] have decreasing curves.

Overall, ADRO compares very well with other non-linear aids in this study with the flattest STMI curve throughout the input SPL range. However, more simulations can be made to test out the effects on intelligibility produced by fast varying sound pressure levels to slow acting ADRO and other compressors with slower attack and release times. Since the current study tends to focus more on speech and noise levels that had a stable SPL, using a full sentence neurogram that tests the average STMI was convenient to compare different schemes' effectiveness. However, if SPLs were to change rapidly, STMI values will probably change quickly as well. Therefore, perhaps running STMI simulations that trace the intelligibility during a sentence would be a better evaluator for fast varying SPL sentences. Also, such setup might better pinpoint where a particular aid does better than another.

7 Appendix

7.1 Main MATLAB Scripts

```
function [STMI] = sim_ag (filename)
%  function sim_ag (filename)
%  Main script for hearing aids simulation and stmi calculations
%  e.g. inputs
%  filename = 'SX255DR2MRHL0062BAL09NSC_5_20';
%  filename = 'SX255DR2MRHL0065';

% addpath(genpath(cd));
addpath(genpath('/home/leungb/SIM'));
disp(' sim_ag - PAHCS v1.0');

tic;
if length(filename) > 16; per = filename(22:24); else per = 'xx'; end;
% Check if it's for ADRO simulation
if strcmp(per(1:2),'AA')
%     load (['C:\WORK\DATA\ADRO\' filename(1:21) 'AAA']);
    load (['/home/leungb/DATA/ADRO/' filename(1:21) 'AAA']);
    if strcmp(per(3),'A')
        TestSignal = adrofin;   % Adapted output (last sentence)
    elseif strcmp(per(3),'I')
        TestSignal = adroinit;  % intial output (fist sentence)
    elseif strcmp(per(3),'M')
        TestSignal = adromid;   % intermediate output (middle sentence)
    else
        error ('Unknow ADRO part ... Set prescription as: AAA/AAI/AAM.');
    end
else % Not ADRO, then generate signal from TIMIT and hearing aids
    [TestSignal fs finfo] = gen_signal (filename);
    % figure;plot(TestSignal);title('Real ear input Signal');
end
sigdb = 20*log10(sqrt(sumsqr(TestSignal)/length(TestSignal))/20e-6);
Xk = abs(fft(TestSignal));
sbsig = real(ifft(Xk.*exp(j*2*pi*rand(size(Xk)))));
sbase = 10^(sigdb/20).*norm2db(sbsig);

toc % TOCK - shows run time
%************************************************
indx = finfo.indx; % hearing loss index
len   = 16.0e-3;    % length of window/frame
shft  = 0.0;
overlap = 100/50;  % 50% overlap
nrep   = 50; % number of stimulus repetitions (e.g., 100);
BF     = 440 * 2.^ (((0:127)-31)/24);
% For cortical model
rv = 2.^(1:0.5:5);  % rate temporal
sv = 2.^(-2:0.5:3); % scale spectral
%************************************************
% Clean/Noisy audgram
% First set using catmodel
if indx==0 % (normal hearing)
    if length(filename) == 16 % Clean Signal
```

```
        disp('Cleanaudgram...');
        cleanaudgram = speech_base_normal ...
            (TestSignal,fs,BF,nrep,len,shft,overlap,rv,sv);
    else % normal
        disp('Noisyaudgram...');
        noisyaudgram = speech_base_normal ...
            (TestSignal,fs,BF,nrep,len,shft,overlap,rv,sv);
    end
else % impaired
    disp('Noisyaudgram...');
    noisyaudgram = speech_base_impaired ...
        (TestSignal,fs,BF,nrep,len,shft,overlap,rv,sv,indx);
end
pause(1);

% Second set using catmodel
% base neurogram / cleanaudgram saving
disp('____________________________________________________________');
if indx==0 % (normal hearing)
    if length(filename) == 16 % cleanaudgram
        % Creating template, and saving file
        disp('  naudgram .........     ');toc %%%%%%%%%%%%%%%%%%%%%%%%%%%
        naudgram = speech_base_normal(sbase,fs,BF,nrep,len,shft,overlap,rv,sv);
        disp('  saving file ...........     ');toc
        disp('  cleansp ...');
        cleansp = abs(ngram2cortex(cleanaudgram,len/overlap,rv,sv));
        disp('  cleanspbase ...');
        cleanspbase = abs(ngram2cortex(naudgram,len/overlap,rv,sv));

        T  = max(cleansp - cleanspbase,0);
        clear cleansp cleanspbase;
        TT=0;
        for pp = 1:size(T,3)
            Ttmp = squeeze(T(:,:,pp,:));
            TT   = TT + sumsqr(Ttmp(:));
        end
        CleanSignal = TestSignal;
        save (['/home/leungb/DATA/CAG/' filename '_CAG'], ...
                       'cleanaudgram','naudgram','T','TT','CleanSignal');
        time(toc)
        return;
    else % base ngram for normal hearing
        disp('  nnaudgram .........     ');toc
        nnaudgram    = speech_base_normal ...
            (sbase,fs,BF,nrep,len,shft,overlap,rv,sv);
    end
else % base ngram for impaired hearing
    disp('  nnaudgram .........     ');toc
    nnaudgram    = speech_base_impaired ...
        (sbase,fs,BF,nrep,len,shft,overlap,rv,sv,indx);
end

% compensate for neurogram size differences
% load (['C:\WORK\DATA\CAG\' filename(1:13) '065_CAG'],'cleanaudgram');
load (['/home/leungb/DATA/CAG/' filename(1:13) '065_CAG'],'cleanaudgram');
cleansize = size(cleanaudgram,1);
noisysize = size(noisyaudgram,1);
if noisysize > cleansize
    disp(' Resizing audgrams ... ');
```

```
    noisyaudgram = noisyaudgram(1:cleansize,:);
    nnaudgram = nnaudgram(1:cleansize,:);
elseif noisysize < cleansize
    error(' noisysize < cleansize ! should be equal or greater.');
end
disp('  noisyp ...');
noisysp = abs(ngram2cortex(noisyaudgram,len/overlap,rv,sv));
disp('  noisyspbase ...');
noisyspbase = abs(ngram2cortex(nnaudgram,len/overlap,rv,sv));
disp('  N .........  ');
N  = max(noisysp - noisyspbase,0);  clear noisysp noisyspbase;
% load (['C:\WORK\DATA\CAG\' filename(1:13) '065_CAG'],'T','TT');
load (['/home/leungb/DATA/CAG/' filename(1:13) '065_CAG'],'T','TT');


NN=0;
for pp = 1:size(N,3)
    Ntmp = max(squeeze(T(:,:,pp,:)-N(:,:,pp,:)),0);
    NN   = NN + sumsqr(Ntmp(:));
end
disp('  STMI .........    ');
STMI = sqrt(1-NN/TT);
disp('FIN !!!              ');
disp('  saving file ...........      ');toc
if strcmp(per,'AAA')
    save (['/home/leungb/DATA/ADRO/' filename],'x','fs','finfo',...
          'ADROoutput','adroinit','adromid','adrofin',...
          'noisyaudgram','nnaudgram','STMI','NN','TestSignal');
else
    save (['/home/leungb/DATA/' filename], ...
         'noisyaudgram','nnaudgram','STMI','NN','TestSignal');
end
disp('  FIN !!     ');
time(toc); % Shows elapse time in hr : min : sec
```

```
function [data fs finfo] = gen_signal (filename)
% dr = 'DR2'; % Dialact Region
% id = 'MRHL0'; % file ID
% spl = 65;             % Sound presentation level [52 62 74 82]
% noisetype = 'WG';     % noise type [WG/BA]
% noiselvl  = 'L';      % noise level [H/M/L]
% indx = 9;   % hearing loss index [1-10]  (7,5,9,10,8)(L,M,H,P,Steep)
% prescription = 'Nxx'; % prescription: xxx, Nxx, Dxx, NSC, NMC, DSC, DMC
%                       N - NAL-RP / NAL-NL1
%                       D - DSL / DSL-[i/o]
%                       S / M - Single / Multi, C - Channel compressor
% %%%%%%%%%%  Sample input parameters for compressor %%%%%%%%%%%%%%%
% tatt = 0.120;   %(sec) Attack Time [1-10ms] (for fast acting)
% trel = 0.500;   %(sec) Release Time [10-50ms] (for fast acting)
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Note for noise type: WG -> White
%                      BA -> Babble
%*****************************************************************
% filename examples:
```

```
% filename = 'SX255DR2MRHL0065';
% filename = 'SX255DR2MRHL0060BAL09Dxx_2_15';

% filename input handles
sname = filename(1:5);
fname = filename(1:13);
p = ['/home/leungb/SIM/TIMIT/' sname '/' fname '.WAV'];
[data,fs] = readsph(p);

% organize input string
slength = length(data);
n = length(filename);
fname = filename(1 :5);
dr    = filename(6 :8);
id    = filename(9 :13);
spl   = filename(14:16);    spl = str2double(spl);
if n > 16
    % Testsignal
    noisetype = filename(17:18);
    noiselvl  = filename(19);
    indx      = filename(20:21);  indx = str2double(indx);
    prescription = filename(22:24);
    if length(filename) > 24
        t = [0 0];
        s = 0;
        for i = 1:n
            if filename(i) == '_'
                s = s+1;
                t(s) = i;
            end
        end
        ATT = filename((t(1)+1):(t(2)-1)); ATT = str2double(ATT);
        REL = filename((t(2)+1):end);      REL = str2double(REL);
    end
else
    % Cleansignal (use 65 dB SPL)
    spl = 65;
    noisetype    = '';
    noiselvl     = '';
    indx         = 0;
    prescription = '';
    ATT          = '';
    REL          = '';
    tatt = 0;
    trel = 0;
end

% load hearing loss profiles
load LossProfile_Studebaker1999;
if indx==1; HL=DBHL(1,:); end;
if indx==2; HL=DBHL(2,:); end;
if indx==3; HL=DBHL(3,:); end;
if indx==4; HL=DBHL(4,:); end;
if indx==5; HL=DBHL(5,:); end;
if indx==6; HL=DBHL(6,:); end;
load LossProfile_Shanks2002;
if indx==7; HL=DBHL(1,:); end;
if indx==8; HL=DBHL(2,:); end;
if indx==9; HL=DBHL(3,:); end;
```

```
if indx==10;HL=DBHL(4,:); end;
clear DBHL CIHC COHC;

data = set_spl(data,spl);

if n > 16   % If test signal, then
    t = 0:1/fs:((length(data)-1)/fs);
%     figure;
%     subplot(311);plot(t,data); %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     title(['signal,  ' num2str(spl) ' dB']);pause(0.1);

    % NOISE  ***********************
    if strcmp(noisetype, 'WG')         % White Gaussian
        noise = (randn(1,length(data)));
    elseif strcmp(noisetype, 'BA')     % Babble
        noise = wavread('babble01.wav');
        noise = noise(1:length(data))';
    elseif strcmp(noisetype, 'EN')     % Environmental
        noise = [];
        disp('Need Environmental noise');
    elseif strcmp(noisetype, 'SC')     % Single Competing speaker
        noise = wavread('singlecompeting.wav');
        noise = noise(1:length(data))';
    else                         % No noise
        noise = [];
        noisetype = 'xx';
        noiselvl = 'x';
    end
    if isempty(noise) == 0       % if there are noise, then set SNRs
        if strcmp(noiselvl, 'H')       % High level noise
            noise = set_spl(noise,spl);
        elseif strcmp(noiselvl, 'M')  % Medium level noise
            noise = set_spl(noise,spl-3);
        elseif strcmp(noiselvl, 'L')  % Low level noise
            noise = set_spl(noise,spl-6);
        else
            noiselvl = 'L';      % if weird input, then set L level noise
            noise = set_spl(noise,spl-3);
        end
        data = data + noise';       % Adding noise to signal
    end

    if indx ==0 % normal hearing
        prescription = '';
        tatt = 0; trel = 0;
    else
        HLindex = sprintf('HL%02d',indx);
        %     tatt = ATT/1000;
        %     trel = REL/1000;

        if strcmp(prescription(2:3),'xx')
            tatt = 0; trel = 0;
        else
            tatt = ATT/1000;
            trel = REL/1000;
        end
    end
```

```
    % repeat for longer attack/release times
    % Comment following if want to see slow compressors adapt
    if (tatt >= 0.1) || (trel >= 0.4)
        disp('repmat-ing');
        data = repmat(data,3,1);
    end

    % Prescription *****************************************************
    if strcmp(prescription,'Nxx') % NAL-RP
        disp('Using NAL-RP.  ');
        data = nalrp(data,fs,CF,HL,109);
        data = hrtffiltfft(data,fs,112);
    elseif strcmp(prescription,'Dxx') % DSL
        disp('Using DSL.  ');
        data = dsl(data,fs,CF,HL,112);
    elseif strcmp(prescription,'NSC') % NAL-NL1 single channel
        disp('Using NAL-NL1 single channel.  ');
        pre = [HLindex '_NAL_REAG'];
        data = CS02(data,fs,tatt,trel,pre);
    elseif strcmp(prescription,'NMC') % NAL-NL1 multi channel
        disp('Using NAL-NL1 multi channel.  ');
        pre = [HLindex '_NAL_REAR'];
        data = CM01(data,fs,tatt,trel,pre);
    elseif strcmp(prescription,'DSC') % DSL-[i/o] single channel
        disp('Using DSL-[i/o] single channel.  ');
        pre = [HLindex '_DSL_REAG'];
        data = CS02(data,fs,tatt,trel,pre);
    elseif strcmp(prescription,'DMC') % DSL-[i/o] multi channel
        disp('Using DSL-[i/o] multi channel.  ');
        pre = [HLindex '_DSL_REAR'];
        data = CM01(data,fs,tatt,trel,pre);
    elseif strcmp(prescription,'AAA') || strcmp(prescription,'AAB') ...
            || strcmp(prescription,'AAC')
        % Use alternate code for ADRO
        error ('  Use alternate program for ADRO');
    else %%%%%%%%%%%%%%%%%%%%%%%%%%%%%% No prescription
        disp('No prescription.  ');
        prescription = 'xxx';
    end
    % adding compressor specs on filename
    if strcmp(prescription,'xxx') || strcmp(prescription,'Nxx')...
            || strcmp(prescription,'Dxx')
        ATT = ''; REL = ''; % add nothing if com not used
        tatt = 0; trel = 0;
    else
        % shown in milliseconds
        ATT = ['_' num2str(ATT)];
        REL = ['_' num2str(REL)];
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
else % For clean signal
    % no modification to sound, just changing to real ear level
    data = hrtffiltfft(data,fs);
end

% keep the last repeat
if length(data) > slength
    data = data ((end-slength+1):end);
end
```

```
pindx = sprintf('%02d',indx); % store as string

% generate printname
printname = sprintf('%-5s%s%s%03d%s%s%s%s%s%s%s%s',...
    fname,dr,id,spl,noisetype,noiselvl,pindx,prescription,ATT,REL);

disp(printname);

% saving file info as structure
finfo.fname = fname;
finfo.dr = dr;
finfo.id = id;
finfo.spl = spl;
finfo.noisetype = noisetype;
finfo.noiselvl = noiselvl;
finfo.indx = indx;
finfo.prescription = prescription;
finfo.tatt = tatt;
finfo.trel = trel;
```

8 Bibliography

Blamey PJ. *Adaptive dynamic range optimization (ADRO): A digital amplification strategy for cochlear implants and hearing aids.* Trends in Amplification 9(2):77–98, 2005

Blamey PJ. *Adaptive Dynamic Range Optimization for Hearing Aids.* White Paper: presented at WESPAC, June 26–28, 2006

Byrne D, Dillon H, Ching T, Katsch R, Keidser G. *NAL-NL1 Procedure for Fitting Nonlinear Hearing Aids: Characteristics and Comparisons with Other Procedures.* J Am Acad Audiol. 12: 37-51, 2001

Child Amplification Laboratory. *An update to DSL 5: WDRC targets for severe to profound hearing loss.* University of Western Ontario. Sept 10, 2007

Dickson B, Steele BR. *Method and device for low delay processing.* International patent application PCT/AU2006/001778. 2006

Dillon H. *Hearing Aids.* New York: Thieme Medical publishers, 2001

Elhilali M, Chi T, Shamma A. *A spectro-temporal modulation index (STMI) for assessment of speech intelligibility.* Speech Comm. 41:331–348, 2003

Goldstein EB. *Sensation and Perception 6th ed.* Pacific Grove: Wadsworth Group, 2002

Guyton AC, Hall JE. *Textbook of Medical Physiology 11th ed.* Saunders, 2005

Lorenzi C, Gilbert G, Carn H, Garnier S, Moore BCJ. *Speech perception problems of the hearing impaired reflect inability to use temporal fine structure.* PNAS 103:18866–18869, 2006

Robles L, Ruggero MA. *Mechanics of the Mammalian Cochlea.* Physiological Reviews 81(3):1305-1343, 2001

Sachs MB, Bruce IC, Roger LM, Young ED. *Biological Basis of Hearing-Aid Design.* Annals of Biomedical Engineering 30:157-168, 2002

Scollie S, Seewald R, Cornelisse L, Moodie S, Bagatto M, Laurnagaray D, Beaulac S, Pumford J. *The Desired Sensation Level Multistage Input/Output Algorithm.* Trends in Amplification. 9(4):159-197

Warren RM. *Auditory Perception: A New Analysis and Synthesis*. Cambridge University Press, 1999

Wiener FM, Ross DA. *The Pressure Distribution in the Auditory Canal in a Progessive Sound Field.* J. Acoust. Soc. Am. 18: 401-408, 1946

Zilany MSA, Bruce IC. *Modeling auditory-nerve responses for high sound pressure levels in the normal and impaired auditory periphery.* J. Acoust. Soc. Am. 120(3):1446–1466, 2006

Zilany MSA, Bruce IC. *Predictions of speech intelligibility with a model of the normal and impaired auditory-periphery.* IEEE EMBS Con. Neural Eng. SaA1.2:481–485, May 2–5, 2007