AN EDUCATIONAL TOOL FOR HEARING AID COMPRESSION FITTING VIA A WEB-BASED ADJUSTED SMARTPHONE APP

Nasim Alamdari[1], Edward Lobarinas[2], and Nasser Kehtarnavaz[1]

[1]Department of Electrical and Computer Engineering
[2]Callier Center for Communication Disorders
University of Texas at Dallas, Richardson, TX

ABSTRACT

This paper presents an educational tool to learn about how hearing aid compression fitting is prescribed from a signal processing perspective. An interactive web-based program has been developed based on the widely used DSL-v5 fitting rationale. This program can be accessed and used from any internet browser to generate the parameters of compression curves that correspond to the nine frequency bands used in DSL-v5. These parameters are then transferred to a smartphone in the form of a datafile to be used by a compression app, which operates as a virtual hearing aid, running in real-time on both iOS and Android smartphones. This educational tool was found to be very easy-to-use by signal processing engineers as no programming knowledge is needed for adjusting gain across frequency bands and subsequently running the corresponding compression curves in the smartphone app to appropriately compress input sound signals.

***Index Terms*—** Educational tool for hearing aid compression fitting, web-based hearing aid compression, compression smartphone app

1. INTRODUCTION

According to the World Health Organization (WHO), more than 450 million people around the world have hearing loss and it is estimated that by 2050 more than 900 million people will have hearing loss [1]. One way that hearing loss is managed is via the use of hearing aid devices. A hearing aid device is either worn in or behind an ear and consists of three major components: a microphone, a signal processor, and a speaker. The signal processor amplifies acoustic input from the microphone in order to compensate for a person's hearing loss. The amplified sounds are then sent to the ear through the speaker. In modern hearing aids, the signal processor runs a number of signal processing algorithms such as compression, adaptive feedback cancellation, and noise reduction.

The main signal processing algorithm running on the signal processor of a modern hearing aid device to cope with hearing loss is compression. The human auditory system is capable of perceiving a wide dynamic range of sound intensity from soft to loud. For a person with sensorineural hearing loss, the dynamic range can be significantly reduced. Importantly, the relative intensity difference between barely audible and uncomfortably loud becomes smaller because perception for very loud sound remain relatively similar across a wider range of hearing loss and for individuals with normal hearing. Thus, in order for sound to be audible, the acoustic information for soft, moderate, and loud sounds must occupy a smaller working space. In order to address this problem, a method of squeezing information (compression) into this space is used.

The process of compression involves mapping the wide dynamic range of normal hearing range into a smaller dynamic range commensurate with the degree of hearing loss [2, 3]. Compression can be achieved via input/output compression curves or functions as illustrated in Fig. 1. As can be seen from this curve, more gain is provided for inputs that are less than 75 dB SPL (Sound Pressure Level) than for inputs that exceed 75 dB SPL. The gain for inputs exceeding 75 dB SPL is compressed relative to the linear gain provided for lower level inputs. Considering that hearing loss differs across frequency bands (typically 9 frequency bands are used), a series of compression curves are often needed to accommodate differences in residual dynamic range at each of these frequencies. The fitting of a hearing aid device is normally carried out by adjusting gain across different frequency bands for soft, intermediate, and loud input sound levels. Fig. 2 illustrates a frequency response curve with higher gain for frequencies in the 3-4 kHz range. Such a curve could be used for an individual with noise-induced hearing loss that typically creates a notch of poorer hearing in the 4 kHz region.

The objective of this paper is to provide an easy-to-use educational tool for signal processing engineers to learn about how hearing aid amplification fitting are implemented using signal processing compression curves. The widely used prescriptive fitting of DSL-v5 (Desired Sensation Level – version 5) by Hand [4] is considered in this paper. This educational tool consists of an interactive web-based compression fitting program and a signal processing compression algorithm running on the ARM processor of smartphones (both iOS and Android) as an app. The

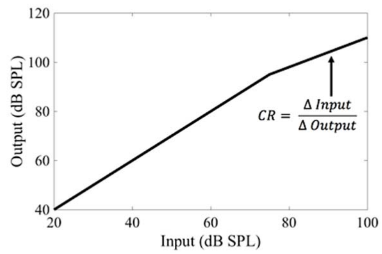


Fig. 1 - Sample input/output compression function.

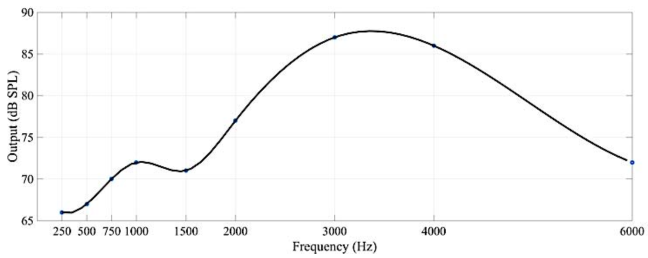


Fig. 2 - Sample frequency response curve for an input sound level of 65 dB SPL.

interaction between the web-based fitting program and the smartphone app is carried out via a datafile. This educational tool is portable, thus enabling its wide spread utilization for studying the effect of changing compression gains in realistic audio environments.

The remainder of this paper is organized as follows. Section 2 describes both the web-based compression and the smartphone app components of the developed educational tool. Sample compression fittings are then presented in section 3 followed by the conclusion in section 4.

2. HEARING AID COMPRESSION EDUCATIONAL TOOL

Fig. 3 shows a block diagram of the components of the developed educational tool. The DSL-v5 compression fitting is carried out interactively on a browser. The languages used to write this interactive web-based program include HTML (Hypertext Markup Language), CSS (Cascading Style Sheets), and JavaScript. This web-based fitting is freely accessible and can be run on any browser via this link: http://www.utdallas.edu/~kehtar/WebBasedFitting.html. No knowledge of the programming languages used to create the program is needed in order to use it.

Fig. 4 shows the graphical-user-interface (GUI) of the web-based compression fitting program. In this figure, the entries of the first table from the top correspond to the age of a subject (more than 3 years old, 4-5 years old, 6 years old, or more than 6 years old), the type of hearing aid (CIC (Completely in Canal), ITC (In-The-Canal), ITE (In-The-Ear, or BTE Behind-The-Ear)), and the ear coupling of hearing aid (Earmolds or Eartips). The second table from the top incorporates the subject's audiometric thresholds for whom compression fitting is done. Audiometric thresholds

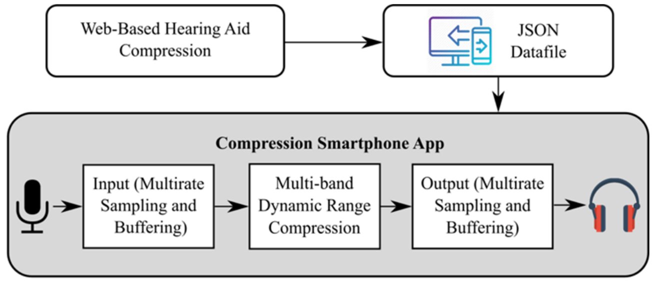


Fig. 3 – Components of the hearing aid compression educational tool.

DSL-v5 by Hand Hearing Aid Fitting

Age (year)	>3
Hearing Aid Type	BTE
Ear Coupling Type	Earmolds

(a) Subject and hearing aid information.

Online Audiogram Measurements

Frequency Bands (Hz)	125-250	250-500	500-750	750-1K	1-1.5K	1.5-2K	2-3K	3-4K	4-6K
Audiogram (dB SPL)	60	60	65	70	75	80	85	90	95

(b) Audiogram measurements from an online test.

Soft Speech (55 dB)

Frequency Bands (Hz)	125-250	250-500	500-750	750-1K	1-1.5K	1.5-2K	2-3K	3-4K	4-6K
Target SPL Output	84	82	81	83	84	90	93	95	94
Subtract Input SPL	45	47	42	40	38	34	32	31	30
Subtract RECD	3	5	5	6	8	6	2	3	8
Subtract Mic Effect	1	1	1	1	2	3	4	2	1
Target Gain	35	29	33	36	34	43	53	58	54

Moderate Speech (65 dB)

Frequency Bands (Hz)	125-250	250-500	500-750	750-1K	1-1.5K	1.5-2K	2-3K	3-4K	4-6K
Target SPL Output	89	88	88	91	93	100	103	105	104
Subtract Input SPL	55	57	52	50	48	44	42	41	40
Subtract RECD	3	5	5	6	8	6	2	3	8
Subtract Mic Effect	1	1	1	1	2	3	4	2	1
Target Gain	30	25	30	34	35	47	55	59	55

Loud Speech (75 dB)

Frequency Bands (Hz)	125-250	250-500	500-750	750-1K	1-1.5K	1.5-2K	2-3K	3-4K	4-6K
Target SPL Output	90	92	94	97	100	106	111	113	109
Subtract Input SPL	57	65	66	64	65	61	57	56	50
Subtract RECD	3	5	5	6	8	6	2	3	8
Subtract Mic Effect	1	1	1	1	2	3	4	2	1
Target Gain	29	21	22	26	25	36	48	52	50

Max Output (90 dB)

Frequency Bands (Hz)	125-250	250-500	500-750	750-1K	1-1.5K	1.5-2K	2-3K	3-4K	4-6K
Target SPL Max Output	106	107	110	113	114	118	121	122	121
Subtract RECD	3	5	5	6	8	6	2	3	8
Target OSPL-90	103	102	105	107	106	112	119	119	113

Update Tables

(c) Gains set according to the audiogram and DSL-v5 prescriptive settings for 9 frequency bands.

Fig. 4 - Graphical-user-interface (GUI) of the web-based compression fitting program.

(called audiogram) correspond to the softest level of sound a person can hear at different frequencies. The online utility in [5] is used by the program to obtain these measurements.

The next four tables indicate gain across 9 frequency bands for speech inputs at soft, moderate, loud, and maximum levels. The entry Target SPL Output in these tables is automatically determined based on the audiogram measurements and the pre-specified values in the DSL-v5 prescriptive tables. The other pre-specified entries of Input SPL, RECD (Real-Ear to Coupler Difference) and Mic Effect in DSL-v5 are automatically subtracted to generate specific gain in the frequency bands for soft (55dB), moderate (65dB), loud (75dB), and maximum output (90dB) levels.

Gain from the DSL-v5 tables is then converted into compression curves corresponding to the 9 frequency bands. Each compression curve is expressed in terms of the following four parameters [6] (see Figs. 1 and 6): (i) Compression Threshold (CT) - This parameter indicates the point after which the compression is applied. (ii) Compression Ratio (CR) - This parameter indicates the amount of compression. (iii) Attack Time (AT) – This parameter indicates the time it takes for the compression module to respond when the signal level changes from a low to a high value. (iv) Release Time - This parameter indicates the time it takes for the compression module to respond when the signal level changes from a high to a low value. The default values of the attack and release times are set to 5ms and 100ms, respectively. The compression ratio of each frequency band for the compression thresholds of 55 dB and 75 dB SPL is computed as follows:

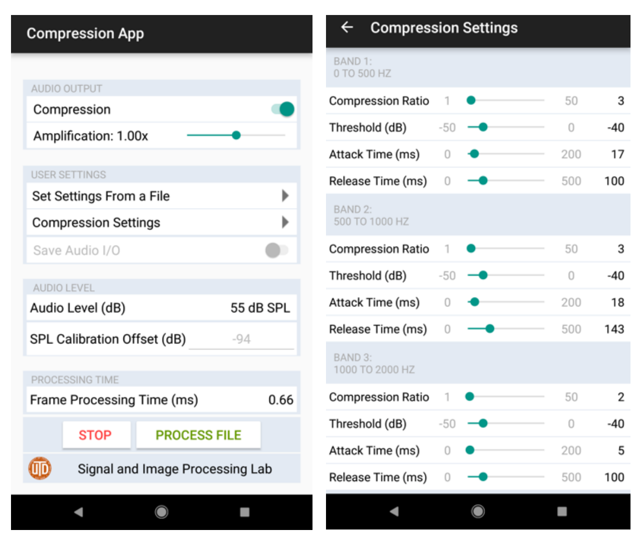
$$CR = \frac{\Delta Input}{\Delta Output} = \frac{75-55}{(75+Gain_{at\ 75\ dB})-(55+Gain_{at\ 55\ dB})}$$

The button at the end of the web-based program generates a JSON (JavaScript Object Notation) datafile which includes the parameters of the compression curves in a readable form. JSON is a popular web programming approach for transmitting data to other platforms. The created JSON datafile is then transferred to the smartphone and placed inside the compression app folder to be used by it.

Another component of the developed educational tool is a smartphone app for both Android and iOS smartphones. This app performs multiband wide dynamic range compression (WDRC) according to the compression curves determined by the web-based compression program. The app consists of an input module, a multi-band dynamic range compression module, and an output module. The compression module in the app uses the multiband WDRC in [7] in which CTs are defined in the range of [-50, 0] dB. Positive CTs of the JSON file are redefined based on this range via this equation $CT_{app} = CT_{JSON} - 95dB$.

Multi-rating is used in the input/output modules to allow achieving the lowest audio latency on smartphones. For iOS smartphones, the audio i/o needs to run at 48kHz and the i/o buffer size needs to be kept at 64 samples or 64/48000=1.33ms in order to achieve the lowest audio latency. For Android smartphones (for example Google Pixel), the audio i/o needs to run at 48kHz and the i/o buffer size needs to be kept at 192 samples or 192/48000 = 4ms in order to have the lowest audio latency. For the app to run in real-time, the compression module runs in frame-based manner at 16kHz with a 25ms processing frame size and with 50% overlap, (i.e. a frame gets processed every 12.5ms). To synchronize the audio i/o and the compression modules, circular buffers are utilized. An input circular buffer collects input samples from the audio i/o buffer until the overlapped frame size of 12.5ms (600 samples) is reached. It is then down-sampled, decimated by a factor of 3 and fed into the compression module. After a frame is passed through the compression module, it is up-sampled and interpolated before being placed into an output circular buffer, which then outputs the audio at the rate of 64 samples (iOS) or 192 samples (Android) at 48kHz. This process maintains the lowest audio latency available by the smartphone i/o hardware. Interested readers are referred to [8-12] for more details of the modules in the compression smartphone app.

Fig. 5 shows the GUI of the Android version of the app. The compression settings are read from the web-based generated JSON datafile. To gain computational efficiency, the nine frequency bands are mapped into the following five frequency bands [0Hz-500Hz], [500Hz-1000Hz], [1000Hz-2000Hz], [2000Hz-4000Hz], and above 4000Hz. The compressed output can be heard via a hardwired earphone or wirelessly (via Bluetooth) on the smartphone. The web-based compression fitting program also incorporates the DSL-v5 by Hand document. A video clip demonstrating various components of this educational tool is provided at this link: http://www.utdallas.edu/~kehtar/EducationalCompressionFitting.mp4. This video clip shows the entire fitting process



(a) (b)

Fig. 5 - GUI of compression smartphone app (Android version): (a) main page, (b) compression parameters.

Table 1 - Sample gains for three degrees of hearing loss in 9 frequency bands for a >6-year subject, a BTE hearing aid, and an earmold ear coupling.

Degree of Hearing Loss	Audiogram (dB SPL)	Gains from DSL-v5 Tables (dB SPL)
Normal-Mild	{10, 15, 20, 25, 25, 20, 25, 25, 15}	*Soft Speech*: {4, 5, 10, 13, 11, 13, 25, 27, 9} *Moderate Speech*: {2, 4, 9, 11, 11, 16, 25, 25, 9} *Loud Speech*: {1, 2, 5, 7, 6, 12, 22, 21, 7} *MPO:* {88, 89, 90, 92, 90, 91, 98, 95, 86}
Moderate	{60, 60, 65, 70, 70, 75, 80, 85, 85}	*Soft Speech*: {35, 29, 33, 36, 34, 43, 53, 58, 54} *Moderate Speech*: {30, 25, 30, 34, 35, 47, 55, 59, 55} *Loud Speech*: {29, 21, 22, 26, 25, 36, 48, 52, 50} *MPO:* {103,102,105,107,106, 112, 119, 119, 113}
Severe	{60, 60, 65, 70, 85, 85, 95, 100,110}	*Soft Speech*: {35, 29, 33, 36, 43, 48, 64, 69, 71} *Moderate Speech*: {30, 25, 30, 34, 45, 52, 66, 70, 72} *Loud Speech*: {29, 21, 22, 26, 36, 43, 58, 61, 67} *MPO:* {103,102,105,107,114, 116, 127, 126, 127}

including obtaining audiogram measurements for the 9 frequency bands, computing gains from the DSL-v5 tables, updating the parameters of the compression curves according to gain, and finally generating a JSON datafile containing the compression parameters. It also shows how to transfer this datafile to a smartphone platform and how to run the compression app based on this datafile.

3. SAMPLE COMPRESSION RESULTS

In this section, sample compression fitting results are provided for different degrees of hearing loss. Table 1 shows sample gains corresponding to three degrees of hearing loss: normal/mild, moderate, and severe. The input-output compression curve or function generated by the developed educational tool for the moderate hearing loss row of Table 1 is shown in Fig. 6. This I/O compression function has the following two CTs that can get adjusted: (i) linear gain for sound levels lower than the CT1 (soft sounds), (ii) gain between CT1 and CT2 based on the compression ratio CR, (iii) gain for sound levels higher than CT2, and finally (v) a peak clipping control to ensure the output does not exceed the maximum power output (MPO).

Based on the scale of 1 (very difficult-to-use) through 5 (very easy-to-use), five signal processing engineers were asked to rate this educational tool. All five testers rated the tool as a 5; very easy-to-use. All the engineers commented that their rating was based on the fact that no web programming knowledge and no familiarity with smartphone software tools were needed in order to run the smartphone app using the compression curves derived from the gain settings via the DSL-v5 prescriptive fitting webpage.

The processing time of the developed compression smartphone app for each frame is 0.8ms on the iPhone 8 and 0.9ms on the Google Pixel smartphones. Considering that these times are well below the overlapped frame time of 12.5ms, the app runs in real-time with no frames getting skipped. The CPU and memory utilization for the iOS version of the app, reported by Xcode IDE [13], is 4% and 18MB, and for the Android version of the app, reported by Android Studio IDE [14], is 6% and 66MB, respectively.

4. CONCLUSION

In this paper, an educational tool to learn about hearing aid compression has been introduced. A web-based program has been developed that effectively mimics the DSL-v5 hearing aid fitting steps and prescriptive gain across nine frequency bands for four speech intensity levels. These gain values are converted into nine compression curves or functions whose parameters are stored into a datafile. This datafile is then used by a smartphone app (both iOS and Android) which processes input sound files in real-time based on the DSL-v5 designed compression curves. The developed web-based program can be easily duplicated for other popular hearing aid prescriptive fittings such as NAL-NL2 [15]. The approach presented in this paper provides an effective learning tool and a framework for teaching of and research in hearing aid compression in an easy-to-use and universally accessible manner.

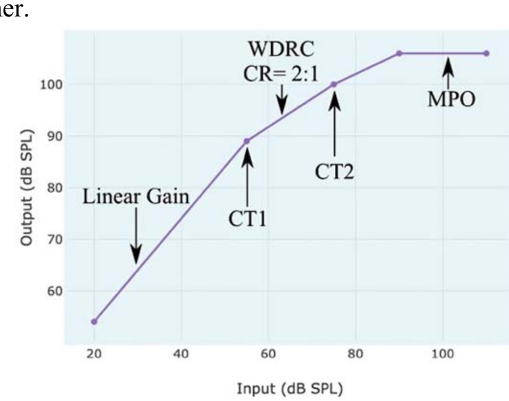


Fig. 6 - Input-output compression curves or functions in the 5th frequency band for the moderate degree of hearing loss listed in Table 1.

5. REFERENCES

[1] World Health Organization, http://www.who.int/mediacentre/factsheets/fs300/en/, 2019.

[2] D. Eddins, *Sandlin's Textbook of Hearing Aid Amplification,* Plural Publishing, 2014.

[3] T. Venema, *Hearing Aid Amplification: Technical and Clinical Considerations*, Singular Publishing Group, 2000.

[4] Western University, DSL-v5 by Hand, https://www.dslio.com/wpcontent/uploads/2014/06/DSL-5-by-Hand.pdf, 2014.

[5] S. Pigeon, https://hearingtest.online/, 2017.

[6] Starkey, Compression Handbook, https://starkeypro.com/pdfs/The_Compression_Handbook.pdf, 2017.

[7] MathWorks, https://www.mathworks.com/help/audio/examples/multiband-dynamic-range-compression.html, 2018.

[8] N. Kehtarnavaz, S. Parris, and A. Sehgal, "Using smartphones as mobile implementation platforms for applied digital signal processing courses," *Proceedings of IEEE Signal Processing and Signal Processing Education Workshop (SP/SPE),* pp. 313-318, 2015.

[9] N. Alamdari, S. Yaraganalu, and N. Kehtarnavaz, "A real-time personalized adaptive noise reduction smartphone app for realistic audio environments," *Proceedings of IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, Philadelphia, PA, Dec 2018.

[10] T. Chowdhury, A. Sehgal, and N. Kehtarnavaz, "Integrating signal processing modules of hearing aids into a real-time smartphone app," *Proceedings of IEEE 40th International Conference on Engineering in Medicine and Biology*, Honolulu, HI, July 2018.

[11] A. Sehgal and N. Kehtarnavaz, "Utilization of two microphones for real-time low-latency audio smartphone apps," *Proceedings of IEEE International Conference on Consumer Electronics*, Las Vegas, NV, Jan 2018.

[12] N. Kehtarnavaz, A. Sehgal, and S. Parris, *Smartphone-Based Real-Time Digital Signal Processing*, *Second Edition*, Morgan and Claypool Publishers, 2018.

[13] Apple, https://developer.apple.com, 2018.

[14] Google, https://developer.android.com, 2018.

[15] G. Keidser, H. Dillon, L. Carter, and A. O'Brien, "NAL-NL2 empirical adjustments," *Trends in Amplification*, vol. 16, pp 211-223, 2012.