PERSONALIZATION OF HEARING AID DSLV5 PRESCRIPTION AMPLIFICATION IN THE FIELD VIA A REAL-TIME SMARTPHONE APP

Aoxin Ni[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 a real-time smartphone app for the purpose of conducting personalization of hearing aid DSLv5 prescription amplification in the field. The developed smartphone app enables hearing healthcare providers and researchers to study the amplification or compression function of hearing aids in realistic audio environments. The personalization approach is based on a previously developed method of maximum likelihood inverse reinforcement learning (MLIRL) which incorporates a user's hearing preferences in paired audio comparisons. The app consists of a real-time online training module and a real-time testing module. In the training module, an optimal personalized set of DSLv5 gains across five frequency bands, commonly used for audiograms, is obtained based on paired audio comparisons which are carried out in an on-the-fly or online manner in real-world audio environments or in the field. In the testing module, the MLIRL derived personalized set of gains can be compared to the standard DSLv5 set of gains in terms of hearing preference. The audio processing steps taken to achieve this implementation as a real-time smartphone app are presented and the results related to its ease-of-use and real-time operation are reported.

Index Terms— smartphone app for hearing aid fitting, personalization of hearing aid DSLv5 amplification or compression prescription, field deployment of maximum likelihood inverse reinforcement learning

1. INTRODUCTION

The conventional hearing aid fitting process consists of obtaining a person's pure tone hearing thresholds or audiogram and applying a prescriptive fitting rationale. In essence, the fitting process involves applying appropriate gain values across a number of frequency bands (often 5, 7, or 9) depending on the input level of sound signals. The target gain values in standard fitting rationales reflect an "average" derived from a group of hearing-impaired individuals with similar hearing loss. However, this approach does not take into consideration any individual preferences of a user or the fact that different individuals encounter different audio environments and that their hearing preferences in those environments may not be the same.

Currently, the two most widely used hearing aid prescriptions are DSLv5 [1] and NAL-NL2 [2]. In this paper, the focus is placed on DSLv5 in part due to its public availability of target gains, standardization approach and separate target gains for adults versus minors. However, the same personalization approach is applicable to NAL-NL2 or any other prescriptive fitting rationales such as the one discussed in [3].

Previous studies, e.g. [4, 5], have shown that nearly half of hearing aid users prefer different amplification or compression gains than those provided by standard prescriptive fitting rationales. Hearing aid gain adjustments to accommodate for variations in hearing preferences are typically performed manually in an adhoc and not in a systematic manner by clinicians, e.g. [6]. Recognizing this major limitation, some hearing aid manufacturers have attempted to incorporate user preferences into the hearing aid fitting process but their approaches are proprietary and have not been open to examination, e.g. [7].

Our research team has been working on developing machine learning solutions for systematic personalization of hearing aid amplification or compression prescriptions. In an earlier attempt [8, 9], a human-in-the-loop deep reinforcement learning (DRL) method was developed by our research team to personalize DSLv5 prescription amplification or compression via a deep neural network. Users' hearing preferences of paired comparisons of audio files were used to train the deep neural network. Whereas the DRL personalization provided promising results, the offline training of its deep neural network posed a major limitation for field deployment in real-world audio environments. In [10], an online personalization method was developed by our research team to overcome the offline training shortcoming of the DRL method by using maximum likelihood inverse reinforcement learning (MLIRL).

The online training feature of the MLIRL method enables its field deployment via a smartphone app. Such an app is developed and discussed in this paper. In other words, the objective of this paper is the real-time implementation of the MLIRL method as a smartphone app aimed at achieving personalized DSLv5 amplification or compression in an easy-

979-8-3503-3959-8/23/$31.00 ©2023 IEEE

to-use way in real-world audio environments. This is the first time that a personalization solution for hearing aid amplification has been carried out in an online training manner in real-time. This real-time implementation thus provides a transition of the personalization algorithm from a theoretical and lab tested concept to practical deployment. For training, users only need to select their hearing preference from two algorithm determined sets of gains over a number of paired comparison iterations in order to reach an optimal personalized setting. For testing, users indicate their preference to the two settings of personalized versus original prescription which are not identified to them and are randomly presented.

The remainder of this paper is organized as follows. In section 2, an overview of the MLIRL personalization method is provided. In section 3, the implementation steps of running this method as a smartphone app are covered. The app real-time characteristics and outcome are then reported in section 4. Finally, the conclusion is stated in section 5.

2. OVERVIEW OF MLIRL PERSONALIZATION

In this section, an overview of MLIRL is stated. For more details, readers are referred to [10-12]. In reinforcement learning, an agent is trained to take actions to maximize a known reward function. However, in inverse reinforcement learning, the reward function is unknown and is inferred from a user's feedbacks based on actions taken. The inferred reward function denotes the user's preference.

In the application under discussion in this paper, states and actions are defined in a common space $\mathcal{S}$ whose elements correspond to gains G in a number of frequency bands n as follows:

$$G_{new}(n) = G_{DSLv5}(n) \times scale(n) \tag{1}$$

where $G_{DSLv5}(n)$ denotes the gain value in the n^{th} frequency band according to the standard DSLv5 prescription, and $scale$ denotes a scale change of that gain to generate a new gain value G_{new} in the space $\mathcal{S}$.

The DSLv5 prescriptive gain set is considered to be the initial or starting state at the beginning of the training session. A random gain set from $\mathcal{S}$ is then chosen as an action. Each iteration consists of a paired comparison and in each paired comparison, the user would select the preferred state-action from the pair. In our application here, audio signals amplified or compressed by two gain sets generate such paired comparisons. The audio preferred by the user is considered to be the new state for a next iteration.

Preference selections that a user makes across a number of iterations form a trajectory denoted by $t = \{(s_1, a_1, r_1), (s_2, a_2, r_2), \dots\}$, where r indicates the preference selection which is set to 1 when the user prefers action a in state s and to -1 otherwise. The user interacts with the environment for several trajectories and all the trajectories form a so-called demonstration $\mathcal{D}$.

A feedback model f is then defined to encode the user's feedbacks represented by a demonstration as follows:

$$f\left(s, a_i, r_{a_j} = 1\right) = \begin{cases} 1-\varepsilon, & if\ a_i = a_j \\ \dfrac{\varepsilon}{|\mathcal{S}|-1}, & if\ a_i \neq a_j \end{cases} \quad \forall a_i \in \mathcal{S}, \tag{2}$$

$$f\left(s, a_i, r_{a_j} = -1\right) = \begin{cases} -(1-\varepsilon), & if\ a_i = a_j \\ \dfrac{-\varepsilon}{|\mathcal{S}|-1}, & if\ a_i \neq a_j \end{cases} \quad \forall a_i \in \mathcal{S},$$

where ε denotes the probability of error caused by inconsistencies in the user's feedback and i & j indicate indices of action iterations. Under a specific state s, the favored or preferred action a_j gets the probability of 1- ε while all the other actions a_i where $i \neq j$ share ε error. At the end of each trajectory, the preference model is defined as follows:

$$\mathcal{H}(s, a_i | r_s) = \sum_{j=1}^{|r_s|} f\left(s, a_i, r_{a_j}\right), \qquad \forall a_i \in \mathcal{S}. \tag{3}$$

Next, the above preference model is used to find the best action according to the optimization framework of maximum likelihood. The reward function in our application is considered to be this linear function $\mathcal{R}_\Omega(s,a) = \Omega^T \varphi(s,a)$, where φ is assigned to be a known n-dimensional state-action function and Ω an unknown weighting vector to be determined. The following state-action function is considered as discussed in more details in [10]:

$$Q_\Omega(s,a) = \mathcal{R}_\Omega(s,a) + \gamma \sum_{s' \in \mathcal{S}} \sum_{b \in \mathcal{S}} \pi_\Omega(s', b) Q_\Omega(s', b). \tag{4}$$

where π_Ω denotes a policy indicating the probability of the user choosing action a in state s according to the Boltzmann distribution as follows:

$$\pi_\Omega(s,a) = \mathbb{P}(a_t = a | s_t = s) = \frac{e^{\alpha Q_\Omega(s,a)}}{\sum_{a' \in \mathcal{S}} e^{\alpha Q_\Omega(s,a')}} \tag{5}$$

Then, the likelihood L of a demonstration $\mathcal{D}$ with the preference model $\mathcal{H}$ can be expressed as

$$L(\mathcal{D} | \Omega, \mathcal{H}) = \prod_{(s,a) \in \mathcal{D}} \left[\pi_\Omega(s,a)^{\mathcal{H}(s,a)}\right] \tag{6}$$

As mentioned above, there could be inconsistencies in the user's feedback which means a selected action cannot be regarded to be perfectly correct. The use of a preference model allows determining the degree of correctness of a selected action in a state. The objective of the MLIRL method is thus to seek the reward function $\mathcal{R}_\Omega$ that maximizes the likelihood of a given demonstration, that is

$$\Omega^* = \mathrm{arg}max_{\Omega}\ L(\mathcal{D}|\Omega,\mathcal{H}) \quad (7)$$

The personalized gain set is the one corresponding to the final or optimum Ω.

3. DEVELOPED SMARTPHONE APP TO ACHIEVE PERSONALIZATION IN THE FIELD

In this section, some of the key implementation aspects of the developed smartphone app are discussed. The MLIRL training/testing are all coded in C/C++ for both iOS and Android versions of the app. These codes are then embedded in the Android Java as well as iOS Objective-C shells previously developed in [13]. Figure 1 shows a block diagram of the audio processing pipeline of the smartphone app that is based on the MLIRL online personalization training/testing.

As discussed in [14, 15], in order to process audio frames with the lowest latency offered by the i/o hardware of smartphones, the sampling frequency needs to be set to 48 kHz with the preferred i/o buffer size of 64 samples for iOS smartphones and 96 to 512 samples depending on the Android smartphone. In general, iPhones have a lower audio latency (about 10 ms) compared to Android smartphones (about 60 ms).

An input and an output circular buffer are used in the app to ensure that the audio processing can run in real-time at the lowest i/o latency. The input circular buffer is designed to collect input samples for a 20 ms input buffer to include two frames of size 10 ms (duration often used in audio apps). The sampling frequency and i/o buffer size dictate the maximum amount of time available for audio processing of frames. This time corresponds to 64/48000 = 1.3 ms for iOS devices or iPhones and 96/48000 = 2 ms to 512/48000 = 10 ms for Android devices. The audio processing time per frame cannot exceed these amounts of time as otherwise frame skipping occurs and a real-time throughput cannot be achieved.

The FIR filter shown in Fig. 1 applies a smooth frequency response based on the gain values in dB in these five frequency bands that are commonly used in audiograms: [0 Hz-500 Hz], [500 Hz-1000 Hz], [1000 Hz-2000 Hz], [2000 Hz-4000 Hz], [4000 Hz-above]. Although nine bands are more widely used in commercially available hearing aids, they are merged and mapped into the above five bands in order to lower the training time noting that the app is easily modifiable to include any number of bands. More specifically, the action space is reduced to $2^5 = 32$ actions when using five bands as compared to $2^9 = 512$ actions when using nine bands. The filter design module shown in Fig. 1 applies a cosine interpolation function [16] to the five gain values of the frequency bands to generate a smooth frequency response. Then, a 64-tap FIR filter derived from a filter design module is used to approximate the desired frequency response for a specified gain set.

When the app is opened, the user's audiogram needs to be entered. Then, the gain values for the above five frequency bands are generated based on the DSLv5 prescription and the user's audiogram which is entered into the app. These gain values are saved in the internal storage of the smartphone and are used as the initial state of the personalization process. Before performing a personalization training session, a label of the audio environment is entered into the app to easily identify the audio file of the environment saved on the smartphone.

Figures 2(a) and 2(b) illustrate the GUI of the training and testing sessions of the app, respectively. Once a training session is launched, the user hears the audio sound in the environment captured by the smartphone microphone without any gain values applied. The listening can be done through a wired headset, a Bluetooth headset, or Bluetooth-enabled hearing aids whose amplification or compression is set to flat or no gain. By tapping the button "Gain Set A" or "Gain Set B", corresponding gain values are applied. Then, the user indicates his or her hearing preference between these two gain sets (paired comparison) by clicking one of these buttons: Prefer A, Prefer B, No Preference. The button "No Gain Applied" is used if one desires to remove an applied gain set. Aside from the iteration number, an episode number denoting 31 iterations is also displayed to indicate the progress of training. Normally, 7 episodes of 31 iterations lead to obtaining the optimal personalized gain set for a user. The values of the two gain sets being applied are also displayed. At the end of each episode, the best action or gain set gets updated and displayed at the bottom of the GUI. After conducting the training session in an online manner, the standard DSLv5 and the personalized DSLv5 settings are presented to the user to examine whether the personalized setting is more preferred over the standard setting.

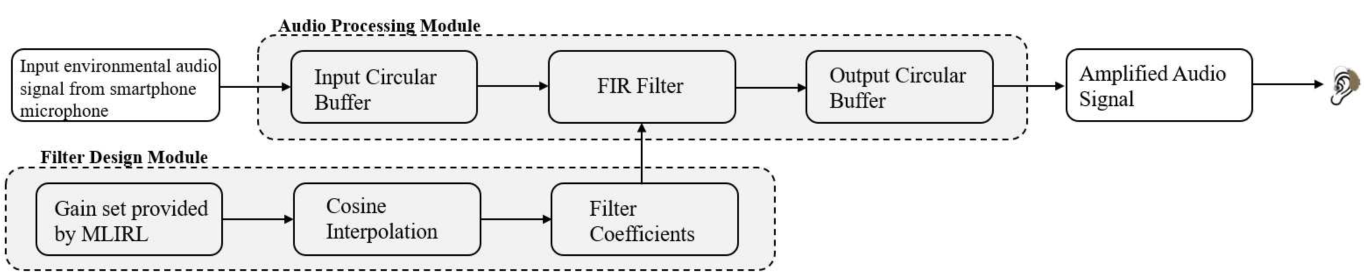

Figure 1 - Audio processing pipeline of the developed smartphone app.

(a) (b)

Figure 2 - GUI of (a) training and (b) testing sessions.

4. REAL-TIME OPERATION RESULTS

In this section, the real-time operation of the developed app in the field is reported. Such a real-time field operation has enabled the personalization method to be transitioned to field deployment. A videoclip of the app can be viewed at www.utdallas.edu/~kehtar/PersonalizedMLIRL.mp4. This videoclip exhibits how the personalization process of training and testing is carried out by the app.

The clinical study reported in [10] showed that the personalized DSLv5 setting was preferred by a large margin (10 times on average) over the standard DSLv5 setting. This study included ten hearing impaired adult subjects (ages 18-80) under an approved IRB (Institutional Review Board) at University of Texas at Dallas.

To verify the operation of the app, five users were asked to go through its training and testing sessions in three real-world audio environments with different levels of background noise. All the users indicated the ease with which the app could be trained in the field and stated that after training, they preferred the personalized setting over the standard setting during the testing in each of the audio environments. In Figure 3, an example of the outcome of a training session in the field is shown for a subject with the audiogram of [25, 30, 30, 40, 45] dB corresponding to the frequencies of 0.5, 1, 2, 4, 6 kHz, respectively. The audiogram indicates a greater hearing loss in high frequencies relative to low frequencies. As can be seen from this figure, the personalized DSLv5 frequency response curve differs from that of the standard DSLv5 frequency response curve. During the testing session, based on 50 paired comparisons, the subject indicated 100% preference for the personalized setting.

Table 1 lists the processing time per frame for an Android smartphone (Xiaomi9 running Android11 operating system with preferred buffer size of 144 samples or 3 ms) and for an iPhone (iPhone13 running iOS16 operating system with preferred buffer size of 64 samples or 1.3 ms) that were used to run the app. The processing time per frame (averaged over 1000 frames) for the Android smartphone corresponds to 0.71 ms and for the iPhone to 0.75 ms, which are well below the allowable limit of 3 ms and 1.3 ms, respectively. These frame processing rates indicate that every frame gets processed and no frame is skipped, thus achieving a real-time throughput. The i/o column in Table 1 denotes the latency of the app audio processing. As expected, this latency is lower for iPhone smartphones as compared to Android smartphones.

Table 2 shows the memory and CPU usage of the app on the Android smartphone and the iPhone used. As can be seen from this table, both the CPU and memory consumptions of the developed app are quite low.

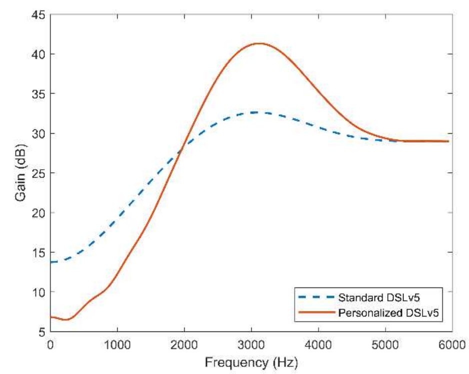

Figure 3 - An example of a training session by a subject showing the difference between the standard DSLv5 and the personalized DSLv5 gains or frequency response curves.

App Version	I/O Latency	Filtering
iOS	9ms	0.75ms
Android	66ms	0.71ms

Table 1 - Frame processing rate of the app.

App Version	CPU	Memory
iOS	9%	39.6 MB
Android	3%	111.6 MB

Table 2 - CPU and memory usage of the app.

5. CONCLUSION

In this paper, a real-time smartphone app has been developed to enable field personalization of the DSLv5 amplification prescription used in hearing aids. The personalization is carried out based on a previously developed maximum likelihood inverse reinforcement learning method. The online training feature of the app enables clinical studies to be carried out in the field or in real-world audio environments for the purpose of improving the widely used hearing aid DSLv5 prescription. The same app can be easily redesigned for other hearing aid prescriptions. In our future work, we plan to develop similar apps for other hearing aid prescriptions such as NAL-NL2.

REFERENCES

[1] M. Bagatto, S. Moodie, S. Scollie, R. Seewald, K. Moodie, J. Pumford, and K. Liu, “Clinical protocols for hearing instrument fitting in the Desired Sensation Level method,” *Trends in Amplification*, vol. 9, pp. 199-226, 2005.

[2] G. Keidser, H. Dillon, M. Flax, T. Ching, T., and S. Brewer, “The NAL-NL2 prescription procedure,” *Audiology Research*, vol. 1.e24, 2011.

[3] A. Ni, S. Akbarzadeh, E. Lobarinas, and N. Kehtarnavaz, “Personalization of hearing aid fitting by adaptive dynamic range optimization,” *Sensors*, vol. 22, paper no. 6033, 2022.

[4] G. Keidser, A. O’Brien, L. Carter, M. McLelland, and I. Yeend, “Variation in preferred gain with experience for hearing aid users,” *International Journal of Audiology*, vol. 47, pp. 621-635, 2008.

[5] G. Aldaz, S. Puria, and L. Leifer, “Smartphone-based system for learning and inferring hearing aid settings,” *Journal of American Academy of Audiology*, vol. 27, pp. 732-749, 2016.

[6] J. Chalupper, D. Junius, and T. Powers, “Algorithm lets users train aid to optimize compression, frequency shape, and gain,” *The Hearing Journal*, vol. 62, no. 8, p. 26, 2009.

[7] Widex, “Machine learning makes this hearing aid smart,” https://www.widex.pro/en/products/evoke-hearing-aid-technology/programmable-hearing-aids, 2019.

[8] N. Alamdari, E. Lobarinas, and N. Kehtarnavaz, “Personalization of hearing aid compression by human-in-the-loop deep reinforcement learning,” *IEEE Access*, vol. 8, 203503-203515, 2020.

[9] S. Akbarzadeh, N. Alamdari, C. Campbell, E. Lobarinas, and N. Kehtarnavaz, “Word recognition clinical testing of personalized deep reinforcement learning compression,” *Proceedings of 14th IEEE Dallas Circuits and Systems Conference*, Richardson, TX, 2020.

[10] S. Akbarzadeh, E. Lobarinas, and N. Kehtarnavaz, “Online personalization of compression in hearing aids via maximum likelihood inverse reinforcement learning,” *IEEE Access*, vol. 10, pp. 58537 - 58546, 2022.

[11] M. Vroman, “Maximum Likelihood Inverse Reinforcement Learning,” PhD Dissertation, State University of New Jersey, 2014.

[12] N. Mourad, A. Ezzeddine, B. Nadjar Araabi, and M. Nili-Ahmadabadi, “Learning from demonstrations and human Evaluative feedbacks: Handling sparsity and imperfection using inverse reinforcement learning approach,” *Journal of Robotics*, paper id 3849309, 2020.

[13] N. Kehtarnavaz, A. Sehgal, S. Parris, and A. Azarang, *Smartphone-Based Real-Time Digital Signal Processing*, Third Edition, Morgan and Claypool Publishers, 2020.

[14] N. Alamdari, E. Lobarinas, and N. Kehtarnavaz, “An educational tool for hearing aid compression fitting via a web-based adjusted smartphone app,” *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 7650–7654, 2019.

[15] 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* (ICCE), pp. 1-6, 2018.

[16] P. Bourke, “Interpolation methods,” http://paulbourke.net/miscellaneous/interpolation/, 1999.