A Realtime, Open-Source Speech-Processing Platform for Research in Hearing Loss Compensation

Harinath Garudadri, Arthur Boothroyd†, Ching-Hua Lee*, Swaroop Gadiyaram*, Justyn Bell*, Dhiman Sengupta‡, Sean Hamilton‡, Krishna Chaithanya Vastare*, Rajesh Gupta‡, and Bhaskar D. Rao**

*Department of Electrical and Computer Engineering, University of California, San Diego
†School of Speech, Language, and Hearing Sciences, San Diego State University
‡Department of Computer Science and Engineering, University of California, San Diego
hgarudadri@ucsd.edu, aboothroyd@cox.net, {chl438, rgadiyar, jwbell, dhimnsen, skhamilt, kvastare, gupta, brao}@ucsd.edu

***Abstract*—We are developing a realtime, wearable, open-source speech-processing platform (OSP) that can be configured at compile and run times by audiologists and hearing aid (HA) researchers to investigate advanced HA algorithms in lab and field studies. The goals of this contribution are to present the current system and propose areas for enhancements and extensions. We identify (i) basic and (ii) advanced features in commercial HAs and describe current signal processing libraries and reference designs to build a functional HA. We present performance of this system and compare with commercial HAs using "Specification of Hearing Aid Characteristics," the ANSI 3.22 standard. We then describe a wireless protocol stack for remote control of the HA parameters and uploading media and HA status for offline research. The proposed architecture enables advanced research to compensate for hearing loss by offloading processing from ear-level-assemblies, thereby eliminating the bottlenecks of CPU and communication between left and right HAs.**

***Index Terms*—Hearing aids, Open Speech Platform (OSP), speech and audio processing**

I. Introduction

Current hearing aids (HAs) are proprietary devices developed by a small number of for-profit commercial companies; due to their closed nature they are not suitable for research in hearing healthcare. The goal of this paper is to fill in this void by providing an open-source, reconfigurable, wearable, realtime speech-processing platform that uses processing and data collection approaches not available in commercial HAs. The developed system, called Open Speech Platform (OSP) [1], is to improve healthcare for hearing loss and associated disorders by enabling audiologists, hearing scientists, and clinicians with advanced instruments based on innovative radios, signal processing, and embedded computing.

The OSP provides an architecture that enables advanced research to compensate for hearing loss. It can be configured at compile and run times for researchers to investigate new HA algorithms in lab and field studies, thus enabling hearing science community to advance hearing healthcare. It is anticipated that this will accelerate translation of technology advances and hearing healthcare research studies into widespread clinical use. On the other hand, industrial sponsors can also commercialize our open source acoustics, hardware, firmware, software, and systems developed and validated by academic community from engineering and clinical disciplines.

Fig. 1 shows the system setup of the OSP. A software developed for the OSP, called Real-Time Master Hearing Aid (RT-MHA), is implemented in ANSI C with basic and advanced features in commercial HAs. The system runs on a laptop and connects to ear-level-assemblies via a custom printed circuit board and an off-the-shelf audio interface box. It has an overall latency of 7.98 msec and connects to an Android App for adjustment of HA parameters. Offloading processing from ear-level-assemblies eliminates the bottlenecks of CPU and communication between left and right HAs. The system is currently suitable for lab studies.

The rest of this paper is organized as follows. In Section II, we describe the RT-MHA and related signal processing techniques implemented in the current version (Release 2017A). In Section III, we provide software modules for the OSP. In Section IV, a comparison of the developed system with commercial HAs is presented. In Section V, we discuss the user device for remote control of the HA parameters. In Section VI, we present conclusion and discuss future work.

II. RT-MHA

In this section we provide an overview of the basic functionalities of the RT-MHA software. Fig. 2 shows the block diagram of the RT-MHA with signal flows. This architecture with different sampling rates (96 kHz for I/O and 32 kHz for main processing) has the benefit of minimizing hardware latency and improving spatial resolution of beamforming with multiple microphones. Below, we describe the three signal processing algorithms typically considered "the basic functions of a HA." These modules are implemented in the 32 kHz domain, with 32-sample frames corresponding to 1 msec.

1) Subband decomposition
2) Wide dynamic range compression (WDRC)
3) Adaptive feedback cancellation (AFC)

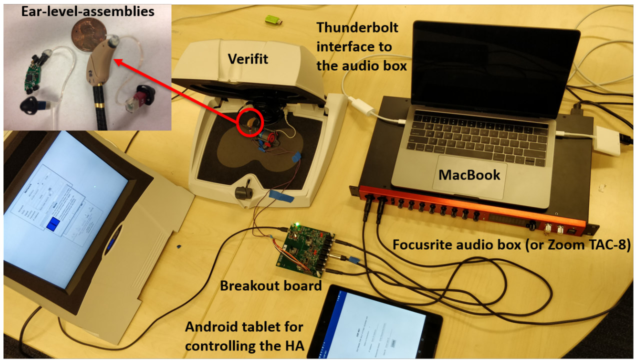

Fig. 1. Setup of the OSP system that runs on a laptop and connects to ear-level-assemblies via a custom printed circuit board and an off-the-shelf audio interface box, with an Android App connected for adjustment of HA parameters.

These algorithms are provided in source code and compiled libraries and will be briefly discussed.

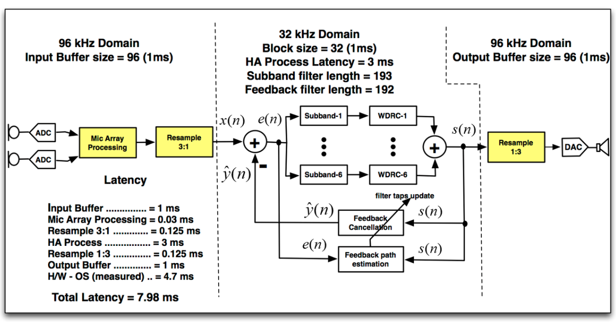

Fig. 2. RT-MHA software block diagram with signal flows. The I/O is operating at 96 kHz sampling rate and the main processing is carried out in 32 kHz sampling domain to realize subband decomposition, WDRC, and AFC.

A. Subband decomposition

In the RT-MHA, subband decomposition is provided to divide the full frequency spectrum into multiple subbands. This decomposition enables independent gain control of the HA system in each frequency region using different WDRC parameters. The decomposition is implemented as a bank of 6 finite impulse response (FIR) filters (Subband-1 to Subband-6 in Fig. 2) whose frequency responses are shown in Fig. 3. Bandwidths and upper and lower cut-off frequencies of these filters are determined according to a set of critical frequency values. The filters are designed in MATLAB and saved as .flt files for inclusion with RT-MHA. MATLAB scripts are provided for changing the number of bands and filter lengths. It is possible for users to modify the MATLAB scripts to use FIR filters of different lengths, center frequencies, and bandwidths and also different number of subbands. These changes require recompiling the library.

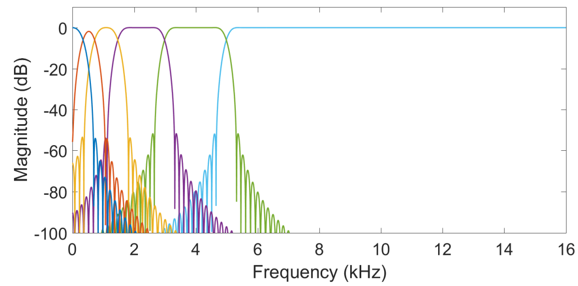

Fig. 3. Frequency response of the 6-channel filter bank whose bandwidths and upper and lower cut-off frequencies are determined according to a set of critical frequency values.

B. WDRC

The WDRC technique is one of the essential building blocks of a HA [2]. The purpose of WDRC is to provide amplification for soft sounds inaudible due to reduced sensitivity; and compression for louder sounds that become uncomfortable due to a condition called "recruitment." WDRC modifies incoming audio signal in such a way as to make the output sound as audible, comfortable, and intelligible as possible for the user. Typically, WDRC amplifies quiet sounds (40-50 dB SPL), attenuates loud sounds (85-100 dB SPL), and applies a variable gain for everything in between [3]. In the RT-MHA a multi-channel WDRC system [4] is implemented, where gain control is realized independently in each subband. The amount of gain in each subband is a frequency dependent, non-linear function of the input signal power. Basically, the overall WDRC algorithm is based on envelope detection (peak detector) and non-linear amplification (compression rule) as illustrated in Fig. 4.

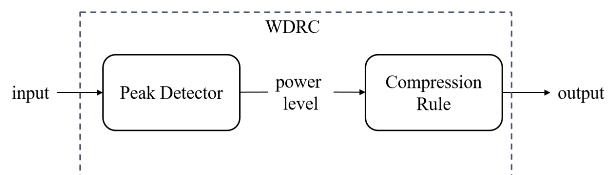

Fig. 4. The WDRC system which consists of a peak detector to track the input signal power and a compression rule to determine the output accordingly.

Primary control parameters of the WDRC system are: compression ratio (CR), attack time (AT), release time (RT), and upper and lower kneepoints (K_{up} and K_{low}) [4] as presented in Fig. 5 and Fig. 6. In each subband, a peak detector tracks the envelope variations of the input subband signal and estimates the signal power accordingly by a recursive update as:

$$
\begin{aligned}
&\text{if} \quad |x_{sb}(n)| \geq p(n) \\
&\quad p(n) = \alpha p(n-1) + (1-\alpha)|x_{sb}(n)|; \\
&\text{else} \\
&\quad p(n) = \beta p(n-1);
\end{aligned}
\tag{1}
$$

where $p(n)$ is the tracked signal power at time n, $x_{sb}(n)$ is the input subband signal, and α and β are AT and RT control parameters, respectively. The AT or RT, as illustrated in Fig 5, is the time the WDRC system takes to recover the output signal level to its steady state when a sudden rise or drop takes place in the input signal level, respectively. The amount of gain to apply will then be determined based on a compression rule as a function of the estimated input power level given by the peak detector, as shown in Fig. 6. The CR, K_{up}, and K_{low} are the control parameters for determining the compression rule. These WDRC parameters for each subband can be specified at compile time and changed at run time using the user device described in Section V.

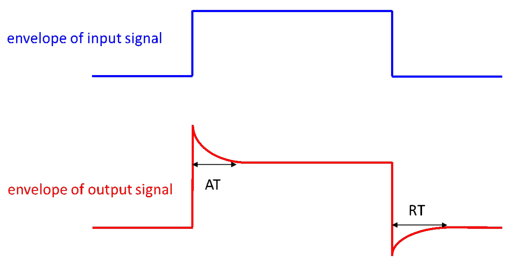

Fig. 5. Illustration of AT and RT which represent the time for the WDRC output signal level to recover to the steady state when an abrupt increase or decrease of the input signal level occurs.

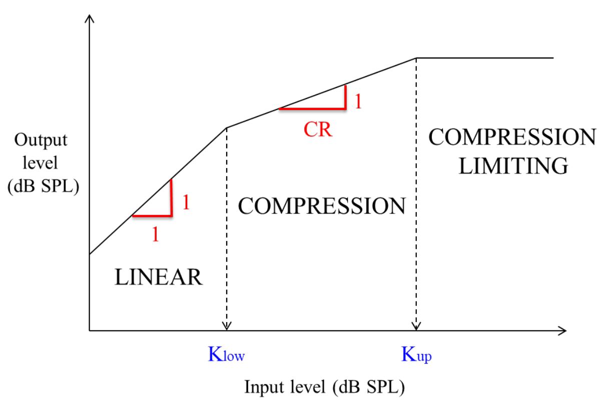

Fig. 6. Input-output curve of the compression rule which is a non-linear function of the detected input signal level.

C. AFC

Physical placement of the microphone and receiver in a HA device poses a major problem known as acoustic feedback [4]. This results from the acoustic coupling between the receiver and the microphone, in which the amplified signal through the receiver is collected by the microphone and re-amplified, causing severe distortion in the desired signal [5]. Consequently, it limits the available amount of amplification in a HA and disturbs the user due to the produced howling or whistling sounds. To overcome this problem, AFC has become the most common technique in modern HAs because of its ability to track the variations in the acoustic feedback path and cancels the feedback signal accordingly.

Fig. 7 depicts the AFC framework implemented in our RT-MHA system, which is mainly based on the filtered-X least mean square (FXLMS) method [6], [7]. In this framework, the AFC filter $W(z)$ is an FIR filter placed in parallel with the HA processing $G(z)$ that continuously adjusts its coefficients to emulate the impulse response of the feedback path $F(z)$. $x(n)$ is the desired input signal and $d(n)$ is the actual input to the microphone, which contains $x(n)$ and the feedback signal $y(n)$ generated by the HA output $s(n)$ passing through $F(z)$. $\hat{y}(n)$ is the estimate of $y(n)$ given by the output of $W(z)$. $e(n) = d(n) - \hat{y}(n)$ is the feedback-compensated signal. $H(z)$ and $A(z)$ are the band-limited and the pre-whitening filters, respectively. Both $H(z)$ and $A(z)$ are realized as FIR filters, with fixed taps. We use $u(n)$ to denote the output of the band-limited filter, and $u_f(n)$ and $e_f(n)$ to denote the outputs of the two pre-whitening filters.

With the basic FXLMS framework implemented, we provide options of using the (modified) normalized LMS (NLMS) [8] and the proportionate NLMS (PNLMS) [9]–[11]. It is possible to extend AFC with improved algorithms such as the sparsity promoting LMS (SLMS) [12]. The AFC module is provided as a library in the source code.

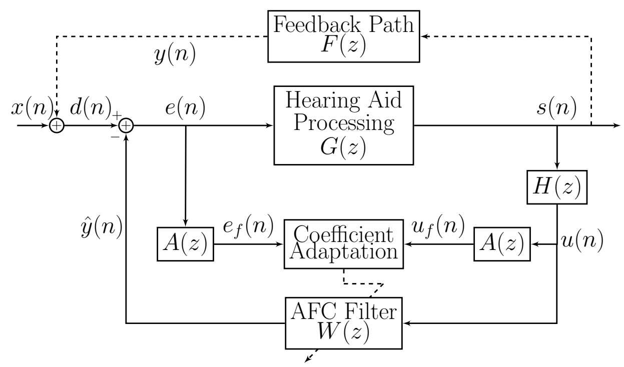

Fig. 7. The AFC Framework implemented in RT-MHA.

III. Software Modules

The software modules in Release 2017A are depicted in Fig. 8. Besides the three basic HA functions, we also identify some advanced features for a HA. Hardware dependent components are presented including both realtime and file-based processing. We also provide some offline tools for filter design, algorithm testing, and performance comparison.

For electrical engineers and computer science practitioners, the reference design is provided in the files *osp_process.c* and *osp_process.h*. For working on alternate implementations of basic HA functions, we suggest cloning a given function and call this cloned function in the reference design. The reason is that keeping the interfaces same will minimize potential bugs – it is recommended not to change the .h files unless there is a good reason. Implementation of additional functionalities can be done by adding the related .h files in the *libosp* and modifying the reference design accordingly. It is possible to reconfigure the system at runtime and compile time. The

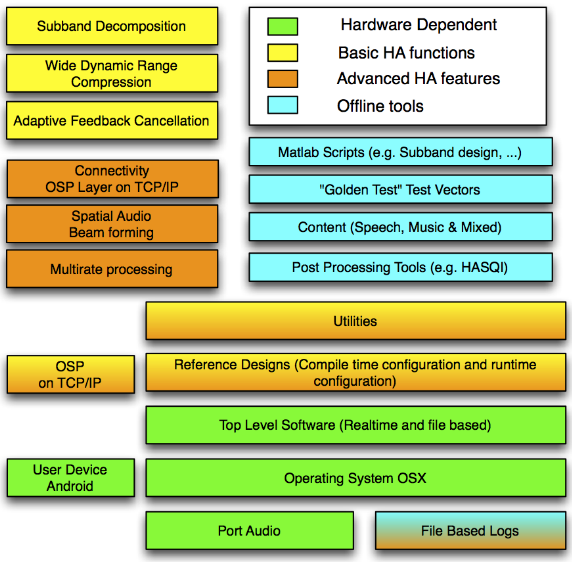

Fig. 8. Software modules in OSP Release 2017A.

current system runs realtime on a MacBook with an overall latency of 7.98 msec. This software will work with off-the-shelf microphones and receivers for realtime input and output. This is adequate for making signal processing contributions to the OSP. For some clinical investigations, form-factor accurate ear-level-assemblies (see Fig. 1) are required.

IV. RT-MHA Performance

We present in Table 1 the performance of RT-MHA for using low-power and high-power receiver (Rx) modules, compared with four advanced commercial HAs (Models A-D). We conclude that the RT-MHA software is comparable to commercial HAs and suitable for research in hearing healthcare.

TABLE I
Performance of RT-MHA, compared with commercial HAs.

AID	Units	Model A	Model B	Model C	Model D	OSP Low-power Rx	OSP High-power Rx
Average Gain	dB	40	40	25	35	40	40
Max OSPL90	dB SPL	107	112	110	111	121	130
Average OSPL90	dB SPL	106	109	108	106	112	126
Average Gain @ 50 dB	dB	37	39	25	35	35	41
Frequency Response	kHz	0.2-5	0.2-6	0.2-5	0.2-6.725	0.2-8	0.2-6.3
Equivalent Input Noise	dB SPL	27	26	30	27	29	28
Distortion @ 500 Hz	% THD	1	1	0	0	2	1
Distortion @ 800 Hz	% THD	1	1	0	0	3	2
Distortion @ 1600 Hz	% THD	0	0	0	0	1	1

V. Remote Control for HA Parameters

For runtime reconfiguration of the RT-MHA, the user device (Android App) is provided for realtime changes of the WDRC parameters. Fig. 9 shows the screenshots of the user device software with runtime parameters enabled in the current implementation. The user device software is implemented above TCP/IP layer in a software stack called OSP Layer as shown in Fig. 10. Future extensions will include uploading audio and other internal states of RT-MHA to the user device for post processing. This modular structure enables investigations in self fitting and auto fitting algorithms. The user device software is provided as source code and compiled executable.

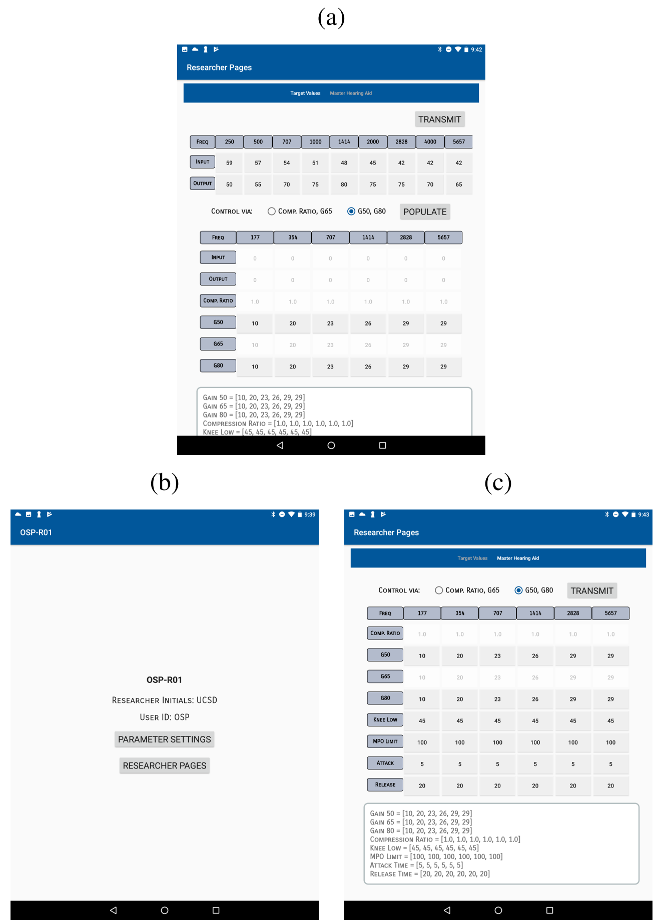

Fig. 9. Screenshot examples of the user device software: (a) shows a "researcher page" where audiologists can control input and output gain values based on the subject's pure tone audiometry; (b) shows an example page when multiple research assistants and subjects are involved in a study; and (c) shows the current RT-MHA parameters that can be controlled at runtime. These changes will take effect within 1 msec, without any audible artifacts.

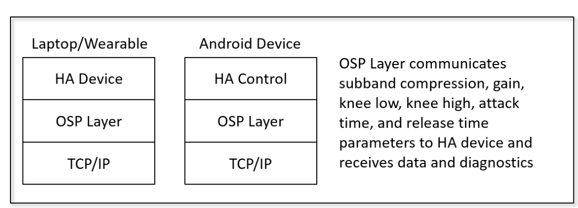

Fig. 10. The wireless communications in OSP.

VI. Conclusion and Future Work

In this paper, we presented the OSP system, which is an open source, realtime speech-processing platform that signal processing experts can contribute towards advanced HA systems. Audiologists and speech scientists can use the system to make new discoveries in hearing loss compensation, hearables, and hearing healthcare in general. We identified basic and advanced HA functions and described the current modules of the RT-MHA software. Performance comparison was presented to show the developed system is comparable to commercial HAs. Additional features such as remote control using a user device were also presented.

Future releases will include speech enhancement and noise management. We are actively porting the RT-MHA code to an embedded platform to enable field studies. The embedded platform will provide for logging parameters that will characterize background conditions. We are also developing additional tools such as ecological momentary assessments (EMAs) to enable field research.

Battery is always the biggest problem in wearable devices – we are working on reducing the computational complexity without compromising performance; we welcome similar contributions from the signal processing community.

ACKNOWLEDGMENT

This work is supported by Nation Institute of Health, NIH/NIDCD grants R33DC015046, "Self-fitting of Amplification: Methodology and Candidacy," subaward from San Diego State Unoversity and R01DC015436, "A Real-time, Open, Portable, Extensible Speech Lab" to University of California, San Diego.

REFERENCES

[1] Open Speech Platform, University of California, San Diego, 2017. Website: http://openspeechplatform.ucsd.edu/download_7yp3.html.

[2] J. M. Kates, "Principles of digital dynamic-range compression," *Trends in Amplification*, vol. 9, no. 2, pp. 45–76, 2005.

[3] S. Banerjee, *The compression handbook: An overview of the characteristics and applications of compression amplification, 4th ed.*, Starkey Laboratories, 2017.

[4] J. M. Kates, *Digital hearing aids*, Plural publishing, 2008.

[5] T. van Waterschoot and M. Moonen, "Fifty years of acoustic feedback control: State of the art and future challenges," *Proc. IEEE*, vol. 99, no. 2, pp. 288–327, 2011.

[6] J. Hellgren, "Analysis of feedback cancellation in hearing aids with Filtered-X LMS and the direct method of closed loop identification," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 2, pp. 119–131, 2002.

[7] H.-F. Chi, S. X. Gao, S. D. Soli, and A. Alwan, "Band-limited feedback cancellation with a modified filtered-X LMS algorithm for hearing aids," *Speech Commun.*, vol. 39, no. 1-2, pp. 147–161, 2003.

[8] J. E. Greenberg, "Modified LMS algorithms for speech processing with an adaptive noise canceller," *IEEE Trans. Speech Audio Process.*, vol. 6, no. 4, pp. 338–351, 1998.

[9] D. L. Duttweiler, "Proportionate normalized least-mean-squares adaptation in echo cancelers," *IEEE Trans. Speech Audio Process.*, vol. 8, no. 5, pp. 508–518, 2000.

[10] J. Benesty and S. L. Gay, "An improved PNLMS algorithm," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2002, pp. 1881–1884.

[11] C. Paleologu, J. Benesty, and S. Ciochină, "An improved proportionate NLMS algorithm based on the l_0 norm," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2010, pp. 309–312.

[12] C.-H. Lee, B. D. Rao, and H. Garudadri, "Sparsity promoting LMS for adaptive feedback cancellation," in *Proc. Europ. Signal Process. Conf. (EUSIPCO)*, 2017, pp. 226–230.