A Realtime, Open-Source Speech-Processing Platform for Research in Hearing Loss Compensation
openspeech.ucsd.edu

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

The 51st Asilomar Conference on Signals, Systems and Computers

November 1, 2017

Center for Hearing Innovations – CHI

- Goal: To improve healthcare for hearing loss and associated disorders by

  enabling audiologists, hearing scientists and clinicians with advanced instruments

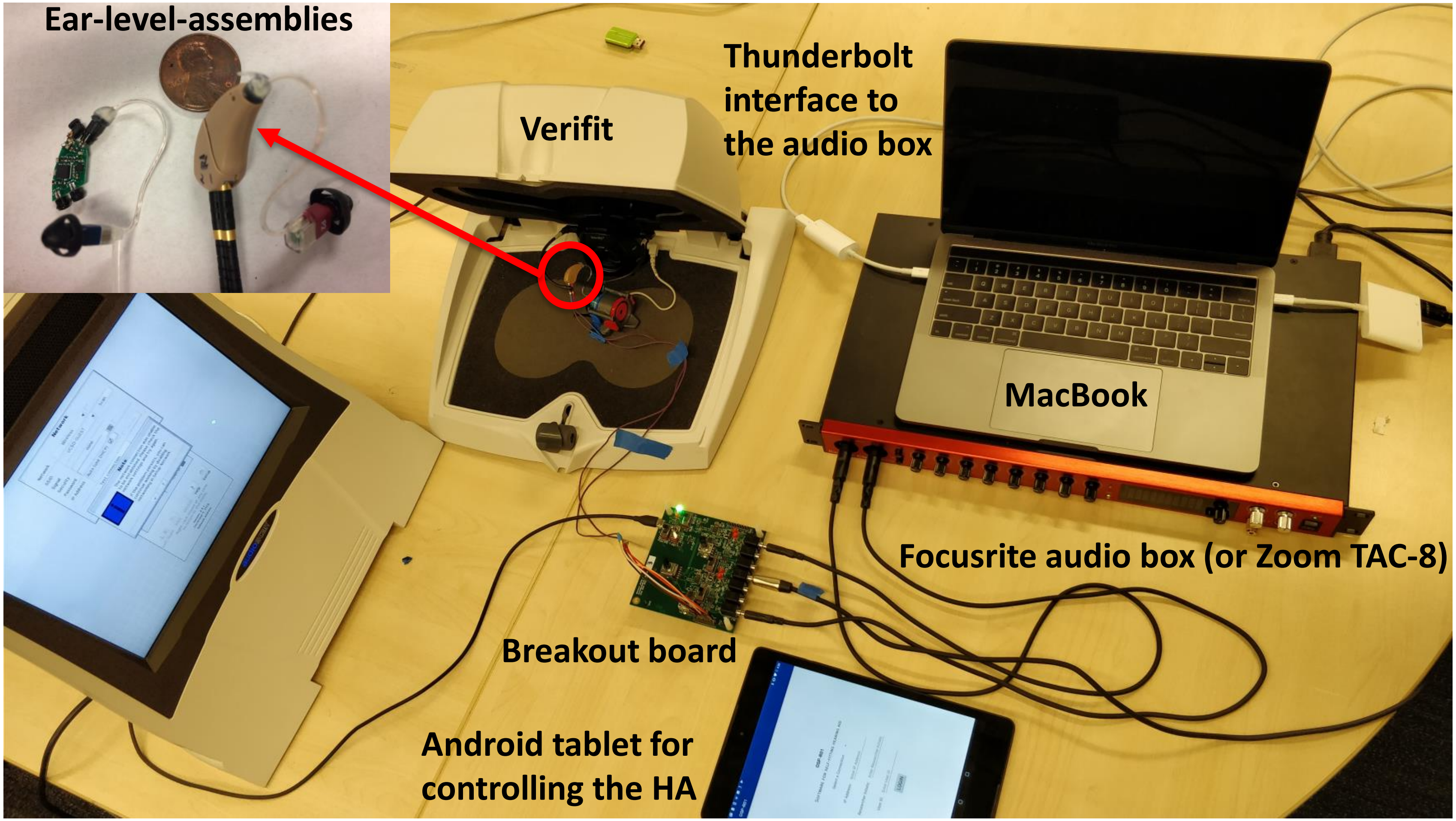
  based on innovative radios, signal processing and embedded computing.

- Commercialization: Industrial sponsors will commercialize our open source acoustics, hardware, firmware, software and systems developed and validated by academic community from engineering and clinical disciplines.

Outline

- Open Speech Platform (OSP): an architecture that enables advanced research to compensate for hearing loss.
- Real-Time Master Hearing Aid (RT-MHA): a software implemented with basic and advanced features in commercial hearing aids (HAs).
- Current signal processing libraries and reference designs.
- User device for remote control of the HA parameters.
- Performance comparison with commercial HAs.

The Open Speech Platform (OSP)



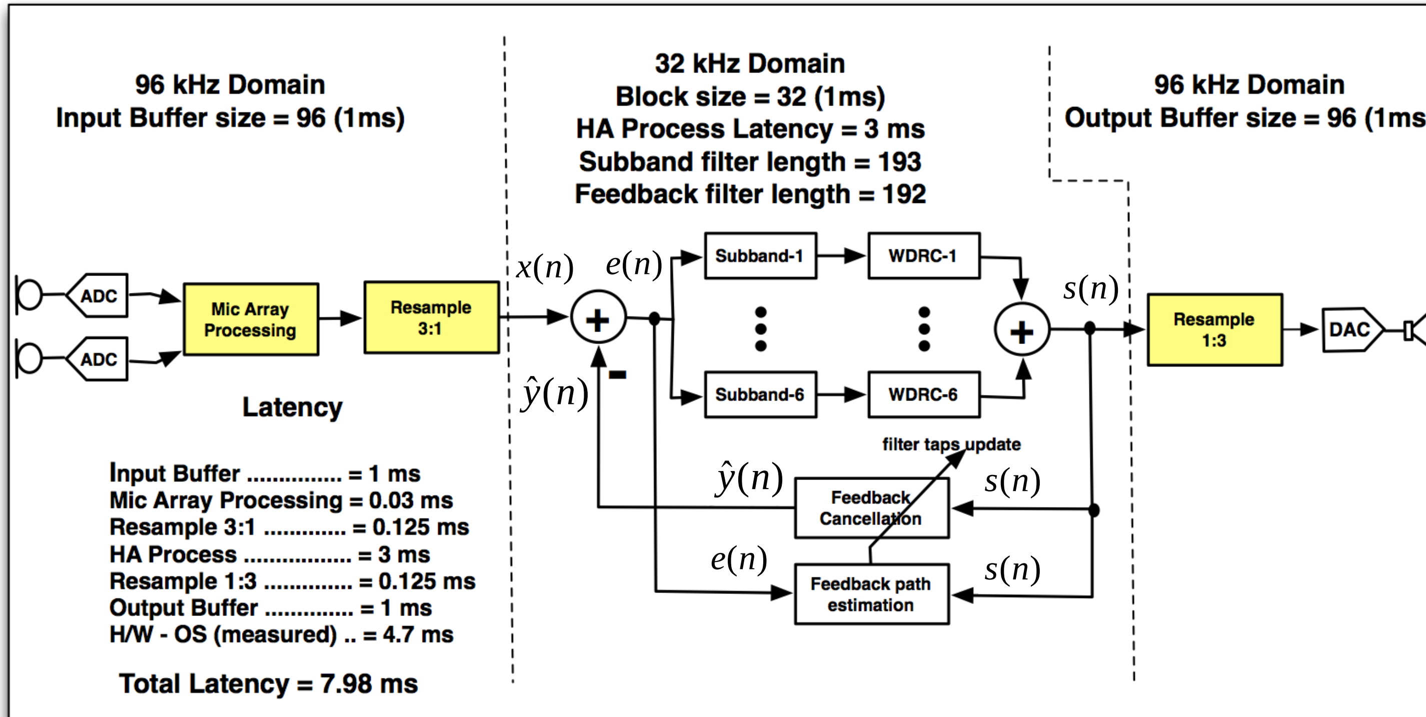
OSP for Hearing Loss Research

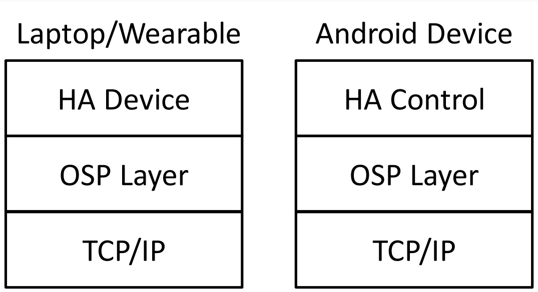
- Realtime, Wearable, Open Source.
- Offloading processing from ear level-assemblies, thereby eliminating the bottlenecks of CPU and communication between left and right HAs.
- Can be configured at compile and run times.
- Aim to support audiologists and hearing aid (HA) researchers to investigate advanced HA algorithms in field studies.

QUALCOMM INSTITUTE UC San Diego

Real-Time Master Hearing Aid (RT-MHA)

- The basic functionalities of Hearing Aid (HA) software completed in our OSP.
- Libraries are implemented in C for (i) basic and (ii) advanced features in commercial HAs.
- Runs on a MacBook with an overall latency of 7.98 ms.
- The software works with off-the-shelf microphones and speakers for real-time input and output.





OSP Layer communicates subband compression, gain, knee low, knee high, attack time, and release time parameters to HA device and Receives data and diagnostics

Hearing Aid functionality simulated in s/w. This implementation meets ANSI 3.22 requirements and currently being ported to an embedded platform.

RT-MHA System Description

- The 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.

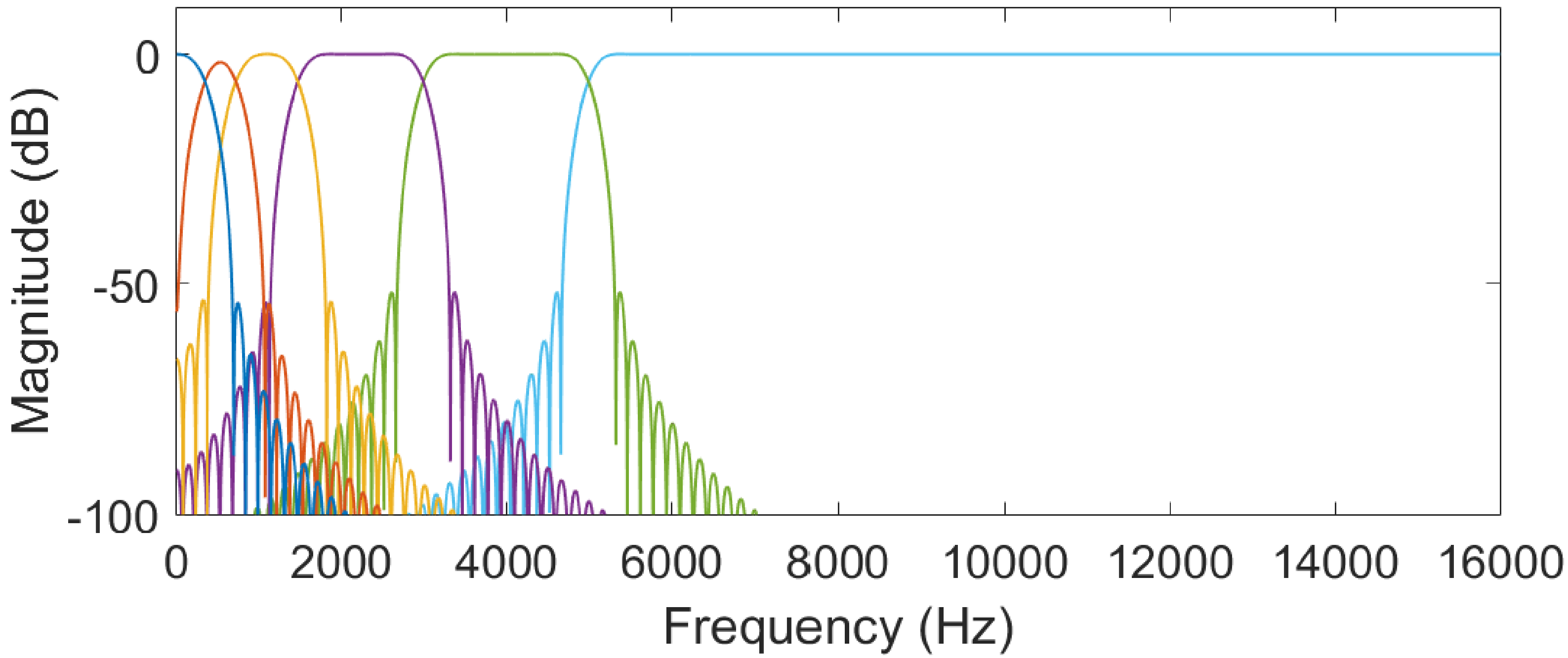
- The basic functions are implemented in the 32 kHz domain:
  (i) Subband Decomposition
  (ii) Wide Dynamic Range Compression (WDRC)
  (iii) Adaptive Feedback Cancellation (AFC)

- Algorithms are provided in source code and compiled libraries.

Subband Decomposition

- Enables independent gain control in multiple frequency regions called subbands decomposed by a set of FIR filters.
- The filters are designed in MATLAB and are saved in .flt files for inclusion with the RT-MHA software.
- Bandwidths, upper and lower cut-off frequencies of the filters are determined according to a set of critical frequency values.
- It is possible for users to modify the MATLAB scripts to use FIR filters of different length and different number of subbands.

Frequency Responses of the Subband Filters



QUALCOMM INSTITUTE UC San Diego

WDRC

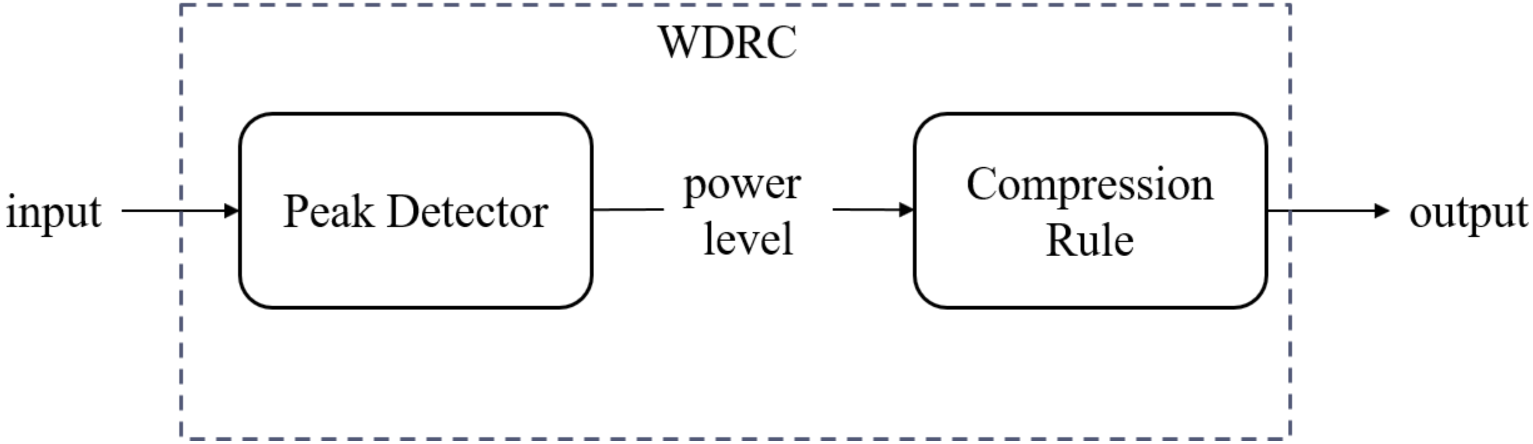
- The WDRC algorithm in the RT-MHA is a based on a version of Prof. James Kates utilizing:
  - (i) Envelope Detection (Peak Detector)
  - (ii) Nonlinear Amplification (Compression Rule)

- Primary control parameters: Compression Ratio (CR), Attack Time (AT), Release Time (RT), and Upper and Lower Knee-points (K_{up} and K_{low}).

- These WDRC parameters can be specified at compile time and changed at run time using the user device.

Peak Detector and Compression Rule

- In each subband, the **peak detector** tracks the envelope variations and estimates the signal power accordingly.
- Then the estimated input power level will become the input to a **compression rule** to determine the amount amplification.

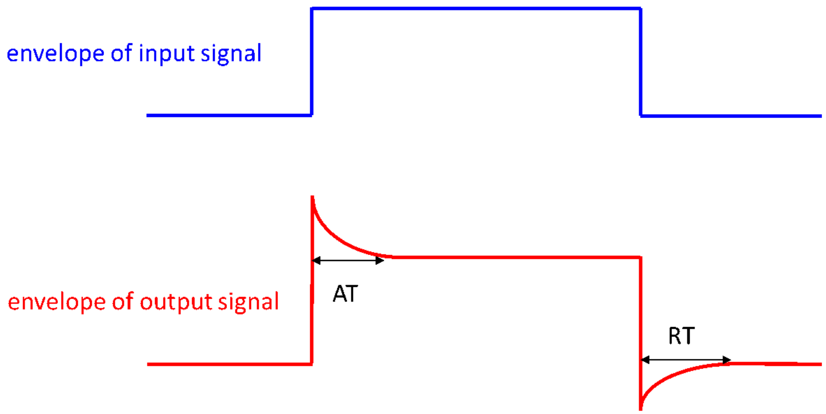


Peak Detector

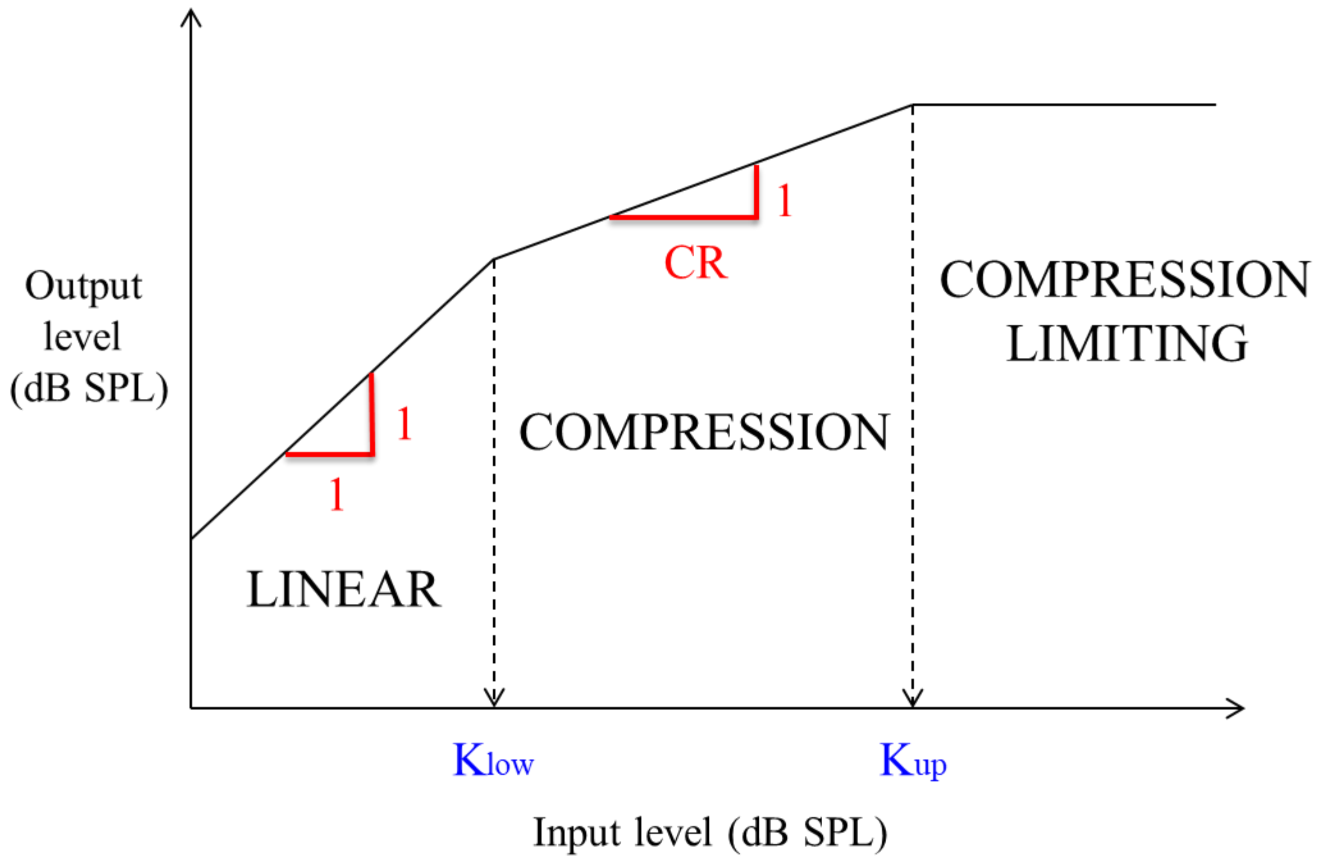
- Tracking the envelope by a recursive update:

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

where α and β are constants determined from AT and RT, respectively.



Compression Rule



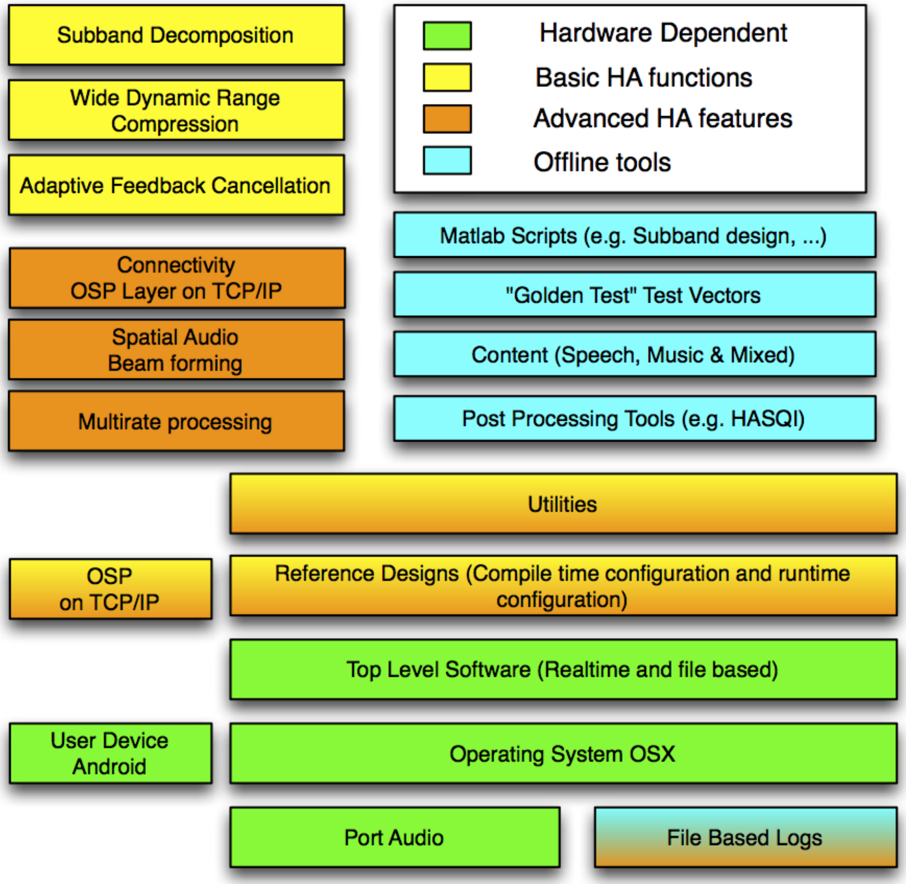
AFC

- Least Mean Square (LMS) based algorithms.
- Filtered-X LMS (FXLMS), Proportionate Normalized LMS (PNLMS), and Sparsity promoting LMS (SLMS) [1].
- A new approach to estimating the Added Stable Gain (ASG) of AFC algorithms [2] for researchers to compare AFC systems in file-based mode.

[1] Ching-Hua Lee, Bhaskar Rao, and Harinath Garudadri, "Sparsity promoting LMS for adaptive feedback cancellation," *European Signal Processing Conference (EUSIPCO)*, 2017.

[2] Ching-Hua Lee, James Kates, Bhaskar Rao, and Harinath Garudadri, "Speech quality and stable gain trade-offs in adaptive feedback cancellation for hearing aids," *The Journal of the Acoustical Society of America Express Letters (JASA-EL)*, 2017.

Software Modules in Release 2017a



Reference Designs

- The reference design is provided in the files `ospprocess.c` and `ospprocess.h`. functions.

- If you are working on alternate implementations of basic HA functions, we suggest clone a given function and call this in the reference design.

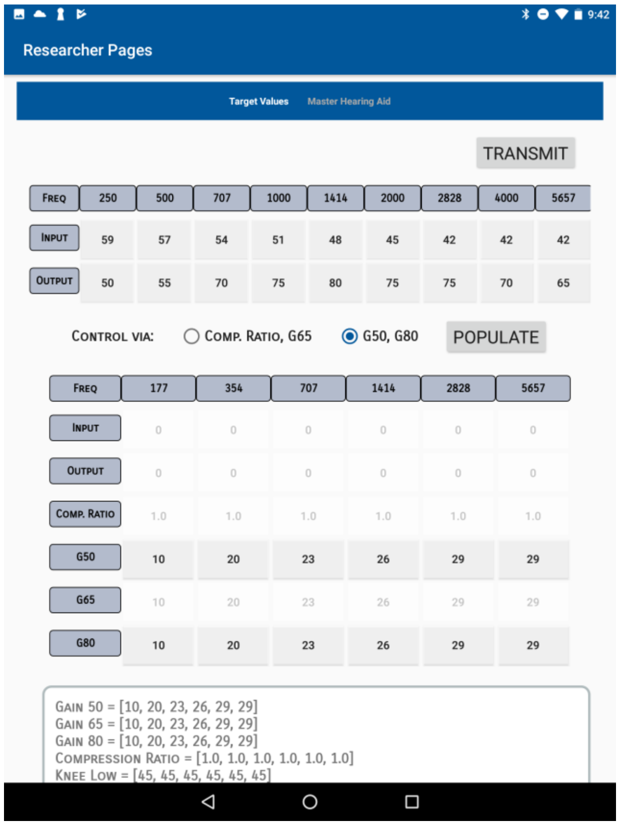
- Implementation of additional functionality can also be done by adding the related .c and .h files in the `libosp` and modifying the reference directory accordingly.

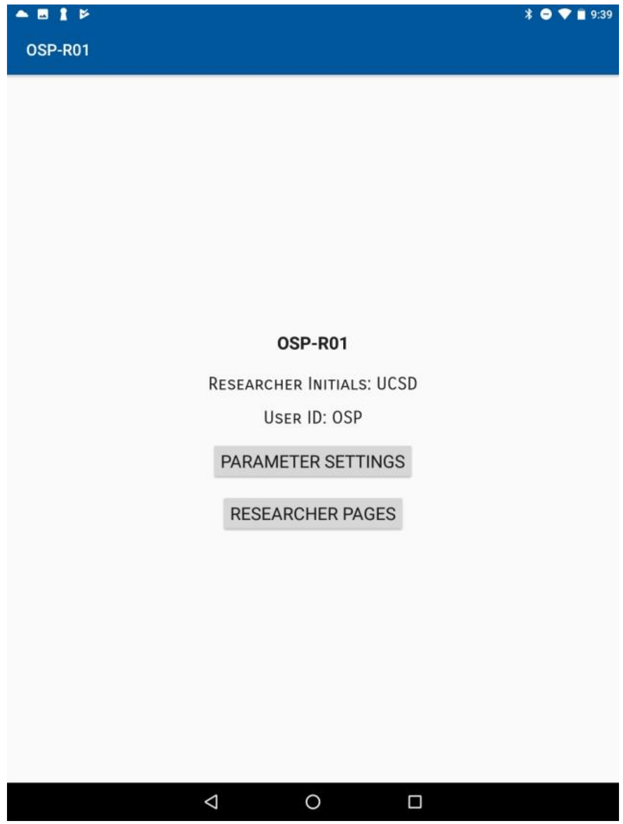
- Keeping interfaces the same will minimize code changes.

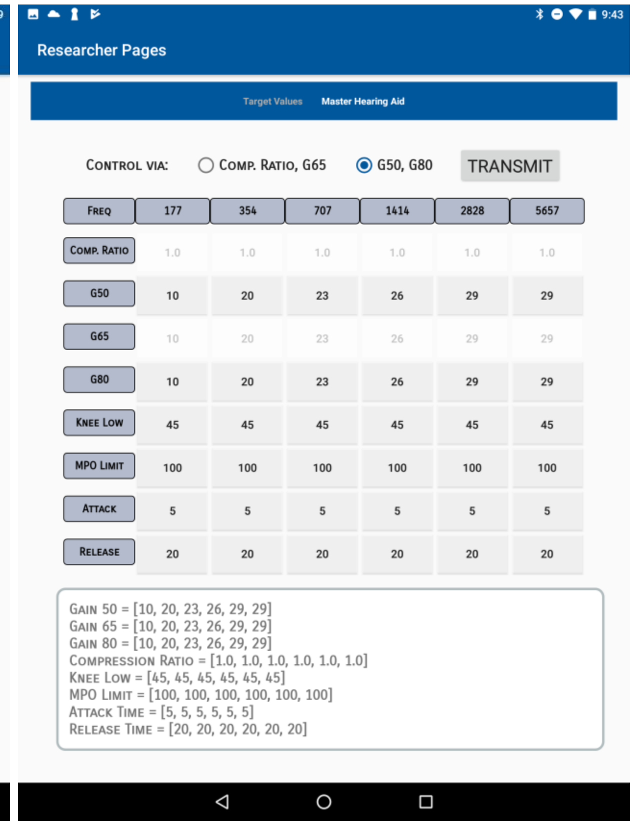
User Device

- An Android APP which provides for real-time changes to WDRC parameters.
- Implemented above TCP/IP layer in a software stack called OSPLayer.
- The modular structure enables investigations in self fitting and auto fitting algorithms.

User Interface







QUALCOMM INSTITUTE UC San Diego

RT-MHA Performance

- Compared with 4 commercial HAs (Models A – D)

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

Summary and Future Plans

- **Takeaway message**: An open source, realtime, wearable speech lab that DSP experts can contribute to – and enable new discoveries in Hearing Aids, Hearables and Hearing Healthcare in general.

- **Release 2017b** – Bug fixes and optimizations for the wearable device.

- **Release 2018a** – RT-MHA ported to the wearable device hardware.

Planned Tasks

- Ear Level Assemblies
  - Codec in ear
  - 4 mics / ear – Front, Rear, In-ear, bone conduction
  - 6-axis Inertial Motion Unit in each ear
- RT-MHA
  - Snapdragon-410c (10% CPU) or TI-OMAP L138 (80% CPU)
  - Speech Enhancement
  - Frequency lowering
- User Interface
  - Web server in the wearable device
  - Example scripts: e.g. generate stimulus → collecting response