Complexity-Effective Auditory Compensation for Digital Hearing Aids*

Yu-Ting Kuo, *Tay-Jyi Lin*, *Wei-Han Chang*, *Yueh-Tai Li*, and *Chih-Wei Liu*
Department of Electronics Engineering, National Chiao Tung University, Taiwan

Shuenn-Tsong Young
Institute of Biomedical Engineering, National Yang-Ming University, Taiwan

***Abstract* – This paper describes a complexity-effective design of auditory compensation, which is the most important building block in digital hearing aids. A multi-rate architecture and the filter bank design thereof are proposed to significantly reduce the data rates of the band-limited channels and thus the computational complexity. Several optimizations are discussed to further simplify the implementation of the dynamic range compressors in the multi-rate auditory compensation. In an 18-band 24-KHz digital hearing aid, 94% multiplications and additions are saved from a straightforward filter bank design. Moreover, the computation and memory requirements of the dynamic range compressor of each band are reduced to only 16% and 1% respectively.**

I. Introduction

The population growth of hearing loss is increasing rapidly because of the raised average human ages, so more and more people need hearing aids to improve their life quality. Digital hearing aids [1] provide much better programmability that can customize the device for individuals, but they need more computing power than analog ones. However, the small dimensions of hearing aids severely limit the sizes of the electronic devices and the battery (thus the power consumption of the devices). Due to the stringent constraints, electronic designers must try their best to optimize their devices from high-level algorithm/architecture to low-level circuit/physical designs. A generic digital hearing aid includes functional blocks such as auditory compensation, echo cancellation, noise suppression, and speech enhancement. Auditory compensation is the most important one since it compensates the hearing loss of the hearing-impaired people. People with hearing loss usually suffer from both the loss of audibility and the reduction of hearing dynamic range. Therefore, the auditory compensation performs following two functions: frequency shaping and dynamic range compression. The former compensates the frequency-dependent audibility loss by amplifying the signal with various gains on each frequency bands. The latter reduces the signal's dynamic range to fit the residual dynamic range of the hearing-impaired people.

This paper proposes a complexity-effective architecture for auditory compensation to save the hardware area and power consumption of digital hearing aids. The rest of this paper is organized as follows. Section 2 describes the functions of the auditory compensation. The proposed architecture and complexity reduction of the dynamic range compressor are shown in Section 3 and 4, respectively. The comparisons of our results are summarized in Section 5. Finally, Section 6 concludes this paper and outlines our future researches.

* This work was supported by the National Science Council, Taiwan under Grant NSC96-2220-E-009-035

II. Auditory Compensation

Fig. 1 shows the block diagram of the auditory compensation for digital hearing aids [2], which comprises an 18-band filter bank with programmable gains and a dynamic range compressor. The filter bank is based on the ANSI S1.11 [3] 1/3-octave bands, which are similar to the signal analysis mechanism of human auditory systems. The specifications of the filter bank and dynamic range compression would be described as follows.

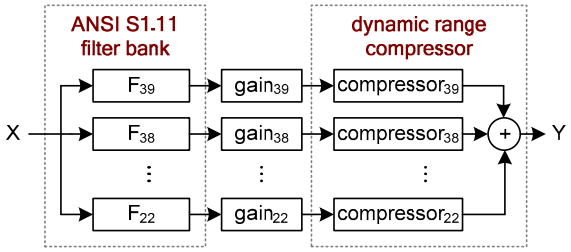

Figure 1. Block diagram of auditory compensation [2]

A. ANSI S1.11 Filter Bank

Forty-three 1/3-octave bands are defined in the ANSI S1.11 standard for the frequencies below 20 KHz. Each band's location and range are specified by its mid-band frequency and bandwidth. The mid-band frequency of the x^{th} band $f_m(x)$ is defined as

$$f_m(x) = 2^{\frac{(x-30)}{3}} \times f_r,$$

where f_r is a reference frequency set as 1,000 Hz. The bandwidth of each band is then defined as the frequencies between the lower (f_1) and upper (f_2) band-edge frequencies, which are given as

$$f_1 = f_m \times 2^{\frac{-1}{6}},\ f_2 = f_m \times 2^{\frac{1}{6}}$$

For instance, the bandwidth of the 30th band is from 891 Hz to 1,112 Hz. Besides, the performance requirement of the 1/3-octave filters is also defined in the ANSI S1.11 standard. The performance specifications include filters' relative attenuation, integrated filter response, environment sensitivity, and some other parameters.

The filter bank in Fig. 1 implements the 22nd ~ 39th ANSI 1/3-octave bands (denotes as $F_{22} \sim F_{39}$), which cover frequencies 130 ~ 8,980 Hz, with 18 parallel IIR filters since the straightforward FIR implementation of 1/3-octave filters is expensive. However, FIR bank is preferred in hearing aid applications for its linear phase

property and stability [4]. Therefore, most hearing aids use proprietary FIR banks [4][5] that however do not comply with the ANSI S1.11 standard. We will present the design of a complexity-effective ANSI S1.11 filter bank with linear phase property in Section 3.

B. *Dynamic Range Compressor*

The datapath of the dynamic range compressor is shown in Fig. 2(a) and its behavior is characterized by an I/O graph, in which the relation between the input and output sound pressure levels (SPL) is described. Fig. 2(b) shows the compressor's I/O graph, where TH_1 and TH_2 are two predefined compression thresholds. The slope of this I/O curve is unit for input SPL smaller than TH_1, half for SPL between TH_1 and TH_2, and one fourth for SPL larger than TH_2. Therefore, the sounds with higher levels are compressed more.

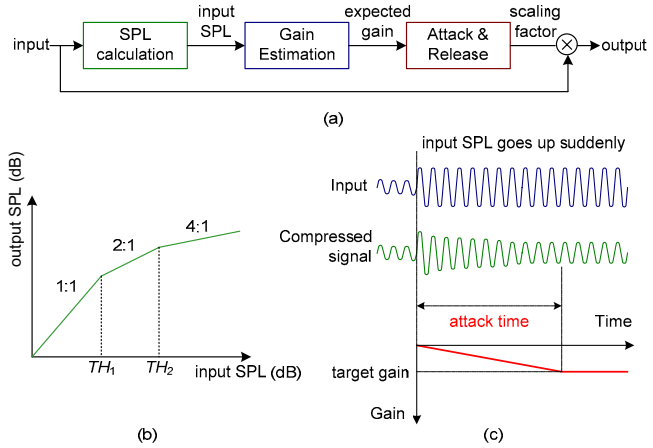

Figure 2. Compressor's (a) datapath (b) I/O graph (c) attack time

The three functional blocks of the compressor in Fig. 2(a) operate as follows. First, the SPL calculation block computes the input SPL with the following equation

$$SPL(x_n) = 20 \times \log \sqrt{\frac{1}{m} \times \sum_{i=0}^{m-1} x_{n-i}^2} = 10 \times \log\left(\frac{1}{m} \times \sum_{i=0}^{m-1} x_{n-i}^2\right), \quad (1)$$

where x_n is the input sample and m is the window length for calculating the input power. The window should be shorter than the signal's short-term stationary period (5~20 ms for speech) and longer than the period of sound waves (0.3~3 ms for 0.3~3 KHz). A 4 ms window is chosen in our design and thus m is 96 for 24 KHz sampling rate. Second, the gain estimation block calculates the expected gain for compressing the signal's level, according to the compressor's I/O graph. The last block, attack & release, controls how fast the applied gain reacts to the changes of the expected gain and calculates the corresponding scaling factor that multiplies the input samples. The control of attack and release time can reduces the changes in signals' fine detail to alleviate the jarring, pumping or fluttering perceptions [1]. Fig. 2(c) illustrates the attack operation, in which the gain gradually decreases when the input SPL suddenly increases. The attack time constant is usually much shorter than the release time to prevent the hearing-impaired people's ear being damaged by loud sounds. Our compressor set the attack and release time as 10 and 100 ms respectively.

The algorithm of the attack & release block is shown in Fig. 3, in which the current gain is approaching the target gain with the step size determined by the attack or release time constant. The target gain is only modified when the expected gain changes a lot so that it needs not to be updated frequently. The scaling factor is calculated by converting the current gain from the decibels (dB) into the linear scale.

```
if (|expected_gain - target_gain| > threshold)
{   target_gain = expected_gain;
    count = (target_gain < current_gain)? attack_time: release_time;
    step = (target_gain - current_gain)/count; }
if (count)
{   current_gain += step;
    scaling _factor = power(10, current_gain/20)
    count --; }
```

Figure 3. Pseudo code for the attack/release time control

III. Proposed Multi-rate Architecture

This section shows a complexity-effective design of the auditory compensation in Fig. 1. We use multi-rate processing to save the computations. Besides, the design of ANSI S1.11 filter bank will also be addressed.

The proposed multi-rate auditory compensation can be derived as follows. Firstly, all the 1/3-octave bands of the original auditory compensation in Fig. 1 are down-sampled to reduce computations since they are band-limited. The down-sampling rate is multiplied by two for every three bands because the bandwidth is halved for each octave. Fig. 4 depicts the multi-rate auditory compensation, which includes two filter sets; the analysis bank that is composed of F_n and the synthesis that is composed of G_n. For simplicity, we can assume $G_n = F_n$

The second step is to transform this multi-rate architecture into a pyramid structure to reduce filter complexities. This is done as follows. Exploiting the fact that the bandwidth is halved for each octave, $F_n(z)$ could be substituted by $F_{n+3}(z^2)D(z)$, where the decimation filter D is a low-pass filter. Then, we could move the down-sampler toward the datapath's input using the noble identity [6], as shown in Fig. 5(a). With the above substitutions and transformations, we could derive the pyramid structure in Fig. 5(b) for the analysis bank. Similarly, the synthesis bank can also be arranged in a pyramid structure, which would contain the filter $G_{37\sim39}$ and an interpolation filter (denoted as I). This pyramid structure avoids the implementation of extremely narrow band filters such as F_{22} and G_{22}, which require much higher filter orders. Therefore, the complexity of the filter bank can be reduced.

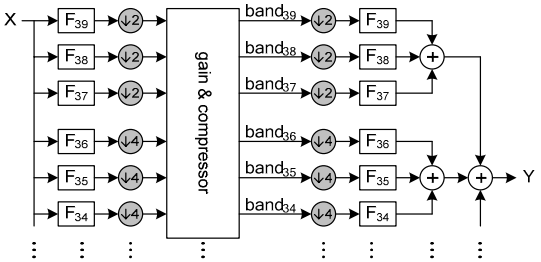

Figure 4. Multi-rate auditory compensation

Finally, we fold the pyramid structure in Fig. 4 to derive a complexity-effective and compact architecture for the multi-rate auditory compensation. Fig. 6 shows the proposed architecture, in which only a set of $F_{37} \sim F_{39}$, $G_{37} \sim G_{39}$, D, and I is implemented. Smaller octaves' bands, such as $F_{22} \sim F_{36}$ are calculated iteratively by F_{37}, F_{38} and F_{39} since those bands' bandwidths are the same after down-sampling by two. The distortions resulted from up and down-sampling are suppressed by filters $G_{37} \sim G_{39}$, D and I.

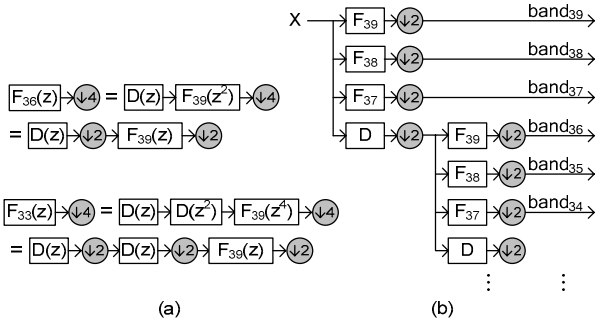

Figure 5. (a) Noble identity (b) pyramid structure

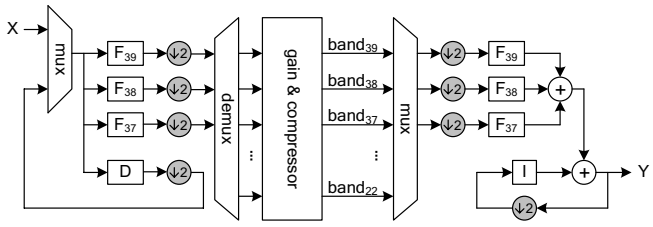

Figure 6. Proposed architecture

The filters' coefficients are designed using MATLAB `firpm` and `firpmord` functions, which synthesize FIR coefficients with the following design parameters: filter's passband ripple, stopband attenuation, passband and stopband frequencies. Since we assume $G_n = F_n$ for simplicity, the design problem is thus to find the optimal setting of these design parameters for F_{37}, F_{38}, F_{39}, D, and I such that the eighteen 1/3-octave bands all meet the ANSI S1.11 specification and the overall complexity is minimized. We design coefficients with a 2-step design flow. The first step designs F_{37}, F_{38}, and F_{39} such that their responses meet the specifications of the $37^{th} \sim 39^{th}$ bands and the orders are minimized. According to the standard, the ripple and attenuation are 1 and -60 dB respectively. Nevertheless, the standard specifies not only filters' ripple and attenuation, but also the responses in the transition bands. We thus exhaustively search the passband and stopband frequencies in their possible ranges and choose the optimal results. On the other hand, the filters D and I are designed such that the aliasing and imaging [6] distortions due to the up and down-sampling are suppressed. The passband ripples of D and I are set as same as that of $F_{37} \sim F_{39}$.

Although the filter set designed in the first step makes $F_{37} \sim F_{39}$ satisfy the specifications. However, the ripple of D and I plus the ripple of $F_{37} \sim F_{39}$ could make the $22^{nd} \sim 36^{th}$ bands violate specifications. Hence, our second step refines the coefficients such that the ripple in each band's overall response would not violate the specifications. That is, we redesign $F_{37} \sim F_{39}$ with different ripples. For each ripple of $F_{37} \sim F_{39}$, we find out how small the ripple of filter D and I should be to make the $22^{nd} \sim 36^{th}$ bands meet the specifications. The best result that has the minimal complexity is thus selected as the final solution, which shows that F_{37}, F_{38}, F_{39}, , D and I are 41, 33, 26, 35 and 41-tap filters respectively. Besides, G_n's coefficient set is the as F_n's.

IV. Complexity Reduction of Compressor

In this section, we reduce the complexity of the three functional blocks of the dynamic range compressor in Fig. 2(a), that is, SPL calculation, gain estimation and attack & release. We propose two implementations for attack & release block. Therefore, actually two optimized compression algorithm are presented. The complexity reductions of SPL calculation, gain estimation and attack & release blocks are described as follows.

A. SPL Calculation

We first define $\overline{x_n^2}$ as the input samples' mean square, which can be approximated as

$$\overline{x_n^2} = \frac{1}{m} \times \sum_{i=0}^{m-1} x_{n-i} \cong \beta \cdot \overline{x_{n-1}^2} + (1-\beta) \cdot x_n^2 \ , \tag{2}$$

where x_n is the input samples and m is the window length for calculating input power. Thus, equation (1) can be modified such that the input mean square is calculated by a single-pole IIR filter and the coefficient β can be estimated as $(m–1)/m$ [7]. The value of β is 0.9958 for m = 96 (4 ms window size and 24 KHz sampling rate). Since a single-pole IIR requires only one memory element, memory requirement is largely reduced compared with exact calculation of the mean square, which requires 96 memory storages.

B. Gain Estimation

Since the I/O graph represents the input SPL versus the output SPL in dB, the expected gain for a given input SPL can be calculated by subtracting the corresponding output SPL with the input SPL. Thus, the relationship between the input SPL and the expected gain is also a piece-wise linear curve. We modify the gain estimation such that the expected gain is directly calculated from the input SPL.

C. Attack & Release

Two different implementations of attack and release are described here. Both reduce the complexity but the mechanisms are different. The first one is a modified version of the algorithm in Fig. 3. In the modified algorithm, we calculate the target scaling factor when the target gain is updated. Thus we can make the current scaling factor approach the target without calculating the current gain. Therefore, we need not to convert the current gain from dB into linear scale for each sample. Besides, the target scaling factor is updated when the input SPL (instead of the expected gain) has significant changes. Hereby the gain estimation needs not to be performed when the input SPL has small variations. On the other hand, we also make the attack and release time more precise by compensating the effects of SPL calculation, in which the method of calculating the input power could increase the attack or release time measured.

The second implementation of attack and release is adopting an attack and release time embedded SPL calculation. That is, equation (2) can be modified as

$$\overline{x_n^2} \cong \begin{cases} \beta_a \cdot \overline{x_{n-1}^2} + (1-\beta_a) \cdot x_n^2 & x_n^2 \geq \overline{x_{n-1}^2} \\ \beta_r \cdot \overline{x_{n-1}^2} + (1-\beta_r) \cdot x_n^2 & x_n^2 < \overline{x_{n-1}^2} \end{cases} . \tag{3}$$

The value of β_a and β_r respectively affect the compressor's attack and release time. To simplify the derivation of β_a, we roughly estimate the attack time as the time that this IIR filter's step response $S(n)$ reaches 3 dB below its steady state. Thus, β_a can be calculated as

$$S(n_a) = (1-\beta_a^{(n_a+1)}) = -3\text{dB}, \quad \beta_a = (1-0.707)^{\frac{1}{n_a+1}},$$

where n_a is the attack time. For a 10 ms attack time with sampling rate 24 KHz, n_a is 240 and thus β_a is 0.9949. Similarly, β_r is 0.9995 for a 100 ms release time.

V. Results

A. Computational Complexity

To demonstrate the effectiveness of the proposed multi-rate architecture, we implement our design with 24 KHz input sampling rate. We design the multi-rate FIR bank for the 22nd ~ 39th 1/3-octave bands in the ANSI standard. Coefficients are generated using the MATLAB filter design toolbox. Table I compares our design with other filter banks for hearing aids. Most of the listed filter banks are FIR-based. [4], [5] and [8] design the filter bank for proprietary bands, while [9] uses the DFT bank. Only [2] and our design comply with the ANSI S1.11 standard. However, our FIR-based design provides linear phase property with the required multiplications comparable to that of the IIR one. On the other hand, we have also designed the parallel (single-rate) FIR bank for the same bands. The parallel FIR bank requires filters with order 25 ~ 1,488 and needs overall 3,270 multiplications per sample. That is, our proposed filter bank effectively saves 94% complexity of the conventional FIR banks for 1/3-octave bands.

Table I Comparison of Filter Banks in Hearing Aids

	[8]	[5]	[9]	[4]	[2]	Proposed
# bands	5	7	16	16	18	18
attenuation (dB)	-	40	50	60	60	60
ANSI-compliant					■	■
linear-phase	■	■	■	■		■
# multiplications	27	84	46	880	220	276
# additions	46	160	46	1,744	165	525

Our proposed multi-rate architecture also saves the computations of the dynamic range compressor. Compared with the single-rate architecture, our design saves 84% computations. Moreover, our improved compression algorithm requires only one buffer for the SPL calculation. This effectively saves 99% of the memory storage.

B. Attack & Release Time Measurement

We design and verify the optimized compression algorithms as follows. The attack and release time are specified as 10 and 100 ms respectively. The input versus output SPL is shown in Fig. 2(b). We measure the attack and release time with the procedures and stimulus described in the ANSI S3.22 standard [10]. The difference between the specified and measured times should be within 5 ms or 50%, whichever is larger, according to the standard. Table II summarizes the measured results of the original [2] and our two optimized compression algorithms, which differ in their attack and release time control mechanism. As described in Section 4, our first design implements the attack and release time with a modified version of the algorithm in Fig. 3, while the second design adopts an SPL calculation block with embedded attack and release time control. Our two designs have different behaviors and thus the different attack or release times measured. However, they both reduce the memory requirement and the measured times are all within the tolerances specified in the ANSI S3.22 standard. Thus, we are now studying our two optimized compression algorithms' subjective perceptions and their effects on the hearing aid performance to find out which one of our two designs is better.

Table II Comparison of Attack & Release Times

	[2]	1st design	2nd design
attack time (ms)	12.9 (29%)	14.1 (41%)	6.5 (-35%)
release time (ms)	90.8 (-9%)	98.7 (-1%)	72.2 (-28%)

VI. Conclusion

In this paper, a multi-rate architecture is proposed to reduce the complexity of auditory compensation. The proposed FIR-based design of the 1/3-octave filter bank has the comparable complexity with the IIR banks and saves 94% multiplications of conventional FIR banks. The multi-rate approach not only reduces the filter bank's data rate but also effectively increase the bandwidth of each band, which eases the design of FIR filters. On the other hand, the multi-rate design also reduces 84% computation of the dynamic range compressor. Moreover, we also improved the compression algorithm such that 99% of memory requirement is saved. Hence, the proposed architecture saves both the power and chip size of digital hearing aids. We are now studying subjective perceptions of our optimized compression algorithms to identify their effects on hearing aid performances. Besides, in the future we will survey the noise suppression as well as the echo cancellation algorithms and try to combine them with the auditory compensation.

References

[1] H. Dillon, *Hearing Aids,* Thieme Medical Publisher, 2001

[2] P. Y. Lin, *Feasibility Study of the Implementation of Hearing Aid Signal Processing Algorithms on the TI TMS320C6713 DSK*, Master thesis, Institute of Biomedical Engineering, National Yang Ming University, 2004

[3] *Specification for Octave-band and Fractional-octave-band Analog and Digital Filters*, ANSI S1.11-2004, Feb. 2004, Standards Secretariat Acoustical Society of America

[4] K. S. Chong, B. H. Gwee and J. S. Chang, "A 16-channel low-power nonuniform spaced filter bank core for digital hearing aid," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, Sept. 2006

[5] L. S. Nielsen and J. Sparsø, "Designing asynchronous circuits for low power: An IFIR filter bank for a digital hearing aid," *Proc. IEEE*, vol.87, no. 2, pp. 268–281, Feb. 1999

[6] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice Hall, 1993

[7] N. Magotra, S. Kamath, F. Livingston and M. Ho, "Development and fixed-pint implementation of a multiband dynamic range compreession (MDRC) algorithm," in *Proc. ACSSC,* Oct. 2000

[8] J. L. Goldstein, "Hearing aids based on models of cochlear compression using adaptive compression thresholds," U.S. Patent US20020057808, May 16, 2001

[9] T. Schneider and R. Brennan, "A multichannel compresson strategy for a digital hearing aid," in *Proc. ICASSP*, Apr. 1997

[10] *Specification of Hearing Aid Characteristics,* ANSI S3.22-2003, Aug. 2003, Standards Secretariat Acoustical Society of America