A REAL-TIME IMPLEMENTATION OF A DUAL RESONANCE NONLINEAR FILTERBANK

RAM KUBER SINGH

University of Western Sydney
Email: rtap.dev@gmail.com

Abstract—Interpreting various forms of sound waves from the human perspective is associated with the auditory pathway. The auditory pathway consists of a complex network of neurobiological systems extending from the outer ear to the auditory cortex in the brain. From a signal processing standpoint, modeling various stages of the auditory pathway has evolved to a point where every stage can be modeled separately. However, analyzing the auditory pathway from a top-down viewpoint involves the integration of fragmented models of all stages. Furthermore, in order to study the signals generated in the upstream modules of such an ecosystem, it is imperative that computing at lowerstages do not cause a bottleneck especially with a large audio input. Therefore, a real-time implementation of the lower stages of the auditory pathway such as the outer, middle and inner ear consisting of the basilar membrane provides an ideal platform for higher levels of sound perception studies in the fields of computational neuroscience and digital signal processing. In this paper, a real-time implementation of outer and middle ear filters as well as a dual resonance nonlinear filterbank simulating the basilar membrane operations is described. The model is run on a general purpose desktop with Intel dual corePentium processor and is capable on simulating close to 180 channels concurrently.

Index Terms—real-time, gammatone, DRNL, basilar membrane, cochlea.

I. INTRODUCTION

The study of the auditory pathway (AP) is evolving with recent findings of anatomical and physiological data on humans and animals. With new available data, quantification and mathematical characterization is possiblegiving exclusive emphasis to computational models. A computational model of an AP is an emerging branch of neuroscience that provides an analytical platform, which extrapolates known physiological and psychophysical attributes governed by mathematical formulason a computer[1].

A key feature of various stages of the AP is its nonlinearity. For downstream stages, this nonlinear feature is critical especially to closely mimic physiological data findings[2]. This ensures that deviation of output in the upstream modules remain as insignificant as possible. Hence, a real-time AP model of such magnitude adds a new dimension in sound perception studies[3]. A real-time AP can also be used as a platform to gauge the attributes of implementing the AP algorithm in cochlear implants sound processors [4]. There is also the possibility of commercializing thereal-time AP in other fields as well. In telecommunications engineering, such implementation could be used for real-time noise-cancellation in mobile phone [5].

The organization of the rest of this paper is as follows: section II describes the human auditory pathway (HAP) briefly. The model simulating the HAP is described in section III. Section IV details the real-time implementation of the model and section V discusses about the implementation results.

II. THE HUMAN AUDITORY PATHWAY

A human auditory pathway (HAP) comprises of the outer, middle and inner ears. Sound waves are channeled from external sources to the ear drum via the outer ear. Sound waves in the mid-range of 2-7 kHz are strengthened based on its directional attributes and intensities. Mechanical vibrations of the ear drum are transmitted to the middle ear where the vibrations cause three small bones from within to vibrate. The first two bones known as the malleus and incus are more rigid in their motion whereas the last bone known as stapes has a larger degree of freedom in motion. The stapes is connected to an oval window on the inner ear or alternatively known as cochlea. As the inner ear has higher impedance, the middle ear functions as an acoustic impedance transformer to prevent sound waves from being reflected from the inner ear[6].

The oval window on the cochlea is used as a gateway to transmit mechanical vibrations from the middle ear to the inner ear.The inner ear is filled with viscous fluid surrounding a basilar membrane (BM). The base end of the BM connected to the oval window is thick and has a narrow width. It is susceptible to high frequency audio stimuli. The furthest point away from the BM is known as the apex point. It is thin with a broad width and is sensitive to low frequency stimuli. Mechanical vibrations of the BM use a principle of travelling waves for the excitation of inner hair cells (IHC) embedded on top of the BM throughout its length from the base to its apex end[7].

The travelling wave of a pure sine tone audio stimulus of a distinct frequency and intensity induces mechanical vibration at the base of the BM. Its amplitude grows as the wave travels to the apex end. It reaches maximum amplitude at a unique site corresponding to the stimulus frequency and subsequently decays rapidly as the wave continues. The IHC located at the peak response site

corresponding to the stimulus frequency transduces the mechanical vibrations to electrical signals, which aretransmitted upstream via the auditory nerves for further processing[8][9]. Figure 1 illustrates the anatomy of HAP and the travelling wave response of the BM.

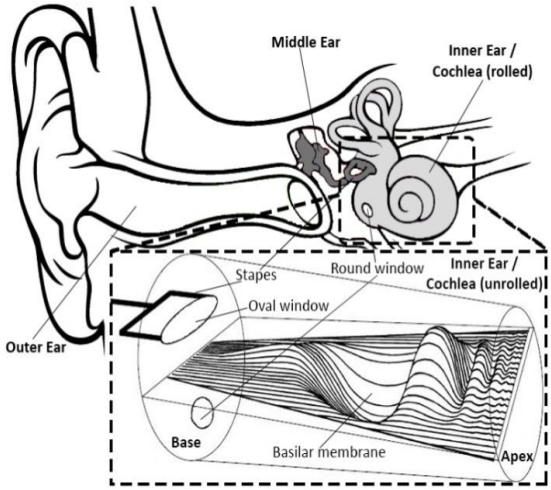


Figure 1: Human auditory pathway.

III. MODEL DESCRIPTION

A. Digital Filter

Infinite impulse response (IIR) filters are used extensively in the HAP filterbank for low pass, band pass and high pass filters. The IIR filter is also used for characterizing the gammatone filter which is essentially a bandpass filter with an exclusive bandwidth and a corresponding peak response known as best frequency (BF) that generates an output based on a unique site on the BM[10]. The IIR filterutilized isan origin of direct form 2 transposed type[11].Figure 2 displays a generalized form of the IIR filter and its equation is defined as follows:

$$y_m = \frac{1}{a_0}\left(\sum_{j=0}^{n_b-1} b_j x_{m-j} - \sum_{k=1}^{n_a-1} a_k y_{m-k}\right) \quad (1)$$

The terms n_b and n_a signify the discrete numerator and denominator order respectively. The terms b_j and a_k signify the discrete numerator and denominator coefficients that are pre-calculated before runtime based on definable settings while the term$m$ is an indexed term pointing to an exclusive data in the input and output array. The single time step delay term is represented by z as illustrated in the figure below.

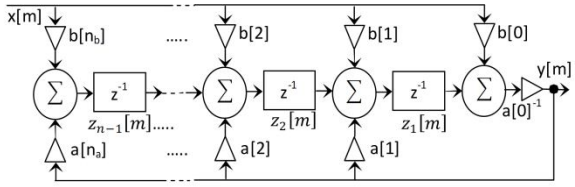


Figure 2: IIR filter with adjustable parameters.

B. Outer and Middle Ear

From a signal processing perspective, the outer and middle ear (OME) signal response can be characterized linearly and therefore be defined by a cascade of digital filters for key sub-stages[12]. Figure 3 illustrates the cascade of IIR filters used to model the propagation of sound waves in the outer and middle ear (OME).

A 1st-order Butterworth bandpass filter (BPF) is used to model the external ear resonances (EER) where sound waves to the ear drum within a range of 1-4 kHz are subjected to a 10dB gain. The output of the BPF, which is in unit Pascal, is cascaded toa 1st-order Butterworth low pass filter (LPF) with a cutoff frequency of 50 Hz. The LPF converts sound pressure level to displacement of the ear drum or tympanic membrane (TM) in meters.This parameter is streamed to a 1st-order Butterworth high pass filter (HPF) with a 1 kHz cutoff that mimics human stapes displacement in the middle ear[13][14].

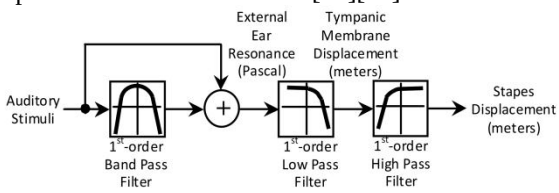


Figure 3: Outer and middle ear filters.

C. Inner Ear - Basilar Membrane

The algorithm characterizing the basilar membrane (BM) response to an auditory stimulus is a dual resonance nonlinear (DRNL) filter[13][15]. The input and output to this filter are stapes and BM displacementsin meters respectively. The filter comprises of two parallel pathways: linear and nonlinear. The linear path consists of a linear gain and three 1st-order gammatone filters. The nonlinear path contains three 1st-order gammatone filters, a memoryless compressive nonlinear function and another three 1st-order gammatone filters.The results of the two parallel paths are summed up at the end resulting in the amplitude of deflection at a specific location along the BM corresponding to a unique spectral property of an auditory stimulus. Figure 4 displays a DRNL filter.

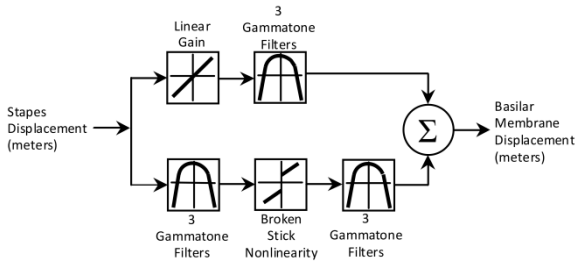


Figure 4: DRNL filter.

The compressive nonlinearity is an essential feature of the DRNL filter. The nonlinear function is defined to be compressive as its degree of nonlinearity diminishes across increasing auditory stimulus intensity. Therefore, the nonlinearity function only partially dominates the DRNL filter output for low and mid-intensity auditory stimulus. For very low and high

intensity audio stimulus, the linear path generates larger magnitude parameters as opposed to the nonlinear path. Hence, at such intensity levels the linear path dominates the DRNL output. In addition, due to its staggered implementation, the compressive nonlinearity is also known as the broken stick function. Equations 2 and 3 define the compressive nonlinear function.

$$NL = sign(x_m) \times CtBM \times \exp\left(\log\left(\frac{a}{CtBM} |x_m| \right) c \right) \quad (2)$$

$$CtBM = 10e^{-9} \times 10^{\left(\frac{CtBMdB}{20} \right)} \quad (3)$$

The term *CtBMdB* refers to the threshold in dB sound pressure level (SPL)where the nonlinearity becomes active. *CtBM* is the input to output ratio given in dB/dB. Terms *a* and *c*refer to variable gains of default values of 50,000 and 0.2 respectively[14].

As a DRNL filter represents the displacement response for a single site on a BM, multiple DRNL filters are required to model the entire BM. With such a filterbank, parallel DRNL filters with expanding bandwidths become ideal for characterizing the travelling wave phenomena from the base to the apex end of the BM. Figure 5 displays the DRNL filterbank. The DRNL filters are logarithmically spaced from one another. This log spacing interval is pre-computed before runtime on the real-time software platform. Equation 4 defines the log spacing interval as follows:

$$step = \frac{\log_{10}(BF_{\max}) - \log_{10}(BF_{\min})}{\eta_{BF} - 1} \quad (4)$$

The maximum and minimum best frequency (BF) of the DRNL filterbank is defined by BF_{max} and BF_{min} respectively. The term η_{BF} defines the total number of BF channels required for real-time processing. All three aforementioned variables are user-definable that are required to be set before pre-computation. Subsequently, log-spaced BFs are calculated. The term *i* in equation 5refers to indexed log-spaced BF.

$$BF_i = 10^{(\log_{10}(BF_{\min}) + (i \times step))} \quad (5)$$

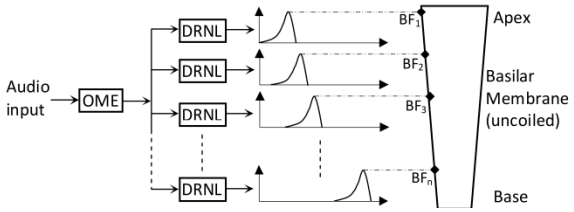


Figure 5: DRNL filterbank.

IV. REAL-TIME IMPLEMENTATION

The real-time model (RTM) of the lower stages of the HAP comprises of linear OME filters and the DRNL filterbank and will be henceforth described as RTM. It is an open source software application available online for assessment [16].

D. Processing Priority

The RTM has the characteristics of a soft real-time system where failure to meet deterministic time frames lead to degradation of computing performance, which leads to unreliable output data[17]. This is especially the issue when running a real-time software application on a general purpose operating system (GPOS) such as Microsoft Windows where the operating system (OS) allows other applications distributed time-sliced CPU attention[18]. As there is no guarantee of attaining a deterministic CPU timeand the failure to meet timing constrains does not amount to critical failure, it is justifiable to label the RTM as a soft real-time system.In order to attain more CPU time, Windows OS allows process prioritization. The RTM is run as a process on Windows OS where its priority is able to be altered before runtime allowing a degree of adaptable runtime loading control.

E. General Design

The RTM is developed in C++ and uses Microsoft DirectSound to acquire audio from the onboard microphone channel the model is run on. A separate input source in the form of a software sine tone generator is implemented and uses the DirectSound audio frame to stream fragmented pure sine tone to the model. Audio data is streamed at a rate of 22.05 kHz. This is achieved by the availability of a window of 1280 data samples every 58ms at runtime.

The time frame of 58ms defines the computing boundary where all output parameters corresponding to input from the current window of interest is required to be available. Failure to meet such a requisition leads to degradation of outputs from subsequent windows processing due to buffer overrun, which is typical of extensive BF channel loading.

Variables such as filter coefficients in the model are precomputed based on user settings.Management of all buffers in terms of de-allocation and allocation are performed before runtime as well. This reduces runtime redundancy of managing computing resources and allows the RTM model to compute output parameters for every cascaded stage, thereby optimizing the RTM. The definition of optimization refers to a higher output resolution of the concurrent time-frequency signals by increased parallel BF channel loadingof the DRNL filterbank.

F. PastIO Parameters Buffering

As part of real-time processing, IO data streaming are required to be fragmented to window frames. The IIR filter relies on past IO values for computation of outputs and these are readily available from the IO buffer. However, the IO buffers are refreshed with new data for every window frame and the continuity of output data computation is compromised due to the absence of past IO data from the preceding frame. Therefore, additional buffers are required to be allocated to store IO data from preceding frames to ensure that the integrity of output data computational continuity is maintained.

Figure 6 displays the OME algorithm implemented

for three window frames of IO data. Each parallel branch represents the three serially cascaded IIR filters from outer ear to middle ear described in section III-B. The first window branch, W1 represents the initial processing of data when the algorithm is initiated. In this phase, the algorithm processes input data and initialized parameters. For the second window, W2 onwards, the algorithm additionally relies on IO parameters from preceding windows. Hence past IO data (PIOD) buffers are allocated for the three IIR filter stages. These buffersare illustrated by the two additional blocks above and below IIR filter blocks in every branch. The contents of PIOD buffers are overwritten at the computingconclusion of every window frame.

The texts beside the blocks in figure 6 denote the source of PIOD. It follows the following format: IO-Filter-Window. "IO" is designated by *x* and *y* for input and output respectively. "Filter" represents the filter stage in the branch and "Window"denotes the window frame number.

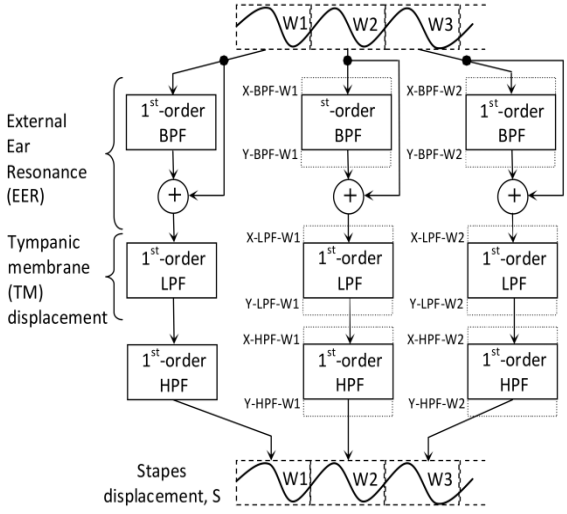


Figure 6: Outer and middle ear algorithm for three contiguously streamed windows of segmented audio data.

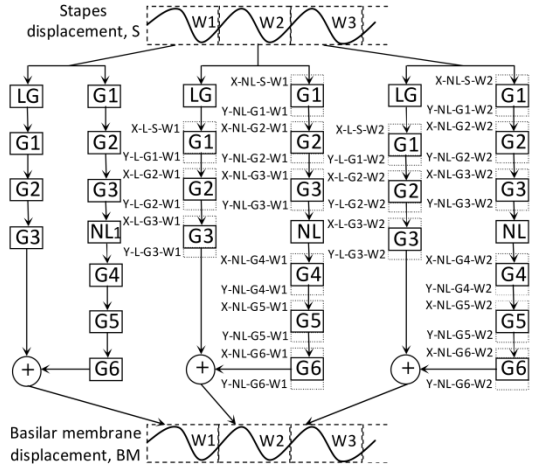


Figure 7: DRNL filter algorithm for 1 BF channel processing three window frames of audio data.

The DRNL filter follows the same setup as the OME filters. For every gammatone filter, two PIOD buffers are allocated for the input and output of a filter stage respectively. Therefore for all nine gammatone filters in a DRNL filter, eighteen PIOD buffers are allocated. A DRNL filterbank having multiple DRNL filters will likewise require a multiple of eighteen PIOD buffers. As an example, if the DRNL filterbank contains thirty BF filter channels, then five hundred and forty PIOD buffers are allocated.

Figure 7 displays a single DRNL filter PIOD buffering for three window frames of IO data. The first window branch, W1 shows the gammatone filters processing at startup which relies on IO data from the frame itself as well as initialized parameters which remain constant during processing. Subsequent windows additionally rely on PIOD, which are denoted by two blocks hemming the gammatone filter blocks "G1" to "G6" on both the linear and nonlinear pathways. The texts accompanying these PIOD blocks describe the buffered data origins as follow: IO-Path-Filter-Window. "Path" is signified by either "L" or "NL" representing linear and nonlinear cascaded algorithm pathway respectively. The other parameters follow the same convention described in the former paragraph on OME PIOD buffering. The block denoted by "LG" and "NL" represent memoryless linear gains and compressive nonlinearity respectively.

V. RESULTS

Figure 8 displays the normalized root-means-square (RMS) error between the original and real-time models for a 3 kHz sine tone input within a range of 10 to 90 dB SPL. A single DRNL filter with a BF response close to 3 kHz was used to acquire the responses. The RMS errors for both stapes and BM displacements remain significantly below 1% and remain a testament to the accurate implementation of the filter in the RTM. The overall RMS error for stapes displacement is observed to be ideally lower than BM displacement due to its early placement in the filter stage and the lower number of IIR filters present in its algorithm.

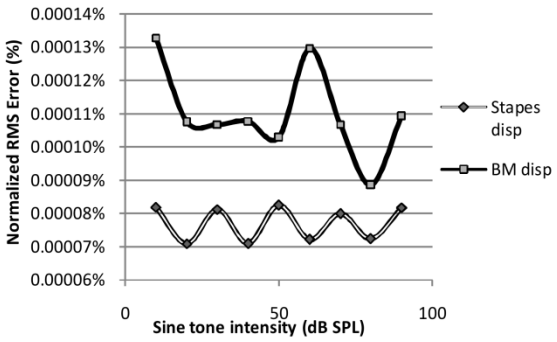


Figure 8: Normalized RMS error between real-time and original model.

Figure 9 displays the model response for an auditory stimulus of a 3 kHz sine tone from an intensity range

of 30 to 85 dB SPL. Figure 9a illustrates the model response based on original data from Meddis[13] while figure 9b displays the response of the RTM.For low intensity stimulus level at 30 dB SPL, the nonlinear path dominates the DRNL filter output with a peak response close to 3 kHz. Phase shifts for the peak response begin to appear from 40 dB SPL and beyond indicating the decreasing dominance of the nonlinear pathway. Beyond 80 dB SPL, the linear pathway completely dominates the DRNL response as well as significant phase shifted peak response. In comparison, the RTM is able to regenerate the linear and nonlinear responses of the DRNL filter with insignificant deviations for various intensities. Furthermore, it is also able to mimic the phase shift of the BF from low to high intensities identical to the original model.

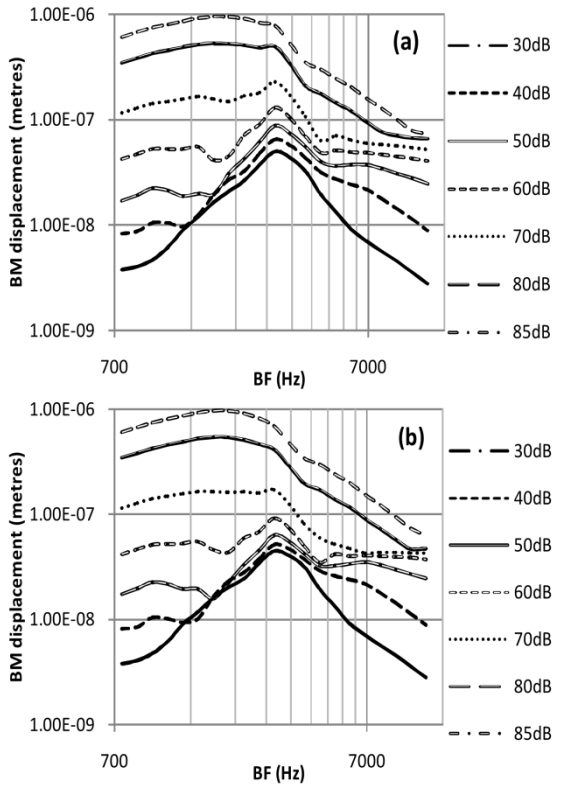


Figure 9: (a) Original data from Meddis[13]. (b) Real-time algorithm response.

Figures 8 and 9 have so far displayed RTM responses for only a single DRNL filter corresponding to one BF channel. Figure 10 displays the DRNL filterbank response for 30 BF channels employed in the RTM. The input stimulus is a 500 Hz sine tone with an intensity of 50 dB SPL. Observing the graph in figure 10, the amplitude of the wave is most significant in the range above 388 Hz and below 670 Hz, which is in close proximity to the auditory input frequency of 500 Hz. Another indication of the accurate implementation of the DRNL filterbank is the effect of PIOD buffering. A vertical dotted line marks the boundary between adjacent window frames. The smooth continuity of the wave plots beyond 58ms is a key indicator of PIOD buffering for inter-window frame transit computing.

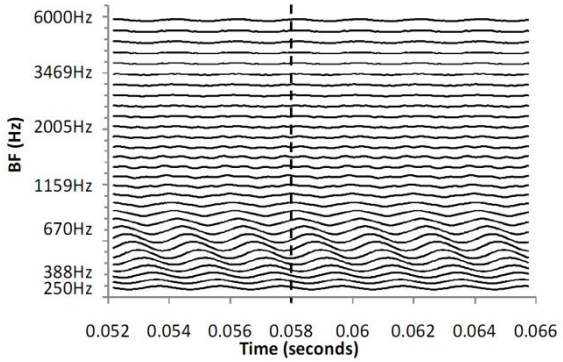


Figure 10: DRNL filterbank response for 30 BF channels.

Figure 11 illustrates the maximum number of BF channels that can be accommodated by a general desktop equipped with a dual-core processor within time frames of 58ms. The RTM is run under different process priority on Windows OS. Except for the default background processes determined by Windows OS, no other applications were executed concurrently with the RTM. Under "real-time" priority, the RTM has the largest BF channel loading due to the longest CPU time exposure.This is because the RTM is given precedence over all other processes including those belonging to Windows OS kernel[19]. This is contrary to running the RTM under "Normal" priority where BF channel loading is at its lowest. This is due to the RTM acquiring equivalent CPU time slices with other background processes running under the same priority. Running the RTM under "Above Normal" and "High" priority produced equal BF channel loading though the loading were higher than running RTM under "Normal" priority. This is mainly because Windows OS assigns larger CPU time to the RTM as compared to processes running under "Normal" priority. However, the relinquishment of CPU is still required so as to run other processes of various priorities including the essential high priority OS kernel.

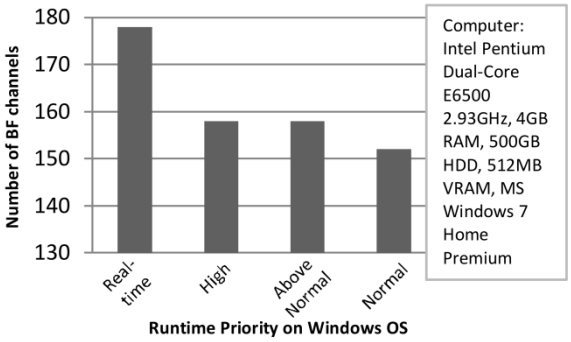


Figure 11: Maximum load profile for the real-time model (RTM).

CONCLUSION

A real-time model encapsulating signal processing attributes of the outer, middle and inner ear of a human auditory pathway has been described. The nonlinear

response of the inner ear stage defined bythe basilar membrane is characterized by a dual resonance nonlinear filterbank. The model is able to simulate very accurate responses based on physiological findings. It is able to run on different priority on Windows operating system and provides dynamic computational loading.

Due to the dynamism of such a model, various features can be added to enhance its response range. Firstly, this model can be implemented on a real-time operating system (RTOS) where it will be able to harness ideal real-time capabilities and possibly increase its BF channel loading as compared to running on a GPOS. Furthermore, this model does not consider the directional attributes of audio to the ear. Adding such a feature would enhance the pedigree of sound perception especially when upstream modules are integrated to the model.

REFERENCES

[1] R. Meddis and E. a Lopez-poveda, "Overview," in *Computational Models of the Auditory System*, R. Meddis, E. A. Lopez-Poveda, R. R. Fay, and A. N. Popper, Eds. Springer, 2010, pp. 1–6.

[2] C. J. Plack and A. J. Oxenham, "Basilar-membrane nonlinearity estimated by pulsation threshold," *J. Acoust. Soc. Am.*, vol. 107, no. 1, pp. 501–507, 2000.

[3] X. Zhang, M. G. Heinz, I. C. Bruce, and L. H. Carney, "A Phenomenological Model for the Responses of Auditory-Nerve Fibers: I . Nonlinear Tuning," *J. Acoust. Soc. Am.*, vol. 109, no. 2, pp. 648–670, 2001.

[4] B. S. Wilson, E. A. Lopez-Poveda, and R. Schatzer, "Use of Auditory Models in Developing Coding Strategies for Cochlear Implants," in *Computational Models of the Auditory System*, R. Meddis, E. A. Lopez-Poveda, R. R. Fay, and A. N. Popper, Eds. Springer, 2010, pp. 237–260.

[5] L. Watts, "Real-Time , High-Resolution Simulation of the Auditory Pathway , with Application to Cell-Phone Noise Reduction," *Measurement*, pp. 3821–3824, 2010.

[6] J. Pickles, "The Outer and Middle Ears," in *An Introduction to the Physiology of Hearing*, 3rd ed., Emerald Group, 2008, pp. 11–24.

[7] R. A. Barker, S. Barasi, and M. A. Neal, "Part 2 - 24 - The auditory system I: The ear and cochlea," in *Neuroscience at a Glance*, Oxford: Blackwell Science, 1999, pp. 62–63.

[8] J. Pickles, "The Cochlea," in *An Introduction to the Physiology of Hearing2*, 3rd ed., Emerald Group, 2008, pp. 25–72.

[9] G. G. Matthews, "Hearing and Other Vibration Senses," in *Neurobiology Molecules, Cells and Systems*, 1st ed., Oxford: Blackwell Science, 1998, pp. 433–457.

[10] A. G. Katsiamis, E. M. Drakakis, and R. F. Lyon, "Practical gammatone-like filters for auditory processing," *Eurasip J. Audio, Speech, Music Process.*, vol. 2007, pp. 1–13, 2007.

[11] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 1999.

[12] R. Meddis and E. A. Lopez-Poveda, "Auditory Periphery: From Pinna to Auditory Nerve," in *Computational Models of the Auditory System*, R. Meddis, E. A. Lopez-Poveda, R. R. Fay, and A. N. Popper, Eds. New York Dordrecht Heidelberg London: Springer, 2010, pp. 7–38.

[13] E. A. Lopez-Poveda and R. Meddis, "A Human Nonlinear Cochlear Filterbank," *J. Acoust. Soc. Am.*, vol. 110, no. 6, pp. 3107 – 3118, 2001.

[14] R. Meddis, "MAP 1_14 Model description." Meddis, Ray - University of Essex, Colchester, pp. 1–32, 2012.

[15] R. Meddis, L. P. O'Mard, and E. A. Lopez-Poveda, "A Computational Algorithm for Computing Nonlinear Auditory Frequency Selectivity," *J. Acoust. Soc. Am.*, vol. 109, no. 6, pp. 2852 – 2861, 2001.

[16] R. K. Singh, "RTAP," 2014. [Online]. Available: https://code.google.com/p/rtap/. [Accessed: 17-Feb-2015].

[17] P. A. Laplante, "Basic Real-Time Concepts," in *Real-Time Systems Design and Analysis*, 3rd ed., IEEE Press,Wiley-Interscience, 2004, pp. 1–21.

[18] K. Ramamritham, C. Shen, O. Gonzalez, S. Sen, and S. B. Shirgurkar, "Using Windows NT for Real-Time Applications: Experimental Observations and Recommendations," *IEEE Real-Time Technol. Appl. Symp.*, pp. 1 – 13, 1998.

[19] S. Akhter and J. Roberts, *Multi-Core Programming: Increasing Performance through Software Multi-threading.*, 1st ed. Hillsboro: Intel Press, 2006, pp. 1–336.

★★★