,,

COCHLEAR IMPLANT SIMULATOR AUDIO PLUGIN FOR UNITY

Carmen Muñoz
Aalborg University
mailto:cmunoz19@student.aau.dk

ABSTRACT

In this paper we propose a cochlear implant simulator which works in real-time. The simulator allows listeners with average hearing abilities to hear as if they had a cochlear implant. It is focused to be an audio plugin for Unity in order to save time as it facilitates getting real time outcomes for a developer team when designing serious games for CI users, without the necessity having an impaired person to test. Additionally, it bypasses part of the unpleasantness caused on CI users testing being it even larger when addressing children and their families.

1. BACKGROUND AND LITERATURE REVIEW

To determine precisely how sound is perceived from a CI user is extremely complicated due to the several different possible CI configurations (designated compression, codification strategy etc.) apart from features of the surgery, physiological factors. . . etc. To acquaint how patients perceive sound, there are several testimonies with the inconvenient of them being are affected by subjectivity. Through validation tests to evaluate quality audio aspects from both, CI and normal hearing people, certain additional information can be obtained [1]. Taking these reports into account, loss of information associated with the analysis process (due to the CI processor, implant and codifying strategy) can be represented through signal processing techniques. Regarding previous CI simulator implementations, they are mostly written in Matlab, but also other languages like C++ [2] with the possibility of creating a .exe or .dmg file (Mac and Windows). Also, there is an app for iOS devices [3].

Depending on the depth of its development, they are able to parametrize a great quantity of CI features such as denoising, frequency range, envelope strategy, electroacoustic stimulation which suppose huge advances. However, most approaches are not real time implementations [2,4,5] with the exception of the iOS app. Nevertheless, it has the disadvantage of offering a more basic configuration than the others.

These simulators use quite a few distinct approaches as discussed subsequently. A general agreed model that most attempts for CI simulators [6, 6–15] implementation is in Figure 1.

This blueprint is rarely slightly changed. Then, regarding this starting point, we consequently will focus on each block.

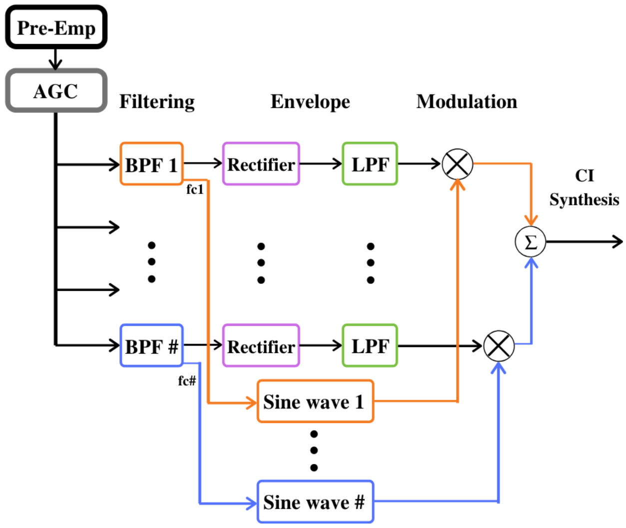

Figure 1. Scheme of the main algorithm in MATLAB.

1.1 2.1 Pre-emphasis Filter Impairment

A pre-emphasis high pass filter is used to emphasize and increase the gain in the high frequency components in the signal [6, 9] and has been implemented in a variety of research interfaces [16]. This desired specific frequency enhancement is due to the auditive human frequency response that the well-known Fletcher-Munson curves models [17]. Although voice fundamental frequency can fall in lower frequency components, the consecutive formants and enough of the harmonic series can be found for the missing fundamental that recreates the impression of hearing the fundamental tone [8, 18]. The technical approach of addressing this problem is common for some researches that solves this issue by implementing a low-pass below 1.200Hz, 6dB/octave [1, 17, 19]. Other proposed methods are the Nucleus® CI which uses a slope 6dB/octave and cutoff frequency at 4000Hz [20] or the CI by Advanced Bionics, which uses a two cascaded first order high-pass filters with fc1= 100 and fc2=1500Hz reducing 6dB/octave [21]. An approach for this last strategy is selected for the simulator.

Last observation that is worth mentioning about pre-processing is that some CI also consider the optimization of the input dynamic range to input signals levels and even, some systems that make use of several microphones, exploits level

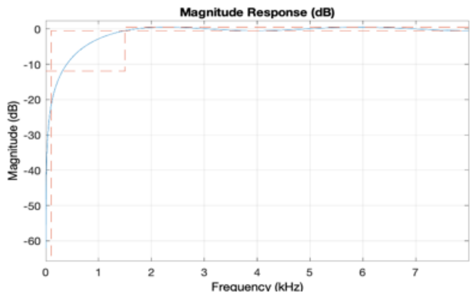

Figure 2. Magnitude response of LPF in Pre-emphasis.

signal differences to enhance desired sounds [22] and are able to perform noise-reduction [21]. This last point is out of the scope of the simulator proposed.

1.2 Active Gain Control

AGC prevents overloading or distortion from happening. A fast-acting infinite compressor is applied after a certain level to avoid dangerous ear damage [20]. It operates exclusively on the frequency preserving the temporal pattern of the speech waveform [9].

1.3 Band Pass Filters Design

To simulate the stimulation of the human cochlea, a bank of band pass filters is constructed to resemble the auditory system. When it comes to BPFs, dissimilar, numerous and "hard-coded" values with hardly an explanation of these are deployed. For example, one approach stablishes the filter bands in Nucleus CI to be spaced linearly from 188 to 1312Hz and logarithmic afterwards [9]. Another strategy for four channels CI design, (each CI channel is modeled by a BPF) states that the frequencies defining the bands are 300 Hz, 713 Hz, 1509 Hz, 3043 Hz, and 6000 Hz [23]. Many more researches state "hard-core" values for these BPFs [19, 24] or do not specify why/how this is addressed [25]. The strategy chosen for the BPFs is the following:

Electable frequency edges (ranges):
· 300-3400Hz [6] which is the frequency range used is the voice channel.
· 20-6KHz (speech signal) [7] .
· 100 - 8000Hz [22], which includes the formants F0/1/2/3 frequency range [26] and even goes beyond the voice spectrum. It is of interest when listening to signals differing from voice such as music.
· 100-16.000Hz, a range used in some CI experiments [27].

Number of BPF
It equals the number of CI channels (electrodes in the array). Can be selected from a minimum of 2 to 24, as maximum number electrodes in the current CI market [28].

Central frequencies
Computed by dividing the subtraction of frequency edges by BPF after its translation to Bark scale Equation (1) [29].

$$
\begin{aligned}
&bark = \frac{(26.81)(hz)}{1960 + hz} - 0.53 \\
&if : bark < 2 \rightarrow bark = bark + (0.15)(2 - bark) \\
&if : bark > 20.1 \rightarrow bark = bark + (0.22)(bark - 20.1)
\end{aligned} \tag{1}
$$

Filter type and order Butterworth of sixth order [19, 24, 30] is the chosen election. It is worth stressing that gammatone filter type is used to describe cochlea simulation [31], however, by lack of functions descriptors for FAUST language, it is not elected.
Filters Bandwidth When it comes to filter bandwidth, once more, this is assessed in different manners, using equivalent rectangular bandwidth [25], greenwood scale [32], and Bark [9, 10] scale among others. For this approach, bandwidths are computed using Bark scale. Among consecutive filters, there is partially an overlapping frequency response which widens when increasing frequency [22, 24].

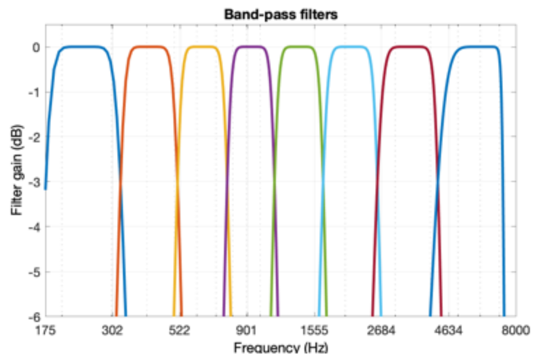

Figure 3. Band pass filters amplitude response.

1.4 Envelope

Formants are peaks in the spectral envelope that corresponds to resonances of the vocal tract. The auditory system uses them to identify sounds such as vowels, quite important for speech recognition [22]. Thus, to compute the spectral envelope, a Hilbert transformation followed by low-pass filtering is proposed by [22]. Some others practice a half or full-wave rectification always ended by a LPF [13, 15, 16, 19, 33].
Eventually, a full-wave rectifier (peak detector [13]) followed by a LPF with fc=160Hz and 4rd order are implemented to function as envelope extractor.

1.5 Modulation

The envelope of each channel is used to modulate an excitation signal. It can be a noise signal or a tone signal. Afterwards, all modulated signals (as many as number of channels) are summed across to yield the original audio signal [32].

1.5.1 Noise

Band passed white noise is used to modulate its corresponding filtered envelope.

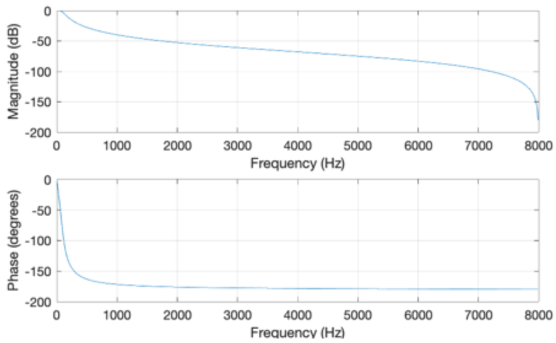

Figure 4. LPF for envelope extraction.

1.5.2 *Tone/sine modulation*

A sine wave with the same central frequency band as the band pass filter is used to modulate each corresponding filtered envelope [7].

2. DESIGN AND IMPLEMENTATION

2.1 MATLAB

Firstly, the audio signal is resampled to 16kHz which is the CI working frequency [9] and converted to mono. The continued code is written following the selected aforementioned parameters for each one of the previous algorithm blocks. Because of the loss of energy through the various processes, a re-computation of the original RMS of the signal is computed at the end of the algorithm.

2.2 3.2 FAUST

MATLAB code is translated to FAUST supplied by documented functions of its libraries. A slight setback is that FAUST filter-bank and other parameters cannot be set during real time. To deal with it, a FAUST code must be generated for each one of the desired parameters (channels, frequency range…). A vast drawback currently is that FAUST cannot implement convolution of two signals. In its manual webpage it is said that a hard work is being carried out to solve this issue as soon as possible. Until so, multiplication operation is used notwithstanding of this found limitation.

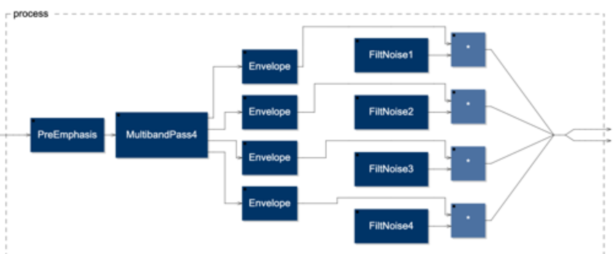

Figure 5. FAUST block design of the algorithm.

2.3 3.3 Unity

The unitypackage file is exported from FAUST web (https://faust.grame. The unity package is uploaded to unity generating a script which is applied to a game object. The sounds this game object plays will be filtered to be heard as a CI user would hear. A practical demonstration video can be found here [34].

3. RESULTS AND DISCUSSION

As previously explained, it is complex to get a precise objective CI sound reference. Some simulated examples are on the web, [35], yet they are not well documented or does not offer origin reliability [36]. To keep a trustworthy equation, the CI simulator performances (MATLAB and FAUST) were compared to the simulator designed by Granada University [2]. These are the outputted spectrograms using 4 channels with noise carrier on a sample phrase:

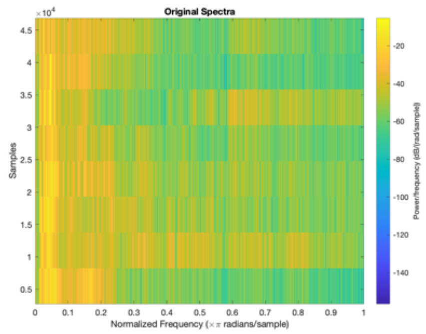

Figure 6. Original spectra of the used of the sample phrase

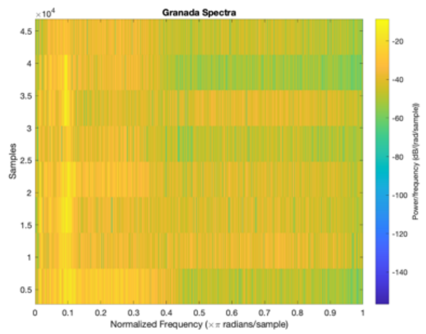

Figure 7. Generated spectra from Granada CI simulator.

From these spectra results, we can derive that MATLAB implementation resembles the results by Granada CI simulator, however, they are not exactly the same as there are some parameters of importance (i.e.: calculation of band-pass filter characteristic) that are not described or cannot be modelled in Granada CI ergo not guarantying totally comparable results. Faust results are quite dissimilar of what it is expected from a CI simulator. The produced audio files can be accessed through this link.

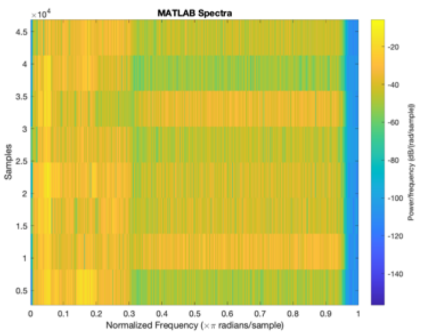

Figure 8. Generated spectra from MATLAB simulator.

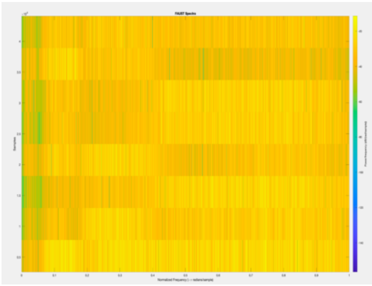

Figure 9. Generated spectra from FAUST CI simulator

4. CONCLUSION

FAUST CI simulator approximation is yet far from what expected because of its convolutional limitations. It sounds more like filtered noise rather than a modulation or a vocoder, like Granada simulator does, still intelligible. Matlab approach resembles more a vocoder, yet has still disturbing noise compared to Granada simulation. Again, it must be said that some parameters are unknown when working with Granada approach. Even though, this convolution constraint in FAUST is hoped to be solved soon by developers making possible a successful multi-platform CI simulator as proved with MATLAB's approximation performance.

5. REFERENCES

[1] M. F. Dorman, P. C. Loizou, and D. Rainey, "Speech intelligibility as a function of the number of channels of stimulation for signal processors using sine-wave and noise-band outputs," *The Journal of the Acoustical Society of America*, vol. 102, no. 4, pp. 2403–2411, 1997.

[2] Á. de la Torre Vega, M. Martí, R. de la Torre Vega, and M. S. Quevedo, "Cochlear implant simulation version 2.0: Description and usage of the program," *University of Granada, Spain*, 2004.

[3] "Impaired hearing simulation (ihsim)." Eric Tar. [Online]. Available: https://appadvice.com/app/ihsim/1421965944

[4] "Cochlear implant speech processor." Mathworks. [Online]. Available: https://se.mathworks.com/help/dsp/examples/cochlear-implant-speech-processor.html

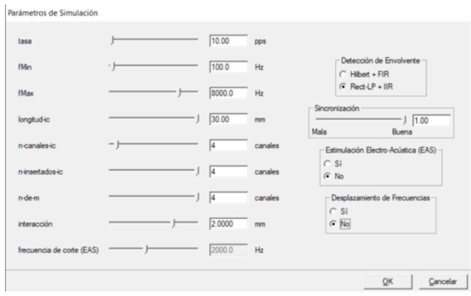

Figure 10. Configuration for Granada CI simulator

[5] "Cochlear implant simulation." University of Dallas. [Online]. Available: https://se.mathworks.com/matlabcentral/fileexchange/69403-cochlear-implant-simulation

[6] H. Zhao and S. Mai, "Simulation of asic/fpga cochlear implant speech processor with simulink and modelsim cosimulation method," in *2013 IEEE International Conference of IEEE Region 10 (TENCON 2013)*. IEEE, 2013, pp. 1–4.

[7] L. Wang, L. Zhou, Y. Xu, and Z. Xiao, "Computer simulation of multichannel cis strategy for cochlear implant," in *2009 3rd International Conference on Bioinformatics and Biomedical Engineering*. IEEE, 2009, pp. 1–4.

[8] B. S. Wilson, C. C. Finley, D. T. Lawson, R. D. Wolford, D. K. Eddington, and W. M. Rabinowitz, "Better speech recognition with cochlear implants," *Nature*, vol. 352, no. 6332, p. 236, 1991.

[9] A. Komal, R. Dowell, and P. Dawson, "Cochlear implant stimulation rates and speech perception," 2012.

[10] N. Kaibao, L. Ju *et al.*, "Implementation of cis speech signal processing for cochlear implants by using bark scale frequency band partition," in *8th International Conference on Neural Information Processing (ICONIP 2001)()*. ; , 2001, pp. 755–759.

[11] P. C. Loizou, "Speech processing in vocoder-centric cochlear implants," in *Cochlear and brainstem implants*. Karger Publishers, 2006, vol. 64, pp. 109–143.

[12] ——, "Mimicking the human ear," *IEEE signal processing magazine*, vol. 15, no. 5, pp. 101–130, 1998.

[13] E. D. Marsman, R. M. Senger, G. A. Carichner, S. Kubba, M. S. McCorquodale, and R. B. Brown, "Dsp architecture for cochlear implants," in *2006 IEEE International Symposium on Circuits and Systems*. IEEE, 2006, pp. 4–pp.

[14] B. S. Wilson and M. F. Dorman, "Cochlear implants: current designs and future 2008.

[15] R. V. Shannon, F.-G. Zeng, V. Kamath, J. Wygonski, and M. Ekelid, "Speech recognition with primarily temporal cues," *Science*, vol. 270, no. 5234, pp. 303–304, 1995.

[16] A. R. Kaiser and M. A. Svirsky, "A real time pc based cochlear implant speech processor with an interface to the nucleus 22 electrode cochlear implant and a filtered noiseband simulation," *Research on spoken language processing: Progress report*, no. 23, pp. 417–428, 1999.

[17] H. Fletcher and W. A. Munson, "Loudness, its definition, measurement and calculation," *Bell System Technical Journal*, vol. 12, no. 4, pp. 377–430, 1933.

[18] R. J. Baken and R. F. Orlikoff, "Clinical measurement of speech and voice. london," *Cengage Learning*, pp. 561–570, 2000.

[19] P. C. Loizou, M. F. Dorman, Z. Tu, and J. Fitzke, "Recognition of sentences in noise by normal-hearing listeners using simulations of speak-type cochlear implant signal processors," *Annals of Otology, Rhinology & Laryngology*, vol. 109, no. 12_suppl, pp. 67–68, 2000.

[20] J. F. Patrick, P. A. Busby, and P. J. Gibson, "The development of the nucleus® freedom™ cochlear implant system," *Trends in amplification*, vol. 10, no. 4, pp. 175–200, 2006.

[21] K. Chung and N. McKibben, "Microphone directionality, pre-emphasis filter, and wind noise in cochlear implants," *Journal of the American Academy of Audiology*, vol. 22, no. 9, pp. 586–600, 2011.

[22] J. Wouters, H. J. McDermott, and T. Francart, "Sound coding in cochlear implants: From electric pulses to hearing," *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 67–80, 2015.

[23] Q.-J. Fu and R. V. Shannon, "Frequency mapping in cochlear implants," *Ear and hearing*, vol. 23, no. 4, pp. 339–348, 2002.

[24] M. F. Dorman, P. C. Loizou, J. Fitzke, and Z. Tu, "The recognition of sentences in noise by normal-hearing listeners using simulations of cochlear-implant signal processors with 6–20 channels," *The Journal of the Acoustical Society of America*, vol. 104, no. 6, pp. 3583–3585, 1998.

[25] F. Chen, "Predicting the intelligibility of cochlear-implant vocoded speech from objective quality measure," *J. Med. Biol. Eng*, vol. 32, no. 3, pp. 189–194, 2012.

[26] F.-G. Zeng, S. Rebscher, W. Harrison, X. Sun, and H. Feng, "Cochlear implants: system design, integration, and evaluation," *IEEE reviews in biomedical engineering*, vol. 1, pp. 115–142, 2008.

[27] M. J. Goupell, B. Laback, P. Majdak, and W.-D. Baumgartner, "Effects of upper-frequency boundary and spectral warping on speech intelligibility in electrical stimulation," *The Journal of the Acoustical Society of America*, vol. 123, no. 4, pp. 2295–2309, 2008.

[28] A. Dhanasingh and C. Jolly, "An overview of cochlear implant electrode array designs," *Hearing research*, vol. 356, pp. 93–103, 2017.

[29] H. Traunmüller, "Analytical expressions for the tonotopic sensory scale," *The Journal of the Acoustical Society of America*, vol. 88, no. 1, pp. 97–100, 1990.

[30] F.-G. Zeng, "Trends in cochlear implants," *Trends in amplification*, vol. 8, no. 1, pp. 1–34, 2004.

[31] M. Slaney *et al.*, "An efficient implementation of the patterson-holdsworth auditory filter bank," *Apple Computer, Perception Group, Tech. Rep*, vol. 35, no. 8, 1993.

[32] N. El Boghdady, A. Kegel, W. K. Lai, and N. Dillier, "A neural-based vocoder implementation for evaluating cochlear implant coding strategies," *Hearing research*, vol. 333, pp. 136–149, 2016.

[33] L. M. Friesen, R. V. Shannon, D. Baskent, and X. Wang, "Speech recognition in noise as a function of the number of spectral channels: Comparison of acoustic hearing and cochlear implants," *The Journal of the Acoustical Society of America*, vol. 110, no. 2, pp. 1150–1163, 2001.

[34] "Run faust plugin on unity." Self-produced video. [Online]. Available: https://www.youtube.com/watch?v=nfs0yhr3l1c&t=26s

[35] "Arizona state university, hearing cochlear implants w/ open captions." Arizona state University. [Online]. Available: https://www.youtube.com/watch?v=00WOao4kpwM

[36] "Ab: Cochlear implant simulation: Advanced bionics." AB. [Online]. Available: https://www.youtube.com/watch?v=eo-HNAoCzjw