SOFTWARE DEVELOPMENT FOR RIRS PROCESSING

Gastón Gabriel Caminos
Universidad Nacional de Tres de Febrero, Ingeniería de Sonido, Caseros, Buenos Aires, Argentina
email: caminos47501@estudiantes.untref.edu.ar

Martín Nocito
Universidad Nacional de Tres de Febrero, Ingeniería de Sonido, Caseros, Buenos Aires, Argentina
email: mnocito8@gmail.com

Abstract

This study presents a Python room impulse response processing software implementation. The software is designed to be user-friendly, especially for users with little to no programming experience. The developed program is capable of obtaining room parameters such as RT20, RT30, EDT, Tt, EDTt, C50, C80, and IACC from a logarithmic sine sweep room response measurement. The results were compared to popular software Aurora Acoustical Parameters and it's concluded that the developed software is reliable revealing a difference that does not exceed 5% in the calculation of temporal parameters within the valid frequency bands.

Keywords: Acoustics, Room Impulse Response (RIR), ISO 3382.

1. Introduction

Speech intelligibility and music perception are always influenced by room acoustics, therefore knowing it's influence and modifying it has turned the acoustics into another physics discipline since the past century. The acoustic characteristics of a room can be described by it's reverberation time, clarity and inter-aural parameters among others, and they can be obtained from the room's impulse response (RIR), based on the assumption that an enclosure is a linear time-invariant system. Various methods have been developed in the course of the years to obtain RIRs, evolving along with electronics and digital techniques.

In this study, a Python developed RIR processing software is presented, providing the capability to obtain various acoustic parameters from an impulse response, generated by the sine sweep recording method. The user is prompted to load the sine sweep file (or the used sweep information) and the recording file, and after choosing diferent options towards the processing technique, the results are shown in graphs and tables, as will be described in further sections. The software is available at `https://github.com/martin-nct/RIR-processing`.

2. State of Art

Reverberation time was first measured and defined by Wallace Clement Sabine using a stopwatch and an organ pipe (providing excitation in different frequencies) in the late 19th century. Other sound sources were also used to generate the decay, such balloon bursts and blank shotguns.

In 1964, Manfred Schroeder presented a new method for obtaining the energy time curve (ETC) based on the reverse integration of the recorded band filtered noise decay (interrupted noise method), using tape recording technologies and electronic integration (capacitance) [1]. This method is equivalent to the average of infinite decay curves, so the time involved in the reverberation time measurement was reduced significantly.

With the advance of technology, the Maximum Length Sequence (MLS) excitation method was widely used because of its white spectrum, so that its cross-correlation with the recorded signal is proportional to the system impulse response. Recently, Angelo Farina developed the logarithmic swept sine method, providing better signal-to-noise ratios than other techniques, further reducing the measurement time and most importantly achieving the discrimination of the nonlinear distortion, inherent to the sound source, from the linear system's impulse response [2]. Nowadays, this method is widely used in Farina's Aurora package plugin implementation.

3. Theoretical Framework

3.1 Room Impulse Response (RIR)

Any enclosure subjected to sound pressure levels up to 120 dB will be treated as linear since it is assumed that sound pressure levels up to 120 dB will not generate significant distortions and non-linearities in the system response. This is the reason why most rooms are regarded as linear time-invariant (LTI) systems, as it can be difficult to achieve a sound pressure level of 120 dB with typical systems. An essential characteristic of LTI systems is that their output is obtained through the convolution of the input signal with the impulse response, denoted as $h(t)$. This signal type fully characterizes the given LTI system, encompassing information about all the acoustic parameters contained within it.

3.1.1 Logarithmic sine sweep

The logarithmic sine-sweep method, introduced by Farina [2], aims to capture the impulse response of an enclosure. This technique offers numerous advantages over more traditional methods such as Maximum-Length Sequence (MLS), balloon popping, hand claps, and paper-poppers. One of the main benefits of this method is its reproducibility. In other words, stimuli generated within the specified parameters will be identical. This reproducibility is attributed to the method relying solely on the characteristics of the loudspeaker system, guaranteeing an impulsive stimulus. Consequently, the method produces a delta function through the convolution of an inverse filter and a logarithmic sine-sweep, resulting in precise and consistent measurements of the enclosure's impulse response.

3.2 Filtering methods

3.2.1 Hilbert Transform

The Hilbert Transform is a mathematical tool used in signal processing to analyze the phase and amplitude of a time-domain signal. It computes the analytic signal, which is a complex representation of the original signal preserving phase and amplitude information. It involves convolving the input signal with the Hilbert kernel, resulting in an analytic signal with a 90-degree phase shift from the original signal [3]. The Hilbert Transform is the first process applied in order to obtain the decay curve of the IR.

3.2.2 Savitzky-Golay filter

The Savitzky-Golay filter is a signal processing method used to smooth and enhance the quality of a signal. It was developed by Abraham Savitzky and Marcel J. E. Golay in the 1960s [4].

Unlike other filtering methods such as moving average or Butterworth filters, the Savitzky-Golay filter utilizes a least-squares fitting technique to estimate the smoothed values of a signal based on a window of neighboring samples.

The Savitzky-Golay filter is particularly well-suited for signals that contain random noise or rapid variations, as it can effectively smooth the signal without introducing significant distortion to important signal features.

One advantage of the Savitzky-Golay filter is its ability to preserve signal details and important characteristics, such as peaks and rapid changes while reducing unwanted noise. The length of the window and the degree of the polynomial used in the fitting can be adjusted to accommodate specific signal characteristics and analysis requirements.

In summary, the Savitzky-Golay filter is a widely used filtering method for signal smoothing and noise reduction, while preserving important signal features. It finds applications in various fields, including biomedical signal processing, data analysis, and spectrum analysis, among others.

3.2.3 Moving Average Filter (MAF)

The Moving Average Filter (MAF) involves taking a time window of N samples from the signal, calculating the average of that sample set, and assigning the average value to that portion of the signal. As the window is temporally shifted along the entire signal, the filtering process is applied. This filer is widely used in time domain signals to reduce random noise preserving the signal shape. Its implementation is easy and fast because recursive techniques may be used [5].

Alongside other available filtering techniques, the MAF serves as an additional tool to enhance the representation of the decay curve in the developed software; the MAF is utilized as one of the filtering options to smooth the decay curve of the signal.

3.2.4 Schroeder-Stirling inverse integral

The Schroeder-Stirling integral is a technique used to derive the energy decay curve of a room impulse response [1]. It involves performing the inverse integration of the squared signal over the interval $[0, T]$, where T represents the point at which the signal is considered to be dominated by background noise. Mathematically, it is defined by equation (1).

$$ETC(t) = \int_t^T h^2(\tau)\, d\tau \tag{1}$$

In order to determine the accurate decay characteristics of the impulse response and avoid overestimation, it is essential to apply a background noise compensation method, such as Lundeby's [6]. This method helps in finding the limiting value and enable the determination of the effective duration of the impulse response.

3.3 Reversed RIR method

When applying band-pass filtering by a fraction of an octave, there is a potential issue with the filter's own ringing [7]. As the filters are not perfect, they introduce a temporal response that causes ringing, especially with higher selectivity (Q) settings. This ringing leads to distortion in the signal's decay, overestimating the calculated reverberation time. This effect is more pronounced in the lower frequency bands when using highly selective filters (e.g. when analyzing by thirds of octaves) and when the reverberation time is very short [8].

To address this ringing problem, F. Jacobsen and J. H. Rindel proposed a solution of reversing the time signal before applying the filtering and then reversing it back again afterward [9]. This approach has been incorporated into the code's development and is one of the processing options that the user can select.

3.4 Lundeby's crosspoint estimation method

The purpose of this method is to find the moment at which the decay curve intercepts the background noise. This point is set as the upper integration limit which significantly improves the results of Schroeder filtering. Lundeby's method employs an iterative algorithm that selects an appropriate time interval and applies linear regression. The algorithm consists of four key steps:

1. The IR is averaged and squared over a short time period (10-50 ms).

2. Second, the background noise is estimated using the last 10 % of the signal.

3. Third, a slope is estimated from 0 dBFS up to the level of the background noise.

4. Fourth, the algorithm identifies the cross-point between the previously applied linear regression and the corresponding noise level.

This process is repeated until a convergence criterion is met [6].

The described method effectively separates the IR decay from the background noise, improving the accuracy of the Schroeder filtering technique. Lundeby's iterative algorithm allows for precise estimation of the relevant parameters, ensuring reliable results in acoustic analysis.

3.5 Acoustical parameters

3.5.1 Reverberation time (RT)

The reverberation time is defined as the amount of time it takes for the intensity of sound in an enclosed space to decay by 60 dB after the sound source has ceased emitting; this is what we call T_{60}.

Regarding the ISO 3382 standard, if the dynamic range is smaller, it is also possible to provide the reverberation time. The decay time between 5 dB and 35 dB can be used to determine T_{30}, and using the same logic, between 5 dB and 25 dB, the T_{20} can be obtained [10].

3.5.2 Early Decay Time (EDT)

The Early Decay Time (EDT) is the duration it takes for the decay curve to decrease by 10 dB from the moment the sound source stops emitting.

3.5.3 Transition time (Tt) and Early Decay Time - transition (EDTt)

The transition time refers to the point when the system transitions from a deterministic behavior to a stochastic one. It is defined as the moment when the accumulated energy of the impulse response reaches 99 % of its total energy. This parameter can also be interpreted as the effective duration of the early field.

Besides the EDT calculated based on the first 10 dB drop, another method to assess the early decay is by calculating a linear regression between the maximum peak of the impulse response and the transition time. This yields the parameter EDTt, which provides further evaluation of the early decay characteristics.

3.5.4 Clarity (C_{te})

This parameter is defined by the ISO 3382 standard as a "balance between early and late arriving energy". The parameter can be calculated for either a 50 milliseconds or a 80 milliseconds early time limit depending on whether the results are intended to relate to conditions for speech or music respectively. Definition of C_{te} is shown in equation (2) [10].

$$C_{te} = 10 \log \left(\frac{\int_0^{t_e} p^2(t)\, dt}{\int_{t_e}^{\infty} p^2(t)\, dt} \right) \tag{2}$$

Where $p(t)$ is the impulse response and t_e is the limit of integration, being 50 milliseconds for C_{50} and 80 milliseconds for C_{80}.

3.5.5 Early Interaural Cross Correlation Coefficient (IACC$_{EARLY}$)

In the process of measuring binaural impulse responses, it is valuable to determine the similarity between the signals captured by each ear. This involves calculating the cross-correlation function between the left and right channels of the stereo signal within specified integration boundaries. Mathematically, the normalized interaural cross-correlation (IACCF) function is defined by equation (3).

$$IACC_t(\tau) = \frac{\int_0^{t} h_L(t) \cdot h_R(t+\tau)\, dt}{\sqrt{\int_0^{t} h_L(t)^2\, dt \cdot \int_0^{t} h_R(t)^2\, dt}} \tag{3}$$

Where $h_L(t)$ and $h_R(t)$ represent the impulse responses acquired from the left and right channels, respectively.

4. Code development

The software was developed with Python 3.9, using the following libraries:

- *numpy*
- *matplotlib*
- *scipy.signal*
- *PyQt5*
- *soundfile*

The block diagram and signal flow of the software is shown in Figure 1.

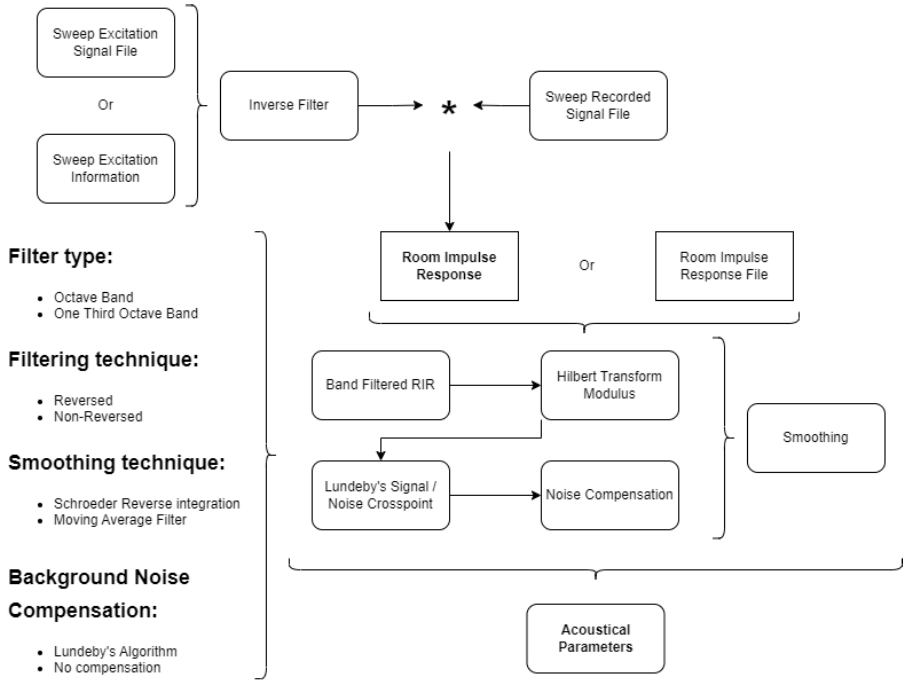

Figure 1: Block diagram of the RIR Processing software

When executing the software a settings window pop's up, in which the user is prompted to select between three different options regarding the input signal. The first option is to load both the exponential swept sine signal and its corresponding recording so that the software calculates the inverse filter and convolves it with the other file. The second option is to load the recording of the sweep and enter the sweep excitation information (start and end frequencies, duration, and sampling frequency) which is used to obtain the inverse filter. Lastly, the user can load a processed impulse response (previously convolved, or a clap-like recording).

When a sweep file is provided, the software obtains the fractional octave bands in which the results will be invalid by calculating the FFT of the sweep. As a full spectrum (20 Hz - 20 kHz) sweep has an attenuation of -10 dB/decade, between its first and last frequency band it could have a maximum difference of -30 dB. Based on that assumption, if a given band is below 30 dB with respect to the first one (the one that has the greatest level) it's discarded. For further precision, if the upper-limit frequency of the first band is below the sweep's spectrum maximum frequency, that band is disregarded too. Nevertheless, calculations are made in all frequency bands and this information is available in the results to the user with invalid bands being represented as red columns.

After the convolution, the impulse is trimmed to discard the higher-order non-linearity responses. The user selects if the analysis must be in fractional octave bands or one-third of octave bands, and if the reverse filtering is to be applied. Afterward, a higher-order function filters the impulse and retrieves a function that calculates the Energy Time Curve (ETC), obtains Lundeby's crosspoint with background noise, and the noise compensation (if desired). If the smoothing is set to be the Schroeder integration method, the software calculates it. Lastly, the results are obtained for each frequency band filtered impulse, and appended to a table.

The user can select the frequency band represented in the graph by clicking a cell in the respective column. Both the graph and the table can be saved to a png and CSV file, respectively.

The graphic user interface of the settings window is shown in Figure 2 and the results window in Figure 3.

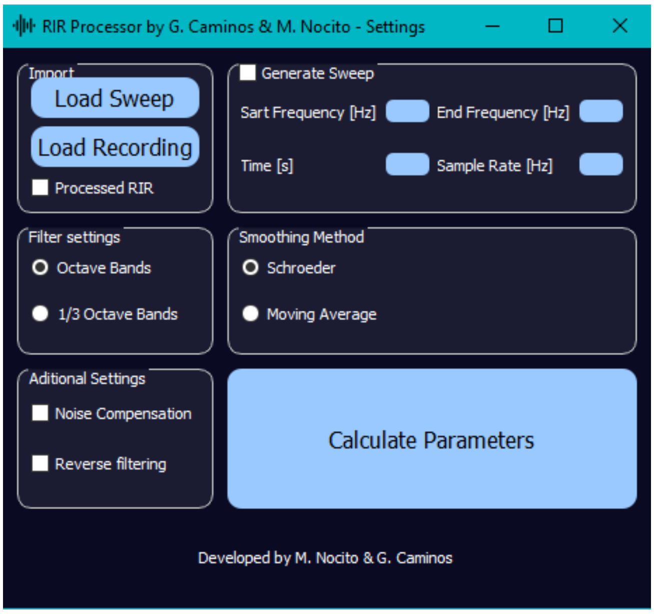

Figure 2: Set Up Window.

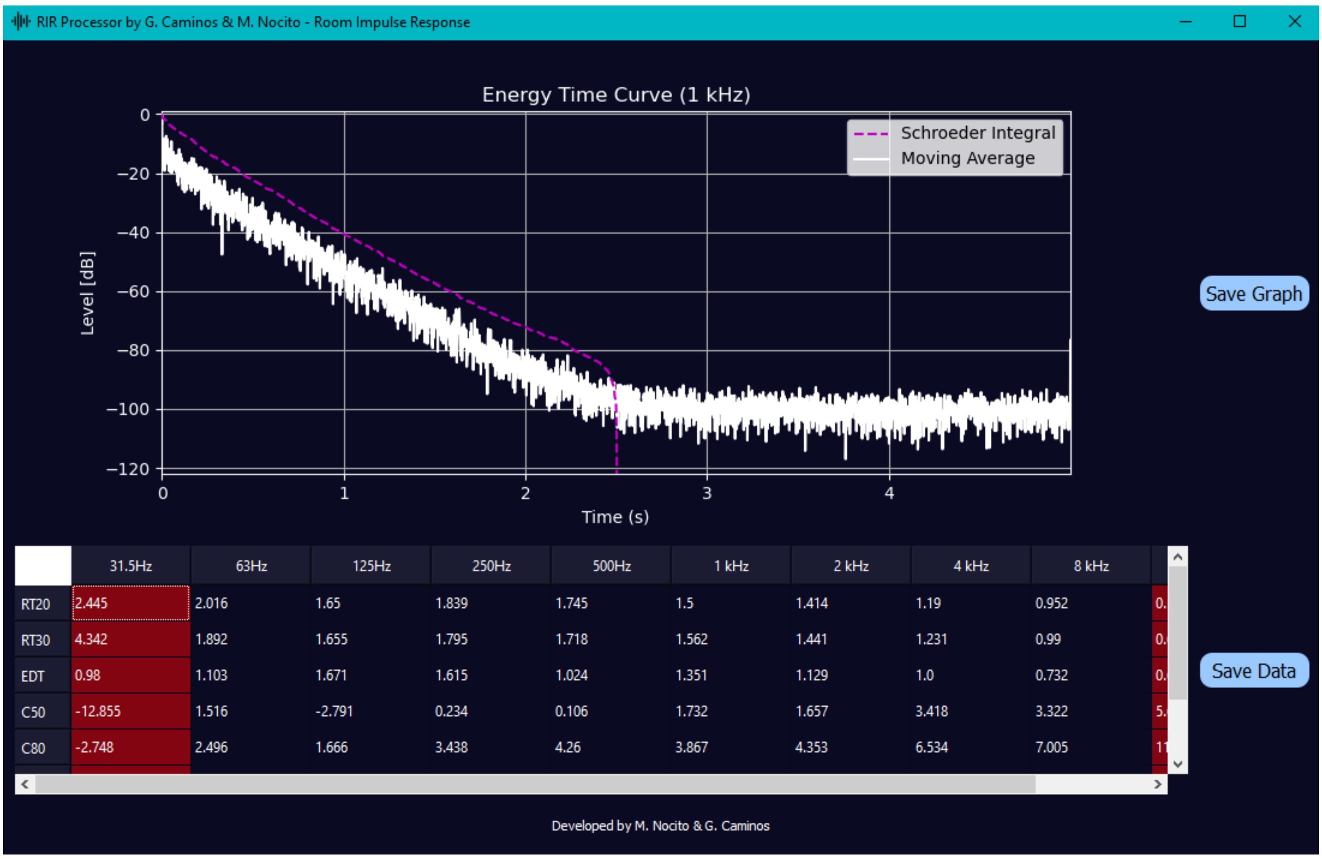

Figure 3: Results Window.

5. Results and discussions

5.1 Comparison with commercial software

The subsequent sections show the comparison of the results obtained with the developed software using Schroeder's smoothing method instead of the MAF, against commercial software such as the Aurora Plugin used in Adobe Audition.

5.1.1 Reverberation time

Figures 4 and 5 show the comparison between the T20 and T30 respectively obtained with the developed software and the commercial software, also the percentage difference between each parameter is shown per octave band.

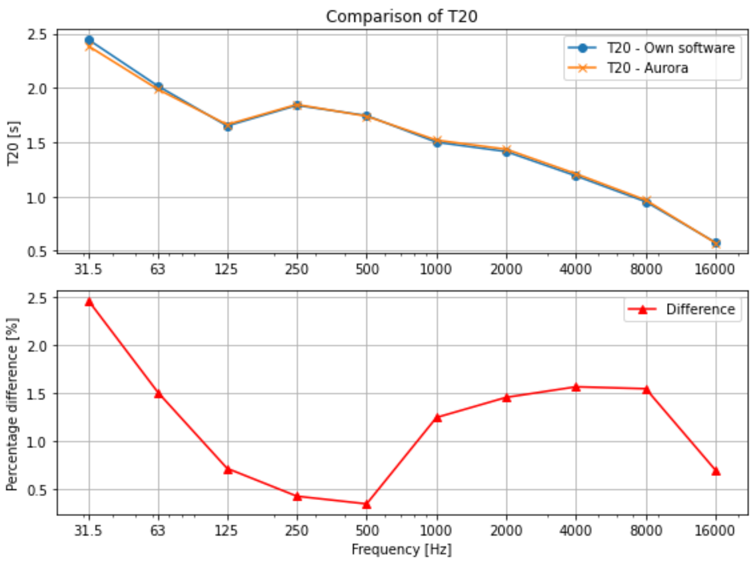

Figure 4: Comparison of T20 between software.

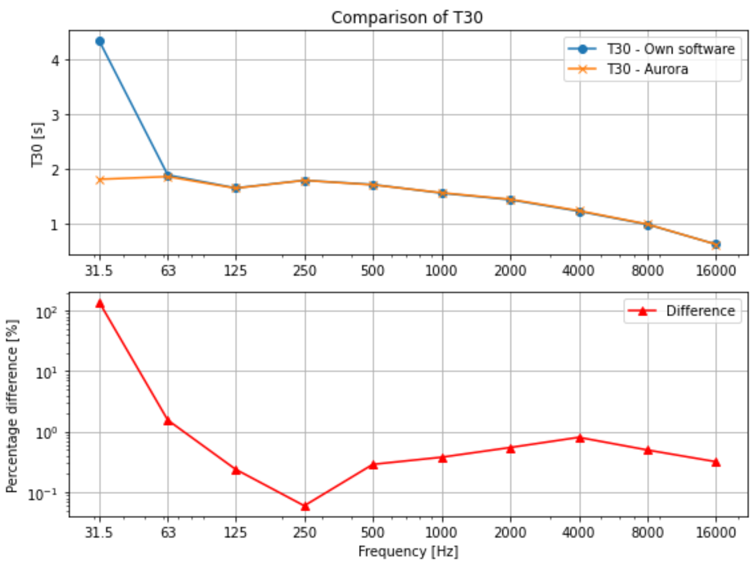

Figure 5: Comparison of T30 between software.

It can be observed that for both reverberation parameters, the designed software yields very similar values, with the exception of the T30 parameter in the 31.5 Hz band. However, for the remaining values, the percentage difference does not exceed 2.5 %. This demonstrates the accuracy of the program in calculating this particular parameter.

5.1.2 EDT, EDTt & Tt

Similar results are obtained for the EDT parameter; the comparison can be seen in Figure 6.

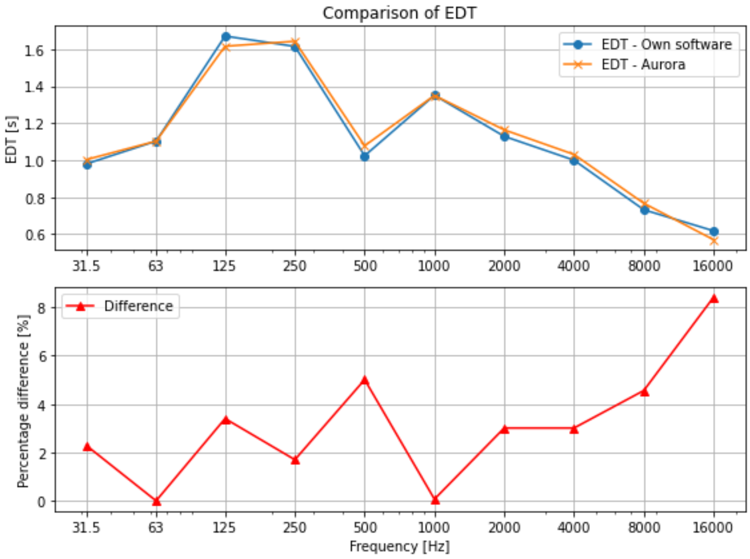

Figure 6: Comparison fo EDT between software.

In this case, the maximum difference occurs in the 16 kHz band, where the maximum difference is 8.4 %. On the other hand, in the 63 Hz and 1 kHz bands, the value is exactly the same as the one obtained from the Aurora plugin. For the remaining bands, the differences between the software do not exceed 5 %.

Since the Aurora plugin does not calculate EDTt nor Tt, a comparison in the same software was made evaluating the different possible filtering methods. Comparison for EDTt is shown in Figure 7

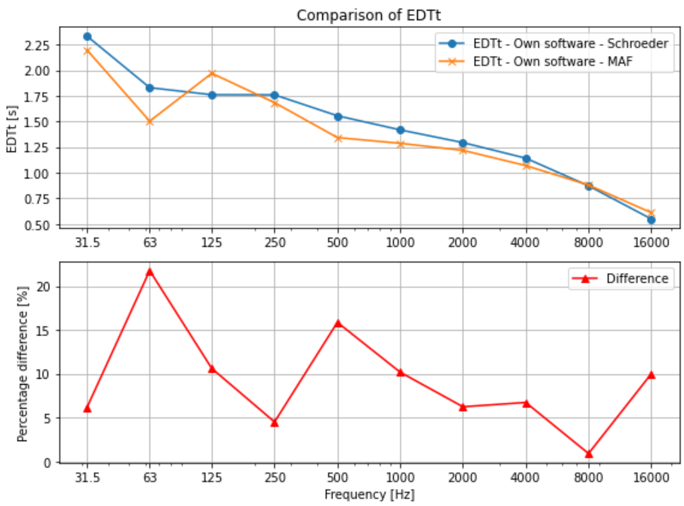

Figure 7: Comparison of EDTt parameter with different filtering methods (Schroeder and MAF) in the same software.

The greater difference between the EDTt calculated with different methods takes place in the band of 63 Hz, with a difference of 21.76 %, additionally, it is possible to observe that the MAF filtering method results in an underestimation of the EDTt practically along all the spectrum.

A similar situation happens with the Transition Time parameter, it was not possible to do a comparison with other commercial software. The results for Transition time are shown in Figure 8.

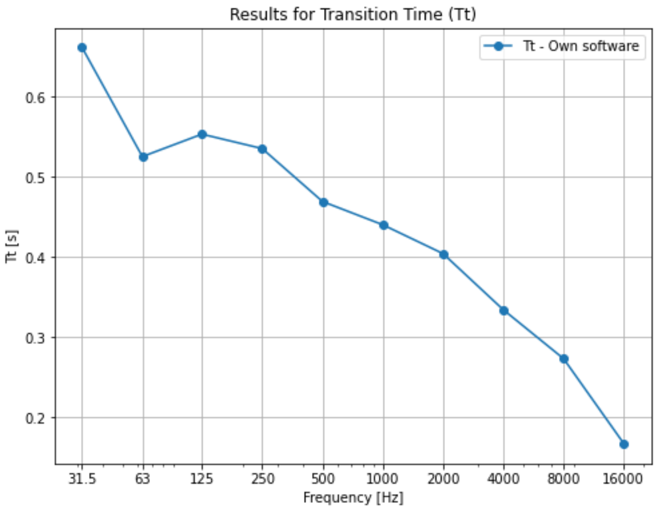

Figure 8: Results for Transition time in the developed software.

Observing the figure, it is possible to note that the lowest Tt value, indicating a relatively faster decay, is observed at 16 kHz with a Tt of 0.167 seconds. Overall, there is a decreasing trend in Tt values as the frequency increases, suggesting a faster decay at higher frequencies. The longer Tt values at lower frequencies imply a slower decay, contributing to a sustained perception of low-frequency sounds. Conversely, the shorter Tt values at higher frequencies indicate a quicker decay of high-frequency sounds, with this analysis, it is possible to conclude that the results yielded by the developed software follow the expected trend in the study of this parameter.

5.1.3 Clarity

The results for the clarity parameters C50 and C80 can be observed in Figures 9 and 10.

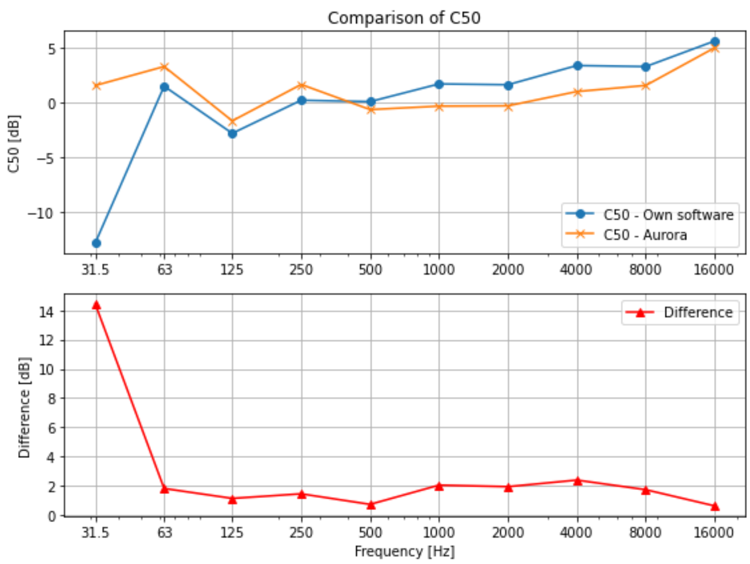

Figure 9: Comparison for C50 between software.

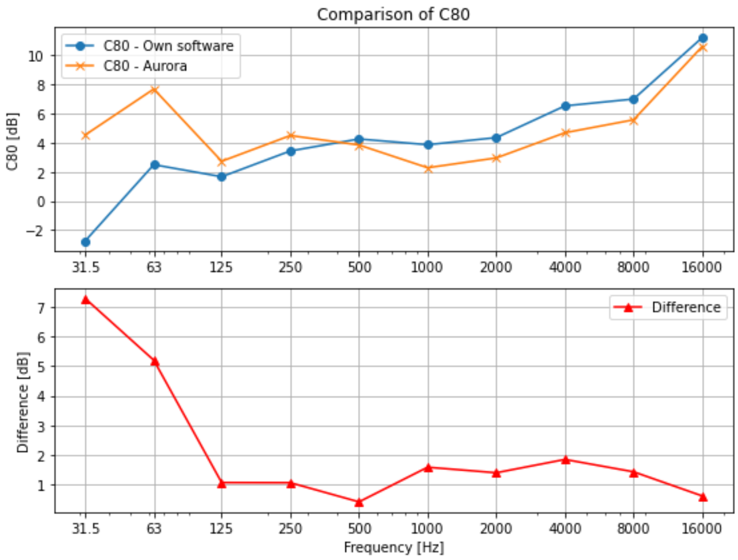

Figure 10: Comparison for C80 between software.

Similar to what was observed with the reverberation times, there is a substantial difference in the clarity parameters in the 31.5 Hz band. However, for the rest of the octave bands, the difference between the parameters calculated using the proposed software and the commercial software does not exceed 2.5 dB, which is assumed to be imperceptible to the human ear. This repeated discrepancy may suggest the presence of some error within the code, requiring further investigation and refinement. A possible cause of this anomaly is that the ringing of the filter may be modifying the decay in that low-frequency band.

6. Conclusions

First and foremost, the analysis confirms the reliability of the developed software for most of the studied descriptors. This demonstrates that the theoretical framework used for analyzing these descriptors is directly applicable to practical cases, as evidenced by the achieved results.

Additionally, it was observed that meticulous implementation of the processing software is crucial for obtaining reliable results across the entire frequency spectrum. Notably, one of the tested filtering methods exhibited lower reliability, indicating the need for further analysis and improvement in future works. Moreover, the implemented filtering method showed decreased reliability when applied to lower frequencies, highlighting the necessity for algorithm calibration in future research.

To enhance the software as a comprehensive analysis tool, several improvements can be considered for future works. Incorporating parameters such as Definition parameters (D50, D80), Impulse to Noise Ratio (INR), and Strength parameter (G) would provide a more comprehensive suite of analysis capabilities. Additionally, implementing Chu's Noise correction method could significantly improve the denoising of processed impulse responses.

To enhance the user experience and facilitate analysis, further implementation of plotting tools within the graphical user interface is recommended. Features like zooming the plot or displaying (x, y) coordinate markers would contribute to a more user-friendly interface.

Lastly, to ensure robust implementation, it would be beneficial to anticipate and account for a broader range of user-induced errors. Including a dedicated "How to use" section would greatly expand the available use cases and provide users with a seamless workflow.

Incorporating these suggestions and considerations would advance the software's capabilities, expand its usefulness, and provide an enhanced experience for users.

References

[1] Manfred R. Schroeder. “New Method of Measuring Reverberation Time”. In: *The Journal of the Acoustical Society of America* 37 (1965), pp. 1187–1188.

[2] Angelo Farina. “Simultaneous measurement of impulse response and distortion with a swept-sine technique”. In: *Audio Engineering Society Convention 122*. Audio Engineering Society. 2000.

[3] David Hilbert. “On the theory of complex waves”. In: *Annals of Mathematics* 16.3 (1915), pp. 135–156.

[4] Abraham Savitzky and Marcel J. E. Golay. “Smoothing and Differentiation of Data by Simplified Least Squares Procedures”. In: *Analytical Chemistry* 36.8 (1964), pp. 1627–1639.

[5] Steven W. Smith. *The Scientist and Engineer’s Guide to Digital Signal Processing*. 2011. URL: `https://www.dspguide.com/ch15/5.htm`.

[6] H. Bietz et al. “Uncertainties of Measurements in Room Acoustics”. In: *Acustica* 81.2 (1995), pp. 344–355.

[7] B. Rasmussen, J. H. Rindel, and H. Henriksen. “Design and measurement of short reverberation times at low frequencies in talks studios”. In: *Journal of the Audio Engineering Society* 39.1/2 (1991), pp. 47–57.

[8] D. Cabrera and X. Jianyang. “Calculating Reverberation Time from Impulse Responses: A Comparison of Software Implementations”. In: *Australian Acoustical Society* (Jan. 2016).

[9] F. Jacobsen and J.H. Rindel. “Letters to the editor, Time reversed decay measurements”. In: *Journal of Sound and Vibration* 117.1 (1987), pp. 187–190.

[10] International Organization for Standardization. *Acoustics - Measurement of room acoustic parameters - Part 1: Performance spaces*. ISO. Geneva, Switzerland, 2003.

7. Student produced question

Is there a filtering technique that provides better results? The filter pre-ringing affects low frequency bands because its high Q. Reverse filtering solves in part this problem, but the ringing is still present in the end of the impulse. May forward - backward filtering techniques be used to definitively solve this problem? What order should be that filter?