RIRs Processing Software Development

Paula Ortega Riera

Universidad Nacional de Tres de Febrero, Ingeniería de Sonido, Buenos Aires, Argentina
email: pauortegariera@gmail.com

Nicolas Pelella

Universidad Nacional de Tres de Febrero, Ingeniería de Sonido, Buenos Aires, Argentina
email: nicolaspelella@gmail.com

This work exposes the development of a code for the processing of Room Impulse Responses by means of Python programming software. An interactive user interface that allows the user to enter a sound record or an RIR in monophonic or stereophonic format is created, in order to determine a series of acoustic parameters. For the purpose of obtaining the EDT, T20, T30, C50, C80, Tt, EDTt and IACCearly, the software filters the user's choice by octave bands and third octave bands. The signal is then smoothed out and compensation for background noise is performed. Tables and graphs contain data obtained from the results of each descriptor that can be exported. Finally, a comparison of the results obtained with the Aurora module of the Audacity software is made.
Keywords: Room Impulse Response, Acoustics Parameters, Python

1. Introduction

When characterizing the response of a room to excitation, reverberation time is always considered as the predominant indicator of its acoustic properties. Although it is still considered an important parameter, there is consensus that other types of measurements, such as relative sound pressure levels, early/late energy ratios, lateral energy fractions, interaural cross-correlation functions, and noise levels background are necessary for a more complete evaluation of the acoustic quality of the rooms [1].
As established by Guski-Vorländer, the most commonly used way to calculate the reverberation time is to measure the impulse response of a room, perform the Schroeder backward integration and estimate the signal decay by means of a linear regression [2].
In this work, a graphical user interface (GUI) that allows obtaining the values by octave bands and thirds of an octave of the main acoustic descriptors of a room is developed. For this, signal processing is implemented to optimize the calculation of these descriptors.

2. Theoretical Framework

2.1 Room Impulse Response

The impulse response (IR) of a system allows the characterization of the acoustic properties of the room. To obtain it, the convolution of a sinusoidal sweep and its inverse filter are carried out. The usage of exponential sine sweep, compared with linear sine sweeps, provides several advantages in terms of signal-to-noise ratio and management of not-linear systems [3].
A sine sweep is a sinusoidal signal in which the frequency varies exponentially as a function of time over the range of interest. Throughout that range, one frequency is excited at a time, increasing the concentration of energy. As a consequence, the signal-to-noise ratio rises. The sine sweep signal is given by (1).

$$x(t) = \frac{TW_1}{ln(\frac{W_2}{W_1})}(e^{\frac{T/t}{ln(\frac{W_2}{W_1})}} - 1) \quad (1)$$

where W_1 is the initial angular frequency, W_2 is the final angular frequency and T is the total duration time.
Farina [4] states that the inverse filter is the input signal itself inverted along the time axis. Thus, the instantaneous frequency decreases with time. In the case of exponential sine sweep, amplitude modulation is added to compensate for the difference in generated energy between high and low frequencies. This is described in equations (2), (3) and (4).

$$w(t) = \frac{\partial[x(t)]}{\partial(t)} = \frac{TW_1}{ln(\frac{W_2}{W_1})}e^{\frac{t}{ln(\frac{W_2}{W_1})}} \quad (2)$$

$$m(t) = \frac{W_1}{2\pi w(t)} \quad (3)$$

$$k(t) = m(t) * x(-t) \quad (4)$$

2.2 Signal smoothing

A median moving filter (MMF) with a window length related to the minimum frequency band under analysis is used. The median is just the middle value in a distribution, and it is less influenced by the outliers of the impulse response, as opposed to the mean that can be "dragged" up or down by extreme values [7].
Other processing related to smoothing is given by the Schroeder backward integral [6]. It allows the production of a smoothed, monotonic decay curve by back integrating the impulse response h (t) over the measurement range [0, T] and converting to a logarithmic scale. The expression of said integration is given by equation (5).

$$ETC(t) = 10\, log_{10}\left[\frac{\int_t^T h^2(\tau)\, d\tau}{\int_0^T h^2(\tau)\, d\tau}\right] \quad (5)$$

2.3 Noise compensation

The problem occurs when the background noise floor is included within the integration range. There are different methods to suppress the effects of noise on the acoustic parameters of the room. In this sense, an attempt to control the tail of the integration for which techniques are proposed to reduce the slope bias is made.

Lundeby [8] presents a method to automatically determine the background noise level, the decay noise truncation point, and the late decay slope of an impulse response. We proceed from the quadratic average of the RIR in local intervals where the initial level of background noise in the late part of the signal is estimated. Then, from the decay slope obtained by a linear regression in the corresponding time interval, the initial truncation point is determined as the intersection of the decay slope and the background noise level. From the above, new time intervals and a new late fall slope are determined. The process is iteratively repeated up to a maximum of 5 iterations.
On the other hand, a second type of background noise compensation proposed by Chu [8] is taken into account. It is a method in which the root mean square value of the noise floor is subtracted from the original square impulse response before backward integration. If the noise floor estimate is accurate and the noise is stationary, the resulting backward integrated curve is close to the ideal decay curve.

2.4 Reverberation Time

According to the ISO 3382 standard [1], the reverberation time is the time necessary for the sound pressure level to decrease by 60 dB after the sound source ceases. Since the difference between the background noise and the excitation signal is generally less than 60 dB, the RT60 can be evaluated at a lower dynamic range and extrapolated to a 60 dB decay time. Therefore, the descriptors T30, T20 and EDT are calculated. They are defined as the time necessary for the sound pressure level to go from -5 dB to -35 and -25 dB relative in the first two, and from 0 to -10 dB relative for the EDT, multiplied by a factor to extrapolate this interval to the -60 dB full scale decay (2 in the case of the T30, 3 in the case of the T20 and 6 in the case of the EDT).

2.5 Modern parameters

In the context of the analysis of the acoustic texture of the rooms, studies on the identification of the early reflections and the differentiation with respect to the late reflections in an RIR are carried out. Given this, new parameters have been defined, including the transition time Tt and the EDTt [9].
A RIR is made up of early reflections and late reflections. The instant of separation of the two is called the transition time Tt. After Tt, the reverb tail has an exponential decay of the Gaussian white noise type.
Early reflections are defined as every amplitude outlier present in the RIR. The transition time can be obtained as the instant in which the accumulated energy of the early reflections reaches 99% of their total energy [7]. Then, the EDTt is given by the time necessary for the sound pressure to decay from 0 to the value of the accumulated energy in dB at Tt.
In the context of the study of intelligibility, the parameter of clarity is introduced. It is defined as the relationship between early energy and late or reverberant energy.
The C50 represents an objective measure of the clarity or intelligibility of speech. This parameter assumes that late reflections are unfavorable for speech understanding, causing masking, which makes speech unclear. However, if the delay does not exceed a certain time limit, the reflections will contribute positively to intelligibility. The critical time limit separating useful from harmful reflections is approximately 50 ms (milliseconds). This parameter is defined in equation (6).

$$C50 = 10\, log_{10}\left[\frac{\int_{0}^{50} p^2(t)\, dt}{\int_{50}^{\infty} p^2(t)\, dt}\right][dB] \qquad (6)$$

Similarly, the C80 represents the objective measure of musical clarity and is calculated from equation (7).

$$C80 = 10\, log_{10}\left[\frac{\int_{0}^{80} p^2(t)\, dt}{\int_{80}^{\infty} p^2(t)\, dt}\right][dB] \qquad (7)$$

Binaural impulse responses (BIRs) are measured to clarify the acoustical characteristics of the sound filed in concert halls and opera houses. Since BIRs indicate the transfer function of sound fields, an anechoic source convolved with the BIRs represents the signal that listeners hear in that sound field. To find out the acoustical characteristics of the original sound source, it is useful to calculate the autocorrelation function (ACF) because the parameters extracted from the ACF are closely related to the subjective preference of listeners. It can be said that convolution and correlation are the most important calculative techniques in room acoustics [11].

The normalized interaural cross correlation function (IACF) is first defined using the equation (8), where pl(t) is the impulse response at the entrance to the left ear canal and pr(t) is the impulse response at the entrance to the right ear canal.

$$IACF_{t_1,t_2,}(\tau) = \frac{\int_{t_1}^{t_2} pl(t)pr(t+\tau)dt}{\sqrt{\int_{t_1}^{t_2} pl^2(t)dt \int_{t_1}^{t_2} pr^2(t)\, dt}} \qquad (8)$$

The interaural cross correlation coefficients, IACC, are then given by equation (9).

$$IACC_{t_1,t_2} = max\left|IACC_{t_1,t_2}\right| \quad for\ -1\, ms < \tau < 1\, ms \qquad (9)$$

Finally, the value of t_1 and t_2 to obtain the IACCearly is $t_1 = 0\, ms$ and $t_2 = 80\, ms$.

3. Python code development

3.1 Obtaining Impulse Response and Filtering

In the first contact with the user interface, two options are presented. On the one hand, it is possible to enter a sound record with a .wav extension and the necessary information for the automatic generation of the corresponding inverse filter. The user is asked to determine the duration of the sine-sweep with which the room is excited and the frequency range of the sweep.

From the *read* function of the *Soundfile* library, the sound record is read. The input audio is then convolved with its mathematically generated inverse filter. The above is carried out from the *fft.fft* function of the *Numpy* library, for which the product of the Fourier transform of the imported audio and the inverse filter is carried out.

On the other hand, it is possible to enter the interface directly with a room impulse response.

The next step is to determine the type of filtering. According to the standard [13], the impulse response can be filtered in octave bands or in third octave bands. This processing is done through the *signal* function of the *Scipy* library.

3.2 Signal smoothing

For signal smoothing it is possible to perform two different methods or combine both and perform them simultaneously.
One of the methods that can be selected consists of applying a moving median filter (MMF) to the signal. This filtering is done from the *median* function of the *Ndimage* library. The user can enter the value of the filter window, otherwise a pop up message is delivered to the user advertising that a window must be set.
Another path is given by the Schroeder backward integral. The code allows the user to calculate this integral by inverting the time interval from the *flip* function of the *Numpy* library. Also, in this function, the background noise compensation algorithms of Chu and Lundeby are applied.
The Chu code calculates the arithmetic mean of the last 10% of the signal, where noise is assumed to be predominant, through the mean function of the Numpy library.
Lundeby Noise Compensation obtains the noise level of the last 10% of the signal in dB RMS. Then it divides the signal into intervals and obtains the mean from the mean function of the Numpy library.
The linear regression is obtained between the 0dB interval and the closest mean noise energy + 10dB. If the signal-to-noise ratio is insufficient, that is, it is greater than -20 dB, then new time intervals for the mean are calculated. From this value an iterative comparison process begins. From the decay slope obtained by linear regression in the corresponding time interval, the initial truncation point is determined as the intersection of the decay slope and the background noise level. The process is repeated up to a maximum of 5 iterations.
Finally, the interface allows the user to apply both the moving median filter and the Schroeder backward integration with their respective compensation for background noise.

3.3 Parameter calculation

A global function that performs the calculation of the acoustic descriptors is defined.
The times from which each parameter will be calculated are determined. There is a common variable to calculate T10, T20 and T30 called initT = -5 that determines the initial time from which there is a 5 dB decay with respect to the smoothed signal. The end time variables for each parameter are also defined as endT10, endT20 and endT30 respectively.
In the case of EDT, its time interval is initialized at initEDT = 0 and ends at endEDT = -10. Similarly for the EDTt, it is initialized to initEDTt = 0 and the end time is determined by endEDTt = x.
Then, from the *linregress* function of the *statistics* module within the *Scipy* library, a linear regression is performed from which the decay slope is obtained.
This global function also performs the Clarity calculation, for which the integration times are defined as time50 = 50 ms and time80 = 80 ms.
For the calculation of Tt, the equation described in the state of the art of this work is implemented. Mathematical operations of the interface are used.

3.4 Binaural RIRs processing

Finally, the IACCearly are calculated in the case of a stereo RIR. In this case the user selects the input of a binaural impulse response.
A function is created that, from the processing of the right and left channels of the RIR, obtains the necessary variables for the calculation of the IACF function. These variables are defined as PL and PR, defined as the pressures of each channel respectively.
Integration times are established as variables as well. Finally, the calculation is carried out according to the regulations for each frequency band.

3.5 Code signal flow

Figure 1 shows the signal flow of the code developed in Python.

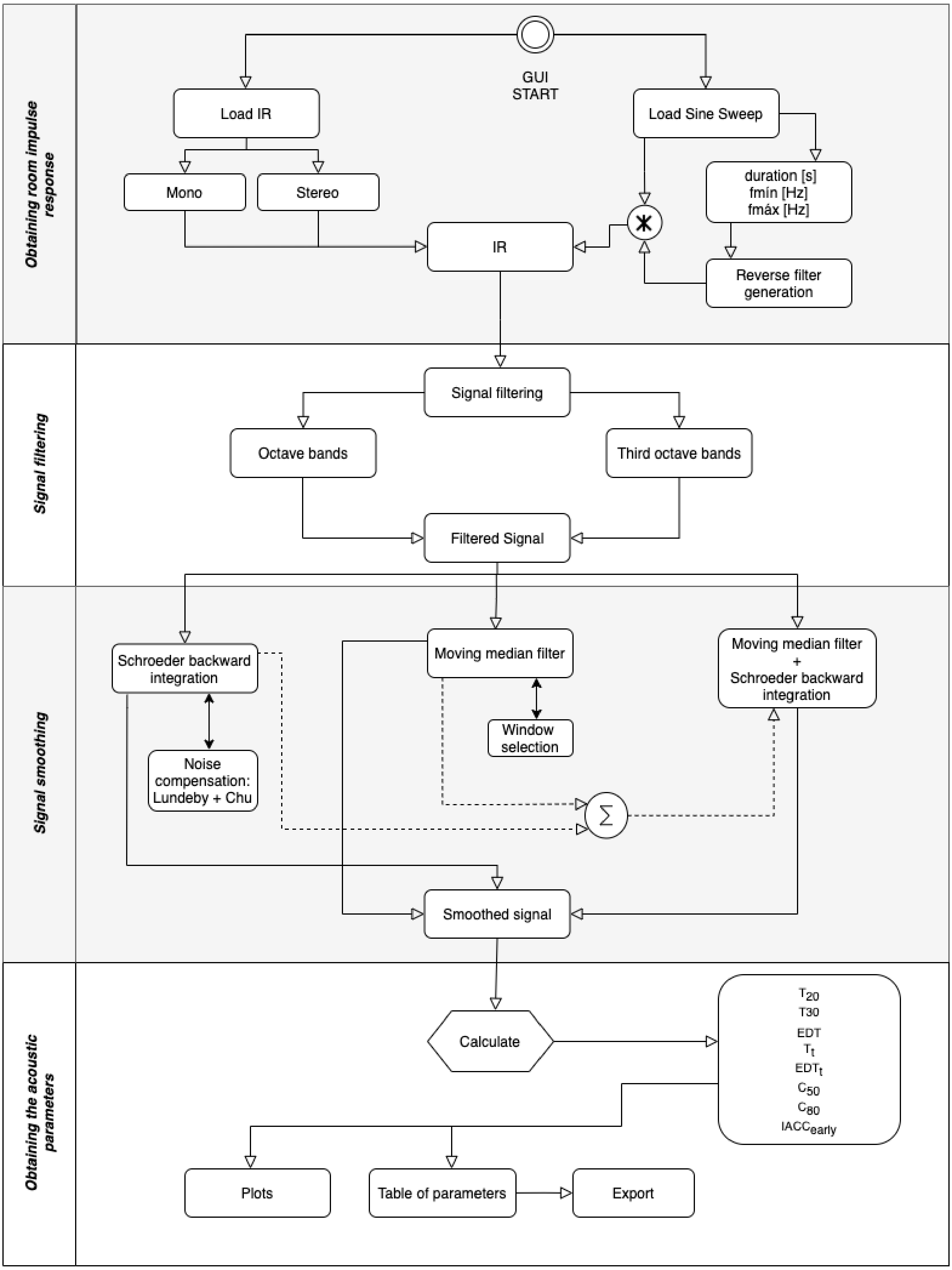


Figure 1. Signal Flow of the Python code developed.

3.6 Graphical User Interface (GUI)

Figure 2 shows the graphical user interface environment.

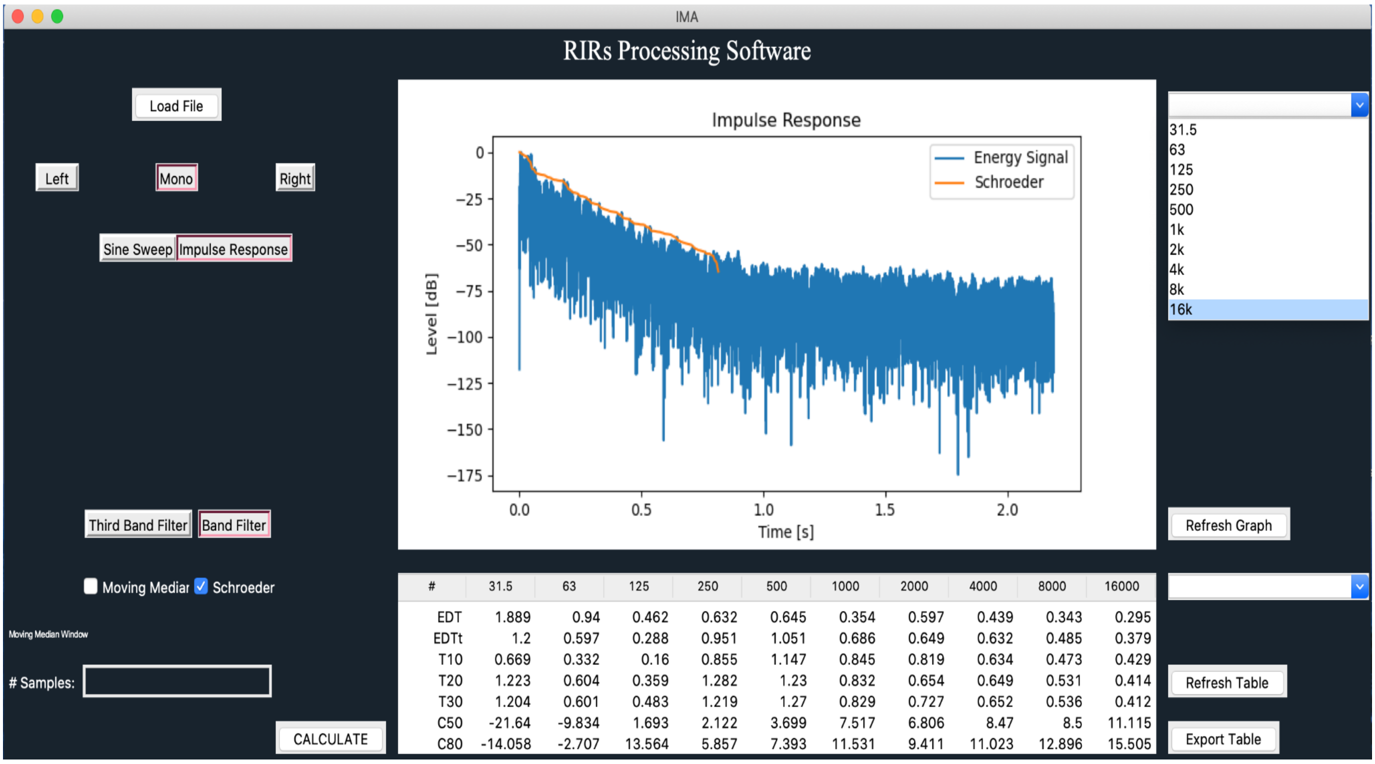


Figure 2. Developed Graphic User Interface .

4. Results

This section shows the values obtained for each descriptor with the developed interface. In addition, for the same audio file, the values of the descriptors under study are obtained in the *Audacity* software, using the *Aurora* module.

4.1 Analysis of acoustic parameters.

The analysis in this section is performed in octave bands. The graphs of the descriptors calculated by thirds-octave bands are shown in Appendix A.

Figure 3 shows the EDT parameter calculated using the software developed by applying the Schroeder backward integral (Sch) as the only smoothing method, only a moving median filter (MM) and a combination of both (Sch + MM). In addition, the value obtained from the descriptor through Aurora is attached.

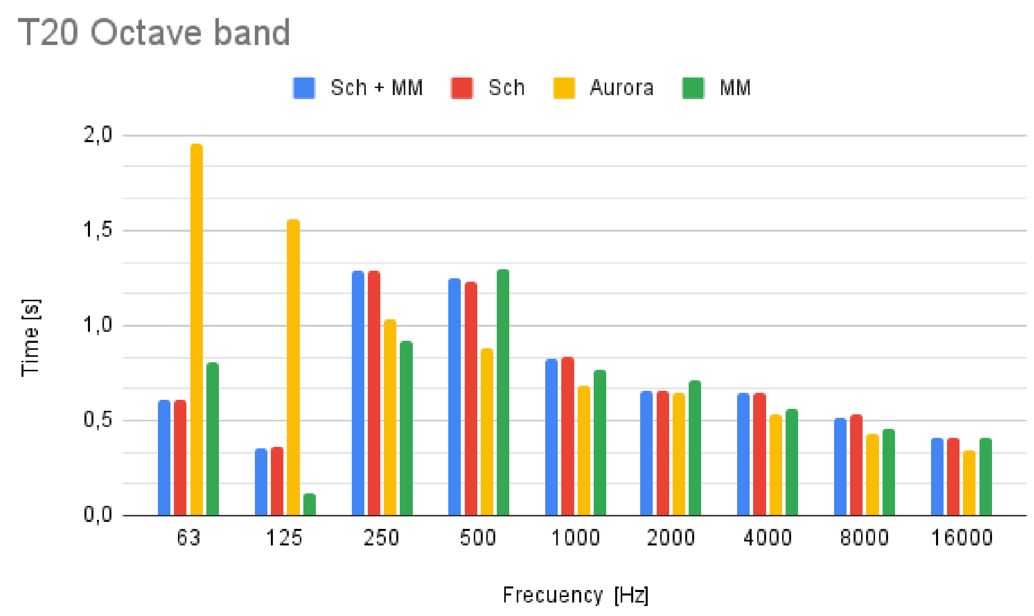


Figure 3. T20 descriptor per octave band.

A greater difference in the low frequency bands with values of almost 2 seconds can be observed . From 1 kHz on, there is a greater correlation between the values obtained with each software. Regarding the filtering, it is shown that, at high frequencies, filtering only by the moving median provides results closer to the values given by commercial software. There are no significant differences between applying only the Schroeder backward integral or implementing it in conjunction with the moving median filter. Figure 4 presents the values obtained for the descriptor T30. In this case, time differences similar to those of the T20 are obtained for each software. The biggest difference resides in low frequencies where the Aurora software reaches higher values than the developed software.

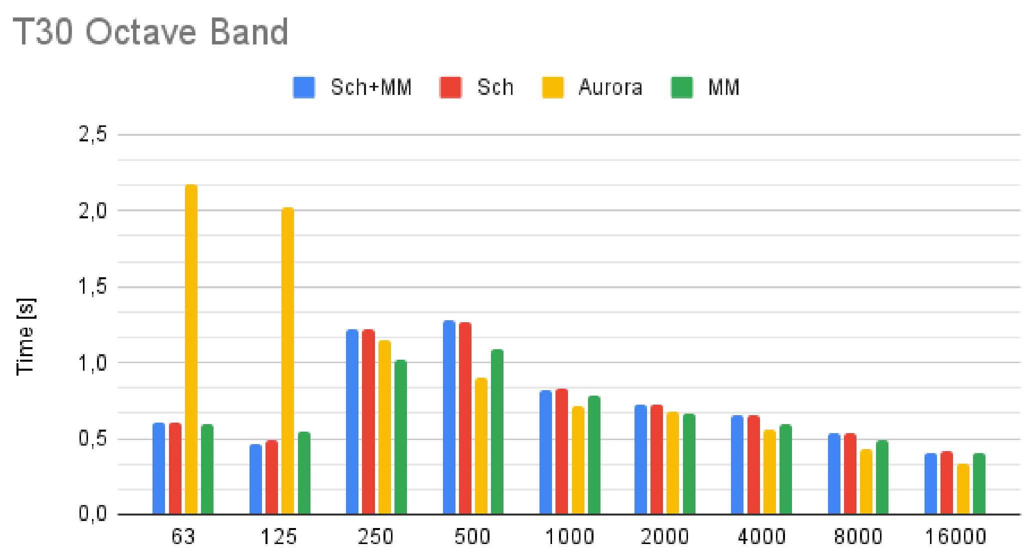


Figure 4. T30 descriptor per octave band.

Figure 5 presents the values obtained for the descriptor EDT.

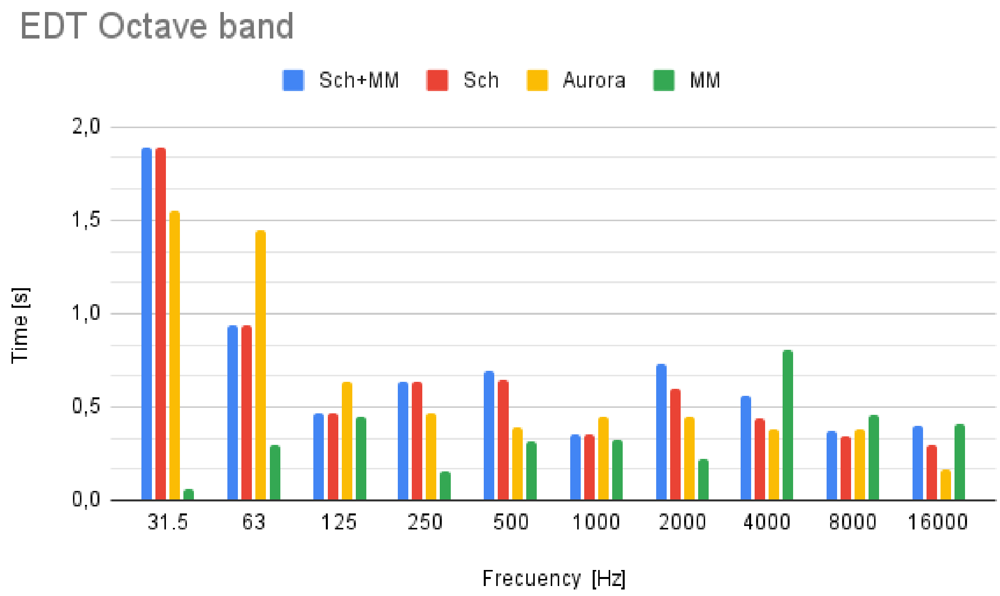


Figure 5. EDT descriptor per octave band.

For this parameter, differences of less than 0.5 seconds are observed between both softwares. The greatest difference occurs at low frequencies in the 31.5 Hz and 63 Hz bands. Filtering only by the moving median shows great dispersion with respect to the Schroeder backward integral and even with respect to Aurora smoothing. The biggest differences for this method are at high frequencies, starting at 2 kHz.

In Figure 6 the values obtained for the clarity parameter are presented. C50 and C80 are exposed.

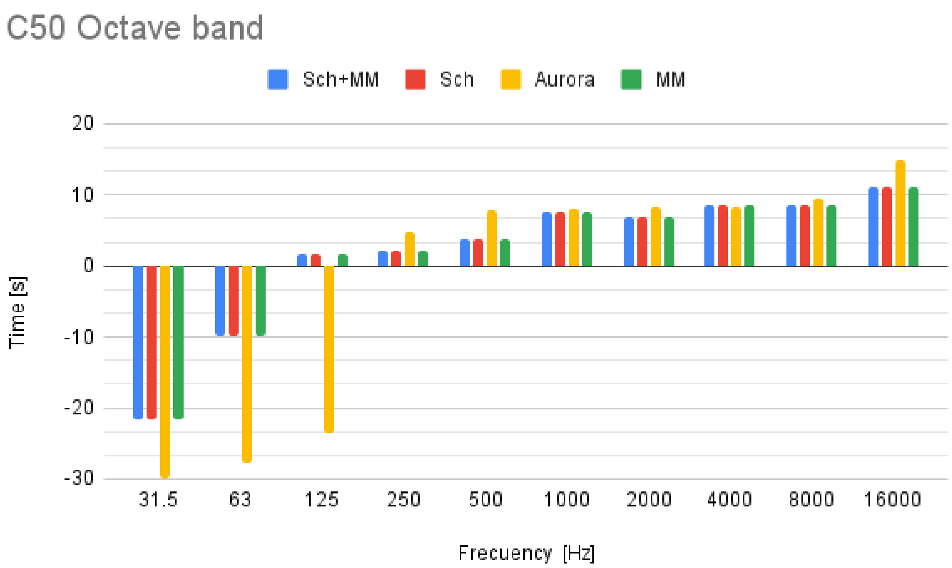


Figure 6. C50 descriptor per octave band.

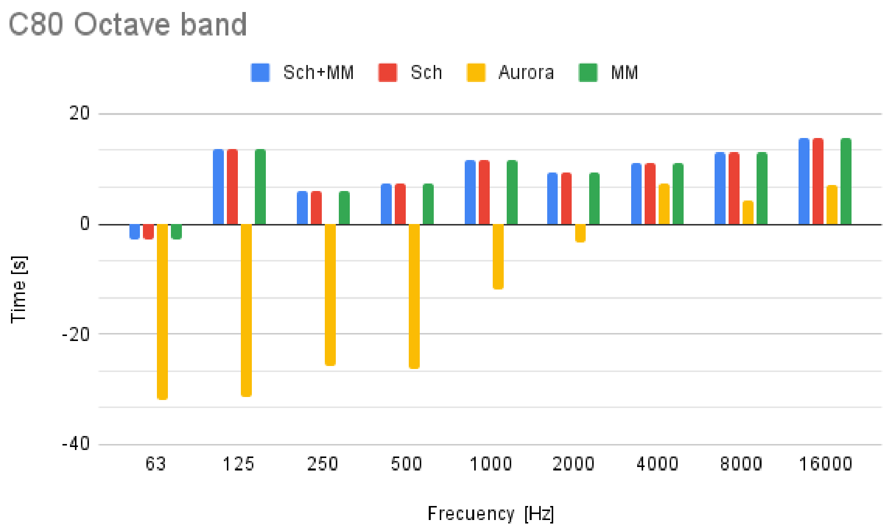


Figure 7. C80 descriptor per octave band.

In the case of C50 in Figure 6, there is a greater incongruity in low frequencies between both softwares. From 500 Hz the time difference is minimal. However, for the descriptor C80, Figure 7 shows large inconsistencies in the values calculated by the software developed with respect to Aurora. Differences greater than approximate 25 seconds are recorded for the 63-500 Hz range.

In the case of Tt and EDTt, a comparison cannot be made with the commercial Aurora software because it does not calculate these parameters, the differences between filtering by the moving median, by Schroeder or by both are outlined in Figure 8 for EDTt.

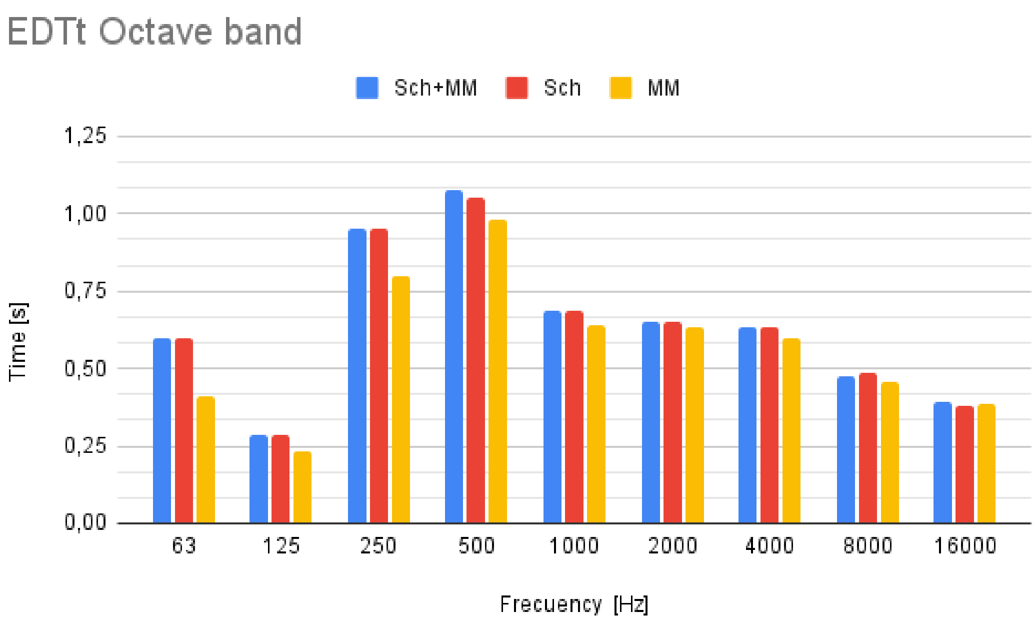


Figure 8. EDTt descriptor per octave band with different smoothing.

4.2 Sources of error and possible improvements.

Two specific sources of error can be identified. The first is associated with the user. If the user does not know how to or incorrectly enters the parameters required by the interface, such as the data to generate the inverse filter of the sine sweep, errors will occur in the calculation of the acoustic parameters.
On the other hand, there are errors associated with the processing of the interface. One of them relates to the processing time for thirds octave filtering and moving median smoothing.
The above is related to the processing time required by the function through the Ndimage library. It is a function primarily intended for image processing and has a long delay in filtering time.
Finally, there are problems due to the lack of warnings and restrictions to the user in the order of selection of the interface functions. If a mistake is made in selecting any function, it is necessary to start the interface again, the options are not automatically reset.
These aspects are future improvements to be made.

5. Conclusions

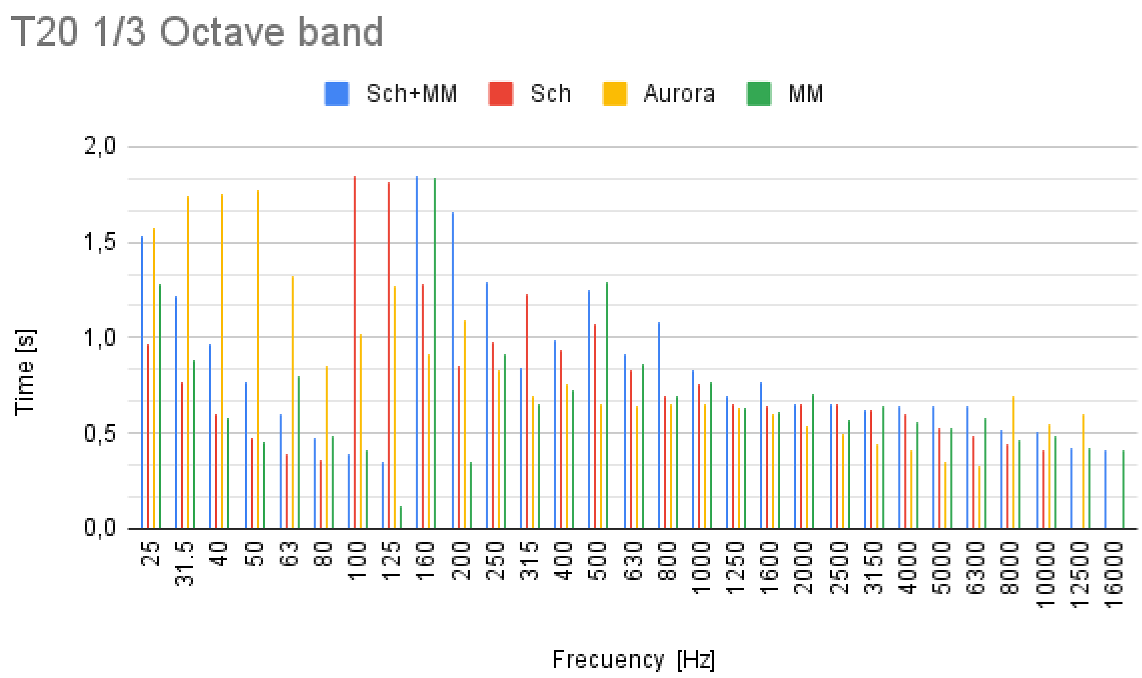
The development of a graphical user interface that allows the calculation of the main acoustic descriptors of a room could be successfully implemented.
It was possible to visualize the difference in the values calculated by the code developed for each descriptor compared to commercial software.

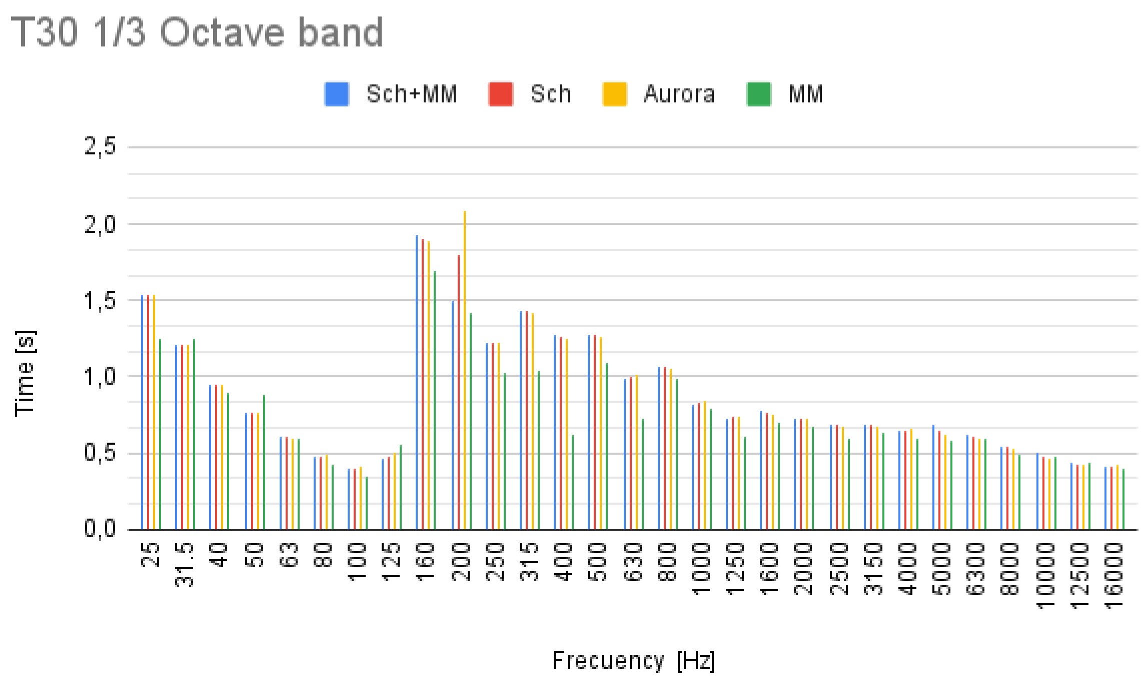
6. References

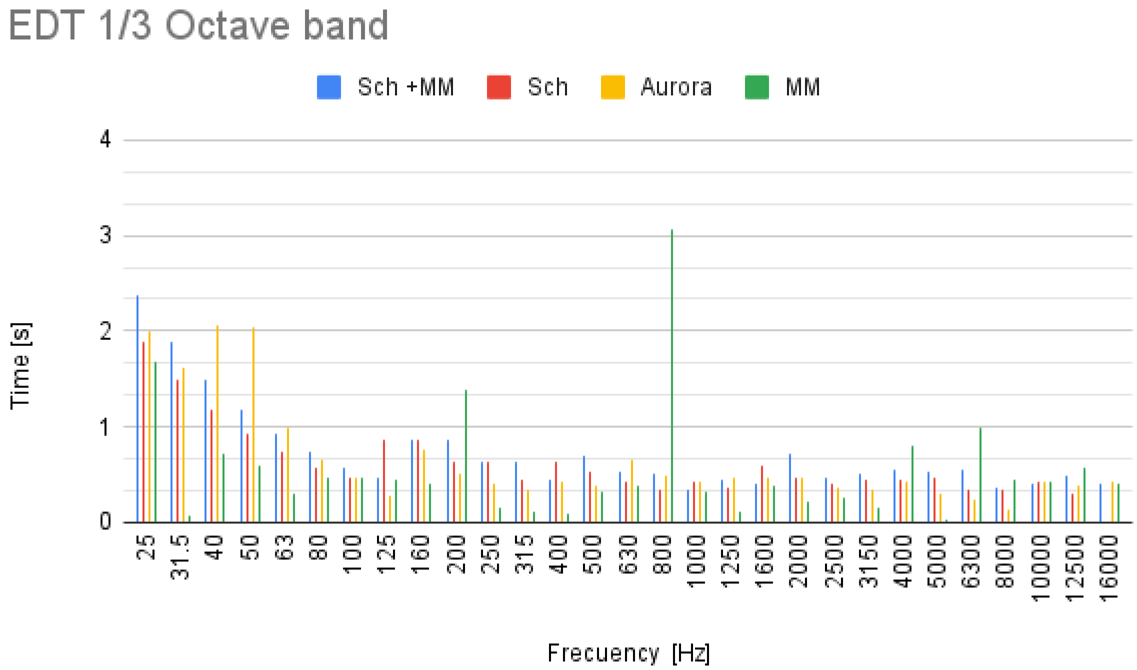
[1] ISO 3382:2001. Acoustics Measurements of the Reverberation Time of Rooms with Reference to other Acoustical Parameters. International Organization for Standardization, 2001.
[2] Martin Guski, Michael Vorländer. Measurement Uncertainties of Reverberation Time caused by Noise.
[3] Angelo Farina. Advancements in impulse response measurements by sine sweeps. In Audio Engineering Society Convention 122. Audio Engineering Society, 2000.
[4] Angelo Farina. Simultaneous measurement of impulse response and distortion with a swept sine technique. In Audio Engineering Society Convention 108. Audio Engineering Society, 2000.
[5] Kalis Jorinus, Macleane Lean, Möllenkramer Bas. A Practical Approach to Assessing Reverberation Time Measurement Uncertainty. June 19, Madrid.
[6] Rasmussen, B., Rindel, J. H., & Henriksen, H. (1991). Design and measurement of short reverberation times at low frequencies in talks studios. Journal of the Audio Engineering Society, 39(1/2), 47-57.
[7] Alejandro Bidondo, Leonardo Pepino. Room acoustic texture: a methodology for its quantification. 23rd International Congress on Acoustics. September 2019, Aachen, Germany.
[8] Schroeder, Manfred R. New Method of Measuring Reverberation Time [PW Smith, Jr., J. Acoust. Soc. Am. 37, 402-412 (1965)]. The Journal of the Acoustical Society of America, 37(2):402 412, 1965.
[9] Matti Karjalainen, Aki Mäkivirta. Estimation of Modal Decay Parameters from Noisy Response Measurements.
[10] A., Bidondo; L., Shtrepi; L., Pepino; A., Astolfi. Analysis of the room acoustic texture parameters in function of diffusers location and distribution inside a small concert hall. International Symposium on Room Acoustics. September 2019, Amsterdam, Netherlands.
[11] R.,Shimokura; A., Cocchi; L. Tronchin; M., Consumi. Calculation of IACC of any music signal convolved with impulse responses by using the interaural cross-correlation function of the sound field and the autocorrelation function of the sound source. Bologna University, Viale Risorgimento 2, 40136 Bologna, Italy.
[12] DETERMINACIÓN DEL JUST NOTICEABLE DIFFERENCE (JND) DE PARÁMETROS ACÚSTICOS MEDIANTE CAMPOS SONOROS REALES
[13] IEC 61260:1995, Electroacoustics. Octave band and fractional octave band filters. Standard, International Electrotechnical Commission, febrero 1995.

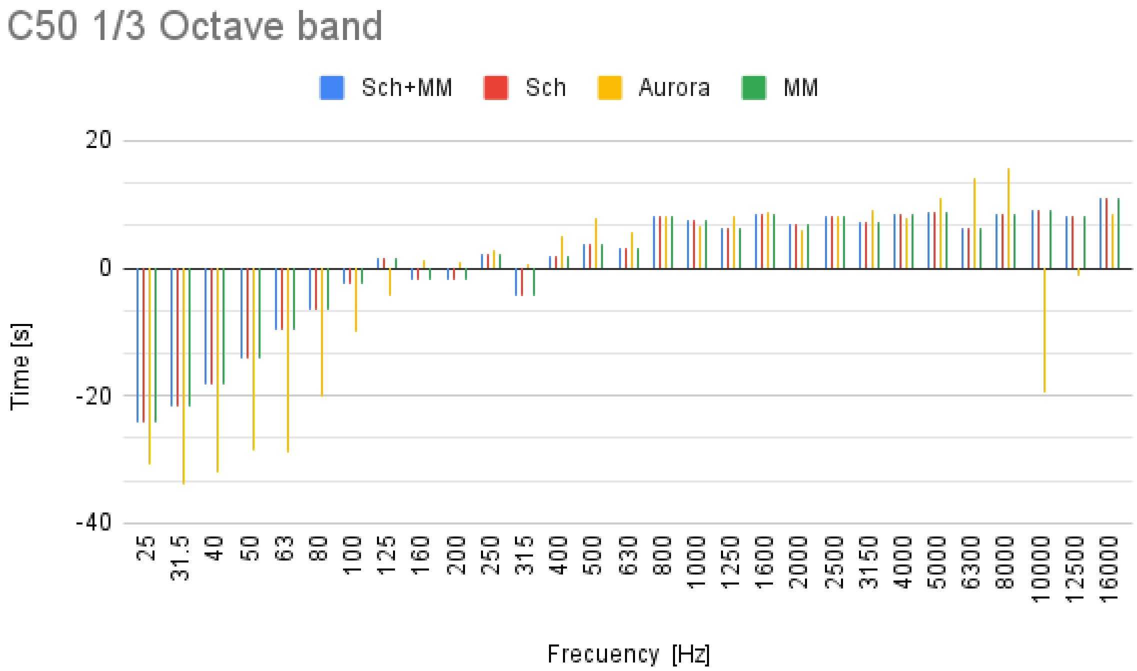
Annex A

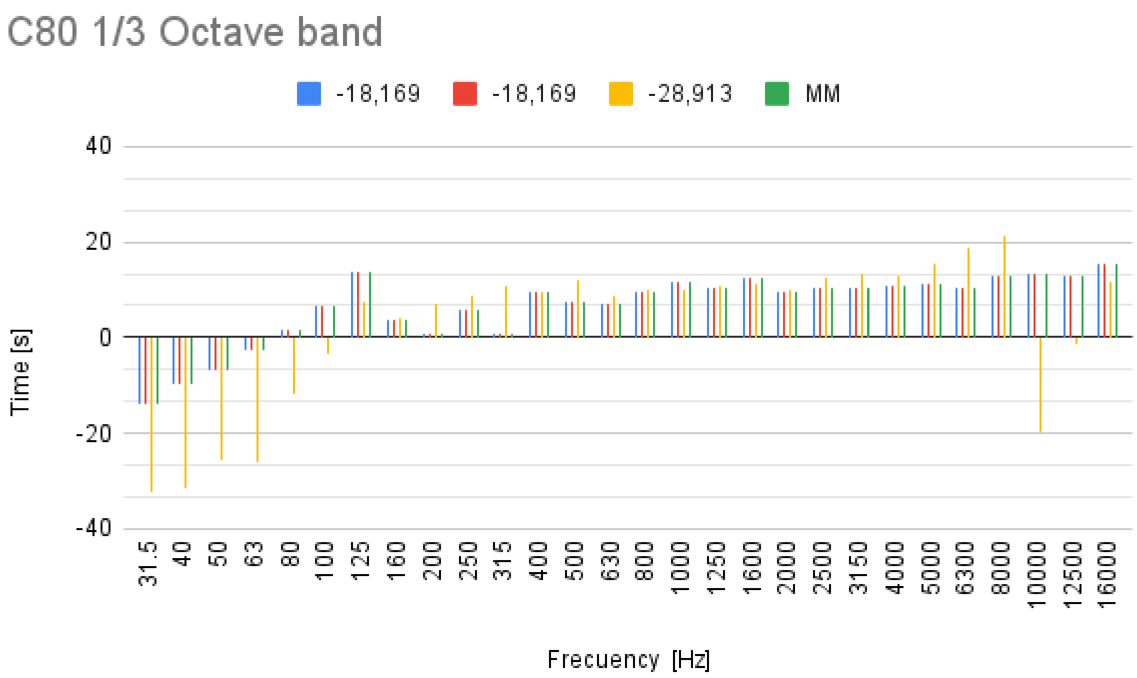
In this annex, the graphs corresponding to the acoustic descriptors calculated by thirds of octaves are presented. These parameters are calculated with the developed software and with the commercial Audacity software.

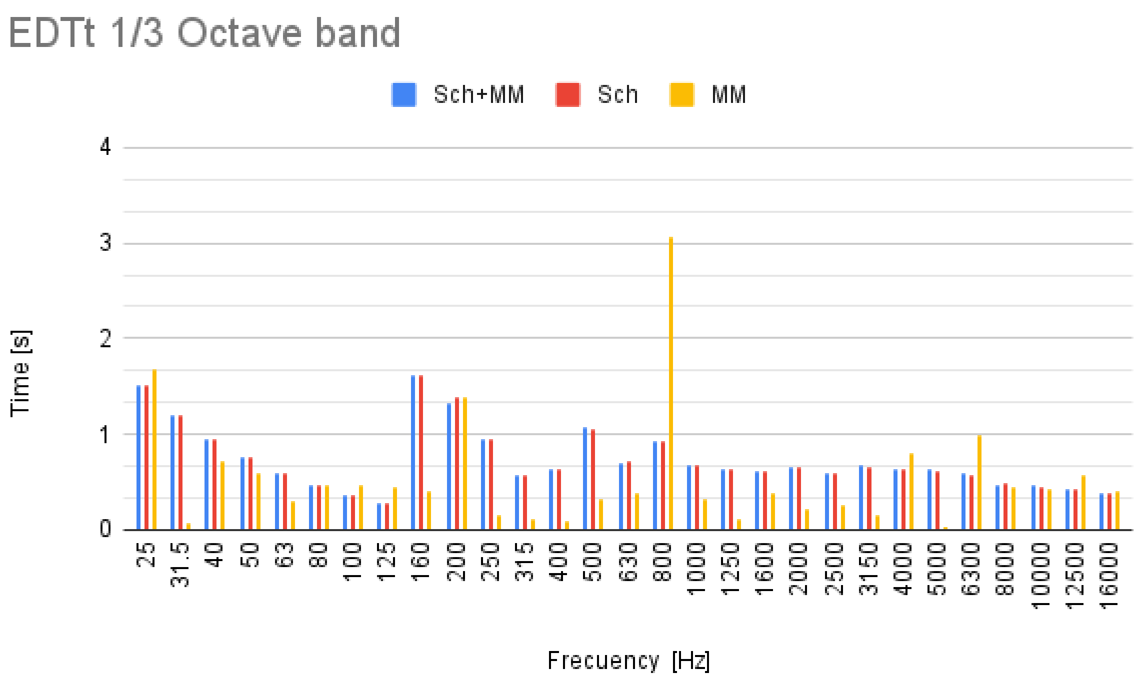





EDT 1/3 Octave band
Sch +MM
Sch
Aurora
MM
Time [s]
4
3
2
1
0
25
31.5
40
50
63
80
100
125
160
200
250
315
400
500
630
800
1000
1250
1600
2000
2500
3150
4000
5000
6300
8000
10000
12500
16000
Frecuency [Hz]


C50 1/3 Octave band
Sch+MM
Sch
Aurora
MM
Time [s]
20
0
-20
-40
25
31.5
40
50
63
80
100
125
160
200
250
315
400
500
630
800
1000
1250
1600
2000
2500
3150
4000
5000
6300
8000
10000
12500
16000
Frecuency [Hz]


C80 1/3 Octave band
-18,169
-18,169
-28,913
MM
Time [s]
40
20
0
-20
-40
31.5
40
50
63
80
100
125
160
200
250
315
400
500
630
800
1000
1250
1600
2000
2500
3150
4000
5000
6300
8000
10000
12500
16000
Frecuency [Hz]


EDTt 1/3 Octave band
Sch+MM
Sch
MM
Time [s]
4
3
2
1
0
25
31.5
40
50
63
80
100
125
160
200
250
315
400
500
630
800
1000
1250
1600
2000
2500
3150
4000
5000
6300
8000
10000
12500
16000
Frecuency [Hz]