SOFTWARE DEVELOPMENT FOR RIRS PROCESSING

Pedro Johansen
Universidad Nacional de Tres de Febrero, Ingeniería de Sonido, Caseros, Buenos Aires, Argentina
johansen30712@estudiantes.untref.edu.ar

Fernando Marino
Universidad Nacional de Tres de Febrero, Ingeniería de Sonido, Caseros, Buenos Aires, Argentina

fernandomarino95@gmail.com

Nicolás Zelener
Universidad Nacional de Tres de Febrero, Ingeniería de Sonido, Caseros, Buenos Aires, Argentina

nicolaszelener@gmail.com

The aim of this study is the development of a software for room impulse response processing. The software consist in an interactive user interface that allows the user to enter a recorded room impulse response in monophonic or stereophonic format and obtain its acoustical parameters. The software allows to process the RIR by applying noise correction and smoothing in order to obtain the parameters EDT, T20, T30, C50, C80, Tt, EDTt and IACC$_{early}$. Once the RIR processing is done, graphs and tables with the results are generated and can be exported. Finally, the obtained results from the software developed are compared with the results obtained from the Aurora module for Audacity.

Keywords: Room Impulse Response, Acoustical Parameters

1. Introduction

In the study and acoustic design of enclosures, the rooms conceived for music and speech are of maximum complexity due to the sound transmission process itself and, fundamentally due to the response of listeners to the perceived signal that in the case of music is increased by the different demands of each type of music for optimal listening. The study of the correlation between objective measurements and subjective evaluations, gives rise to a large number of parameters for determining the acoustic quality of rooms.

There are several reasons to measure reverberation time, the first being that the sound pressure level from noise sources, the intelligibility of speech, and the perception of intimacy in a room are strongly dependent on reverberation time. Rooms may include domestic rooms, stairways, workshops, industrial plants, classrooms, offices, restaurants, exhibition centres, sports halls, and railway and airport terminals. Second, reverberation time is measured in order to determine the correction term for room absorption inherent in many acoustic measurements, such as sound insulation and sound power measurements.

In some jurisdictions, building codes specify the required reverberation times in classrooms and other types of rooms. However, in the vast majority of enclosures, it is left to the design team to specify and design for a reverberation time that is appropriate for the purpose of a room.

For this reason, in this work a software capable of processing impulse responses of any type of room was developed. This tool aids with the calculation of different acoustical parameters, frequently used in the characterization of rooms. As a method of validation, the results obtained are compared with other similar applications designed for the same purpose.

2. Theoretical framework

2.1 Room Impulse Response

In order to acoustically characterize a room it is necessary to possess its impulse response. Every acoustical parameter of a given environment can be derived by obtaining and processing its RIR (room impulse response). This is obtained by exciting the room with a short-time, broad-band impulse (ideally this input signal should be a Dirac delta) and recording the enclosure's response. Because the emission of such signal at high levels can damage the loudspeakers, multiple methods of impulse generation were developed. One of them is the *logarithmic sine-sweep (LSS)* method, which generates an impulse through of the convolution of a sine-sweep and its inverse filter. First, the sine-sweep is played and recorded in the room, capturing the acoustic characteristics of the environment. Following, this recording is convolved with the appropriate inverse filter signal to generate the RIR. The sine-sweep is a sinusoidal signal whose frequency increases over time. Throughout the range of excitation, one frequency is excited at a time, increasing the density of energy. As a consequence of this, the signal-to-noise ratio rises [1]. The sine sweep signal is given by:

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

where W_1 is the initial angular frequency, W_2 the final angular frequency and T is the total duration of the sine-sweep. The inverse filter is the input signal inverted along the time axis. Thus, the instantaneous frequency decreases with time. In case of the exponential sine sweep, the amplitude modulation is added to compensate the energy differences between high and low frequencies. The inverse filter is described by the following equation:

$$w(t) = \frac{TW_1}{ln(\frac{W_2}{W_1})}(e^{\frac{\frac{T}{t}}{ln(\frac{W_2}{W_1})}} - 1) \tag{2}$$

2.2 Moving median filter

The moving median filter is a non-linear digital filtering technique, often used to remove noise from an image or audio signal. Such noise reduction is a typical pre-processing step, usually done to improve the results of later processing. This method is useful for reducing periodic patterns, random noise and it works especially well when the noise amplitude probability density function presents long tails. The median filtering process is accomplished by sliding a rectangular window over the signal and calculating the median of the values in said window. The obtained values are placed at the location of the center of the window.

$$Med(X) = \begin{cases} X[\frac{n}{2}] & \text{if } n \text{ is even} \\ \frac{X[\frac{n-1}{2}]+X[\frac{n+1}{2}]}{2} & \text{if } n \text{ is odd} \end{cases} \tag{3}$$

2.3 Schroeder filter

Schroeder's backwards integration is a method derived by integrating the squared impulse response gathered from any room and is being used for a long time for the estimation of reverberation time. The formula of Schroeder's backwards integration is:

$$E(t) \; = \int_t^{\infty} h^2(t)\; dt \; = \int_0^{\infty} h^2(t)\; dt \; - \int_0^{t} h^2(t)\; dt \tag{4}$$

where $h(t)$ is the impulse response function.

2.4 Noise correction

The limits of integration of the smoothing stage, like the Schroeder inverse integral, can be obtained through Lundeby's algorithm [2], an iterative process that looks for the optimal limits of integration by measuring the noise level and adjusting the calculations on each iteration. This procedure determines the integration point by measuring the noise level and slope of the decay and adjusting them trough multiple repetitions. The integration limit is finally defined at the intersection of the linear decay line and the noise level.

2.5 Reverberation Time

Reverberation time (RT or T60) is defined as the time required for the sound pressure level to decrease 60 dB [3]. There are a plethora of formulas that can be used to estimate the RT of a room, like the Sabine (5) or Eyring formulas (6).

$$RT = \frac{0.161\; V}{S_{tot}\alpha_{tot}} \tag{5}$$

$$RT = \frac{0.161\; V}{-S_{tot}\; log(1-\alpha_{tot})} \tag{6}$$

where V is the volume of the enclosure in cubic meters, S is the total surface of the room and α_{tot} is the total absorption coefficient, defined as

$$\alpha_{tot} = \frac{\sum_i \alpha_i S_i}{S_{tot}} \tag{7}$$

When measuring this parameter, a high noise floor or the inability to get the output of the speaker to be above a certain level might hamper the capacity to get a signal-to-noise ratio greater than 60 dB. This is when the T_{20} and T_{30} parameters should be used. These parameters require a lower SNR, so they allow for more versatility in the measurement conditions. Both the T_{20} and the T_{30} assume that the energy decay is linear, and therefore the T_{60} can be obtained by measuring the time of the initial 20 dB (or 30 dB in the case of the T_{30}) decay and extrapolating the curve further in time.

All the previously mentioned parameters start the measurement at the -5 dB point in the energy time curve and finish at the -25, -35 or -65 dB point.

The early decay time (EDT) parameter is a more accurate measure of the perceived reverberance of a room [4]. It measures the time taken by the energy curve to decrease 10 dB. Unlike the T20, T30 or T60, this decay is measured starting from 0 dB down to the -10 dB point.

2.6 Acoustical Parameters

The C_{50} and C_{80} parameters are defined by equations (8) and (9). These are a measure of the clarity of the sound, defined as a ratio of the early and late energy [5]. The limits of 50 and 80 ms pertain to speech and musical clarity respectively.

$$C_{50} = 10\; log\left(\frac{\int_0^{50ms} h(t)\; dt}{\int_{50ms}^{\infty} h(t)\; dt}\right) \quad [dB] \tag{8}$$

$$C_{80} = 10\,log\left(\frac{\int_0^{80ms} h(t)\,dt}{\int_{80ms}^{\infty} h(t)\,dt}\right) \quad [dB] \tag{9}$$

The interaural cross-correlation coefficient IACC is defined as the maximum value of the interaural cross-correlation function (IACF) [6]. This parameter measures the similarity between the left and right channels in a stereo signal, attributed to the perception of the left and right ears in the human hearing system. It represents the diffuseness of the sound source, where a low IACC value corresponds to a wide source and a high value corresponds to a narrow one. Studies carried out by Ando et al. showed that dissimilarities between the signals arriving at both ears, represented by low IACC values, are associated with an increase in the perception of quality of the sound. There exists a variation of this parameter, called the early interaural cross-correlation coefficient $IACC_e$, that is calculated along the first 80 ms of the signal.

The crossover time, or transition time T_t, is defined as the instant when the impulse response transitions from a deterministic regime to a stochastic one [7]. This means that a RIR contains its information in the interval that extends from the initial instant up to the transition time. The portion of the signal succeeding the T_t containing the tail of the reverberant field consists of amplitude-modulated gaussian white noise.

3. Software Development

3.1 User Interface

The software consists of 3 Python scripts: The user interface script which was generated using the *QT Designer tool*, the script that carries out the calculations to obtain the acoustical parameters of the loaded RIR and the main app script that communicates the user interface with the calculations script.

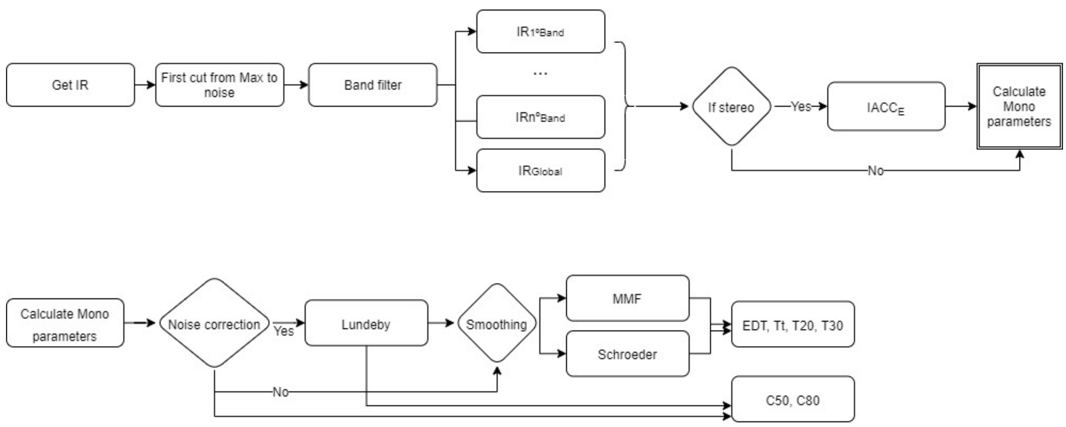
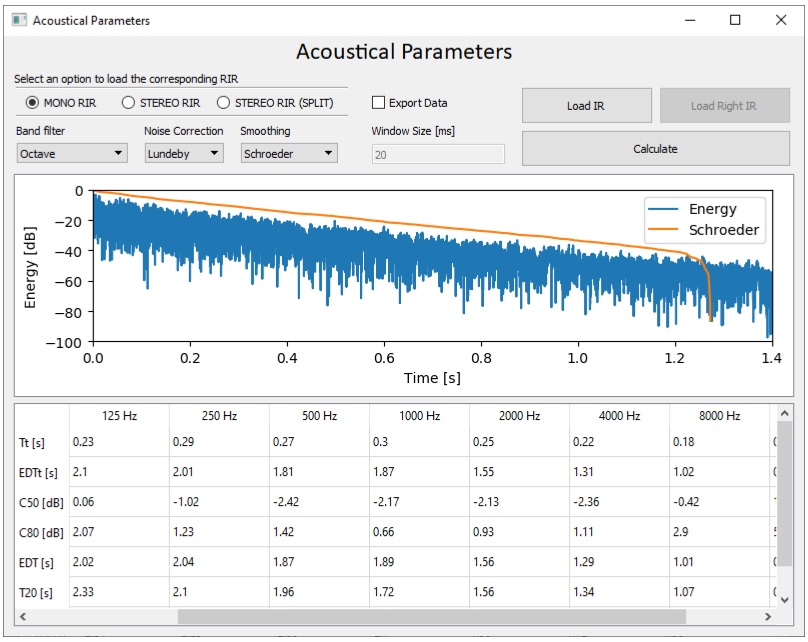


Figure 1: Main App structure.

Once the main app script is initialized, the interface allows the user to select the type of impulse response that is going to be loaded. There are three types of RIR: a mono RIR, a stereo RIR and a stereo RIR with its channels split into left and right in two different audio files. The GUI has two buttons for loading RIRs. In the case of loading mono or stereo RIRs the user must click on the "Load IR" button that opens a Windows file dialog to look for the corresponding RIR audio file on the hard drive. Otherwise, if the user wants to load a split stereo RIR, they must click on the two buttons that load the left and right channels of the impulse response. These buttons are "Load Left IR" and "Load Right IR". By clicking each of this buttons, a Windows file dialog will be open in order to search the RIR file on the hard drive.

Once the signal is loaded into the software, the interface offers configuration options that modify the processing of the RIR to obtain the acoustical parameters. The application allows the user to adjust the filtering, the smoothing and also the noise correction of the processing of the impulse. Finally, when the "calculate" button is clicked, the result area of the GUI displays both the plot of the processed signal and the table containing the acoustical parameters. There is also a checkbox that allows the user to export the tabular data into a CSV file. A screenshot of the user interface is shown in the Figure .



	125 Hz	250 Hz	500 Hz	1000 Hz	2000 Hz	4000 Hz	8000 Hz
Tt [s]	0.23	0.29	0.27	0.3	0.25	0.22	0.18
EDTt [s]	2.1	2.01	1.81	1.87	1.55	1.31	1.02
C50 [dB]	0.06	-1.02	-2.42	-2.17	-2.13	-2.36	-0.42
C80 [dB]	2.07	1.23	1.42	0.66	0.93	1.11	2.9
EDT [s]	2.02	2.04	1.87	1.89	1.56	1.29	1.01
T20 [s]	2.33	2.1	1.96	1.72	1.56	1.34	1.07

Figure 2: Software's user interface.

3.2 Filtering

In order to obtain the acoustical parameters of each loaded RIR, the software applies a window from the maximum value of the RIR to 10 seconds after. Then filters it with Butterworth passband filters, designed according to the UNE-EN 61260 standard [8]. These filters are designed and applied by using the signal function from the *Scipy* library. The user can choose the type of filtering, either by octave band or by one third octave band filters.

3.3 Smoothing

For the signal smoothing process the user may choose between two different methods. The first is the Schroeder's backwards integration. The software calculates this integral by using the *Numpy* library, which allows for the reversal of the time interval and the calculation of the cumulative sum. The second method that the software offers is the moving median filter, which calculates the median over a window with a size previously determined by the user.

3.4 Noise Correction

The algorithms for the determination of room acoustical parameters used by different analyzers introduce systematic differences caused by differences in time-windowing and filtering, in reverse-time integration and in noise compensation. The GUI offers the option to not apply any correction or apply Lundeby's correction algorithm. The latter determines an integration interval through by measuring the noise level and decay slope, and adjusting said values through an iterative process.

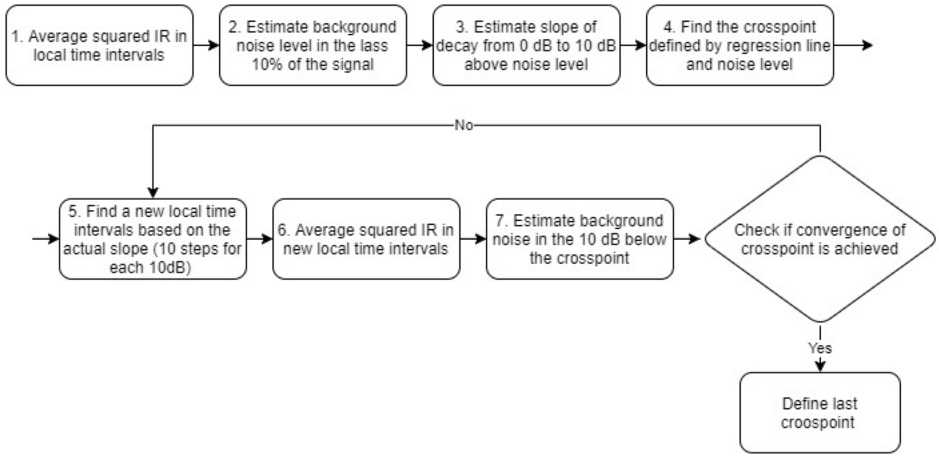


Figure 3: Lundeby's recursive algorithm.

3.5 Acoustical parameters calculation

The acoustical parameters calculation stage is divided in three functions (four in the case of the stereo RIR). These functions calculate the reverberation time parameters EDT, T_{20} and T_{30}, the clarity parameters C_{50} and C_{80} and the time parameters T_t and EDT_t. These functions take the filtered IR as an input. For a stereo RIR analysis an extra $IACC_e$ function is added to the process, that takes both channels as an input. The principal Python library during this process is *Numpy*.

The previous functions were included in a parent function that takes care of the necessary pre-processing of the signal and packages the results into a Python *dictionary* that contains all acoustical parameters per frequency band. In order to facilitate the plotting stage, the energy time curve and smoothed signal are appended to the results dictionary.

4. Results

For this section, stereo and mono RIRs from the Teatro Colón were taken in order to validate the functionality of the software. In that way, the acoustical parameters were calculated with the developed software. The parameters were obtained using both smoothing methods to compare the respective results. In case of the moving median filter smoothing the window size was set to 20 ms, because it was a value that provided enough filtering to smooth the signal and enough resolution to contemplate the 63 Hz octave band. This last condition stems from the fact that a window size of 20 ms contains enough information to represent half a period of a 25 Hz wave, which means that the minimum valid frequency is 25 Hz. Considering that the lower bound of the 63 Hz octave band is 44 Hz, the information obtained after processing the signal with the moving median filter is valid.

The octave bands considered in this analysis are the ones comprised from 63 to 8000 Hz. In the absence of the original sine-sweep utilized during the measurement (or even the inverse filter), the limit was derived from the bands that showed coherent results returned by the Aurora plug-in. These are either null data points or unreasonable values. For example, knowing that the Teatro Colón is an opera house with an average reverberation time of 1.6 seconds [9], a T_{30} measurement of 4 seconds would discard that band.

Also, to compare and validate the obtained results, the acoustical parameters of each RIR were calculated with the *Acoustical Parameters* module from the Aurora plugin for Audacity. This module was configured to apply Schroeder backwards integration to the analyzed RIRs. On the other hand, in order to make a reliable comparison, the processed RIRs in Aurora were windowed with the same window size as in the developed software in this work.

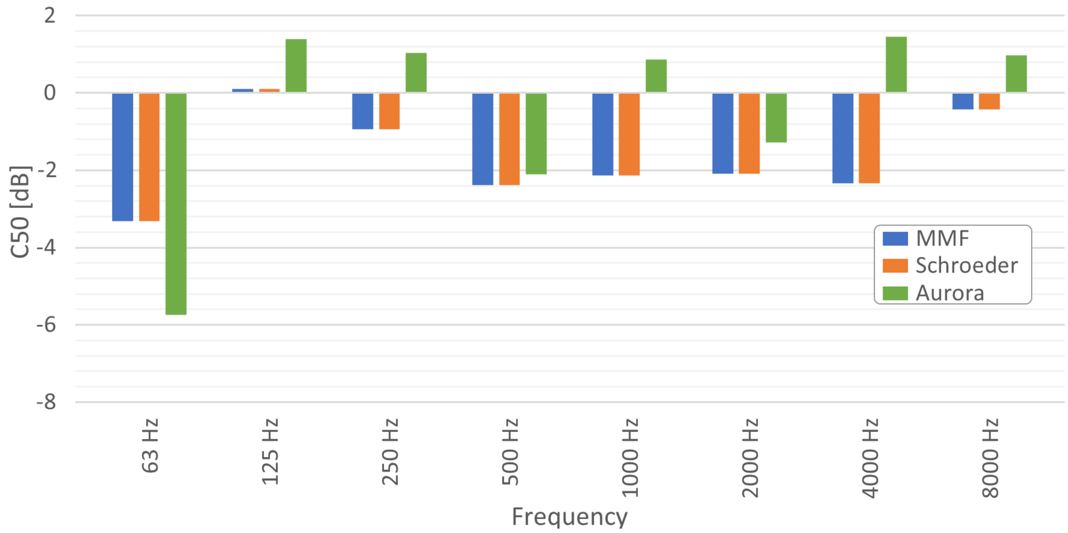


Figure 4: C50 filtered by octave bands

As the Figure 4 shows, there is no difference between the C_{50} values obtained by applying the Schroeder and moving median filter. This is because the software calculates this parameter before the smoothing process. On the other hand, the differences between the developed software and Aurora are less than 2 dB per each band except for the 63 Hz. On that band, the difference is 2.4 dB.

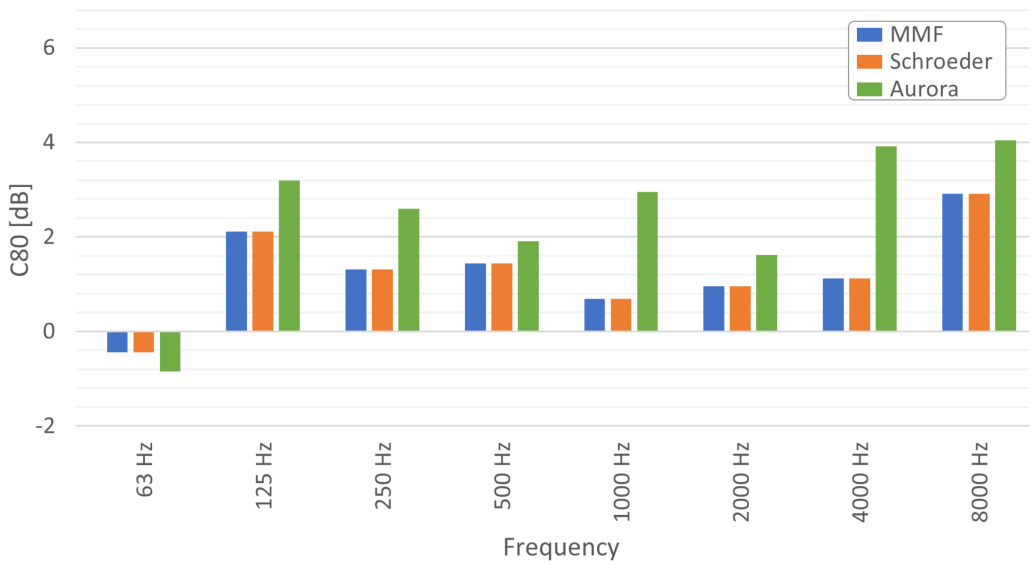


Figure 5: C80 filtered by octave bands

As with the C_{50} results, there is no difference between the smoothing methods of the software of this work, given that the operations are done over the raw ETC curve. The difference between this software and the Aurora module is less than 2.5 dB per each band except on 4 kHz, where the divergence rises to 2.8 dB.

The difference between the values of the clarity parameters obtained through the Aurora plugin and our software might be explained by the way each application manages the truncation of the signal. Given that these parameters calculate the ratio of the energy present in the early portion and the latter portion of the signal, if the RIR is windowed in a different way, the energy contained in the last section of the energy curve may change. This will inevitably result in a variation of the C_X parameters because the time interval of the early portion is fixed, so it is unaffected by the modified length of the tail.

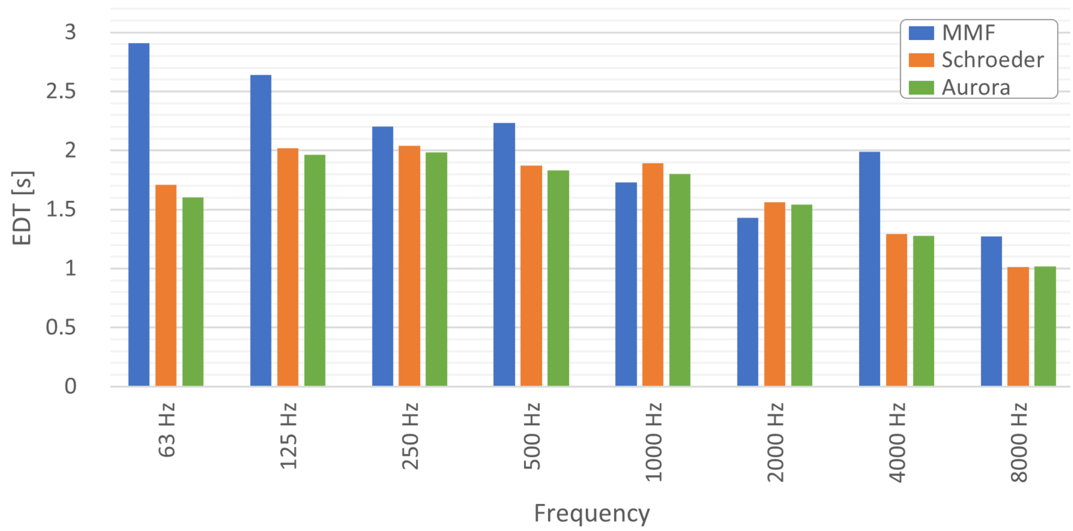


Figure 6: EDT filtered by octave bands

The Figure 6 shows the EDT values obtained through Aurora, and the Schroeder and moving median filter smoothing methods. The MMF method obtains longer decays than the Schroeder method in most of the frequency bands. On the other hand, the software, with the Schroeder smoothing applied, does not present significantly differences with the Aurora module.

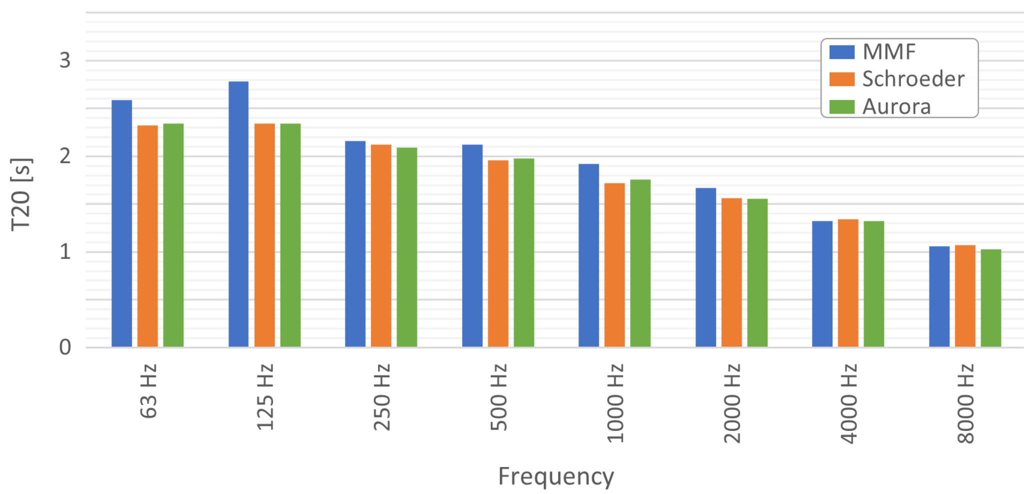


Figure 7: T20 filtered by octave bands

Regarding the T_{20} parameter, the MMF method shows differences with respect to the Schroeder smoothing but they are not as extreme as they were on the EDT values. The greatest deviations are seen in the lowest frequency bands of 63 and 125 Hz. In the rest of the bands the differences are not as significant. This could improve by increasing the window size of the MMF. The values obtained by using the Schroeder reverse integration method and the values obtained through the Aurora plug-in do not exhibit major differences. This suggests that the calculations carried out by Farina's tool are similar to the ones used in the software developed for this report. In other words, the Aurora extension might be employing Schroeder's reverse integral and some variation on Lundeby's noise correction algorithm in order to process the RIR.

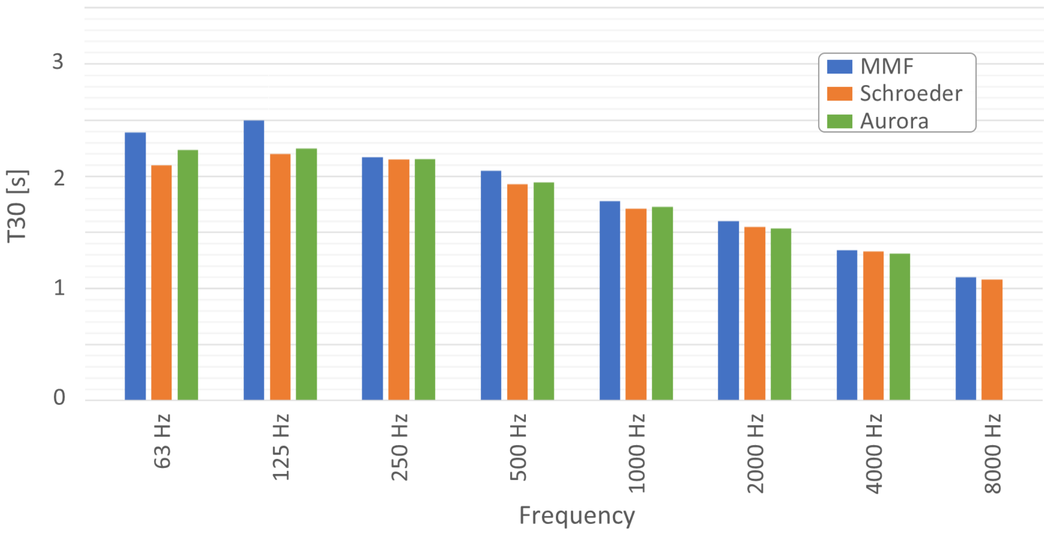


Figure 8: T_{30} values filtered by octave bands.

The results, seen in Table 1, show that the T_{20} values are slightly lower than the T_{30} ones. This is explained by the non linear nature of the impulse response's decay. The energy time curve does not decay at a constant rate, its decay rate varies, which means that the extrapolated reverberation time will be different depending on the interval taken into account. In this case, because the T_{30} is slightly higher than the T_{20}, it is possible to assume that this RIR presents a plateau-type decay. This type of energy curve has a slower decay at the beginning, so in consequence the T_{20} would return a higher reverberation time value, given that the first 20 dB decay at a lesser rate and it takes the curve more time to reach the 60 dB loss.

	Band [Hz]	63.5	125	250	500	1000	2000	4000	8000
Schroeder integral	T_{20} [s]	2.29	2.33	2.10	1.96	1.72	1.56	1.34	1.07
	T_{30} [s]	2.19	2.16	2.05	1.91	1.68	1.52	1.32	1.08
Moving median	T_{20} [s]	2.76	2.67	2.19	2.18	1.92	1.67	1.28	1.06
	T_{30} [s]	2.41	2.47	2.18	2.04	1.79	1.60	1.33	1.09

Table 1: Teatro Colón's reverberation time values per smoothing method.

In order to analyze the accuracy of the software, the reverberation time of three test IRs were measured. These signals are amplitude-modulated pure tones with a predetermined known decay time of 400 ms. By running them through the acoustical analysis tool we can detect whether the values obtained represent the behavior of the input signal or not. The tests were carried out using a 20 ms sized moving median filter window and using third octave filtering. The results are shown in Table 2. It can be seen that the moving median filter technique provides more accurate results than the reverse Schroeder integration. It is also evident that the calculation error is more noticeable for the T_{30} parameter.

	Moving median filter			Schroeder		
Band [Hz]	EDT [s]	T_{20} [s]	T_{30} [s]	EDT [s]	T_{20} [s]	T_{30} [s]
80	0.40	0.40	0.40	0.40	0.39	0.36
125	0.40	0.40	0.40	0.40	0.38	0.34
315	0.40	0.40	0.40	0.40	0.39	0.37

Table 2: Measured decay time values per smoothing method.

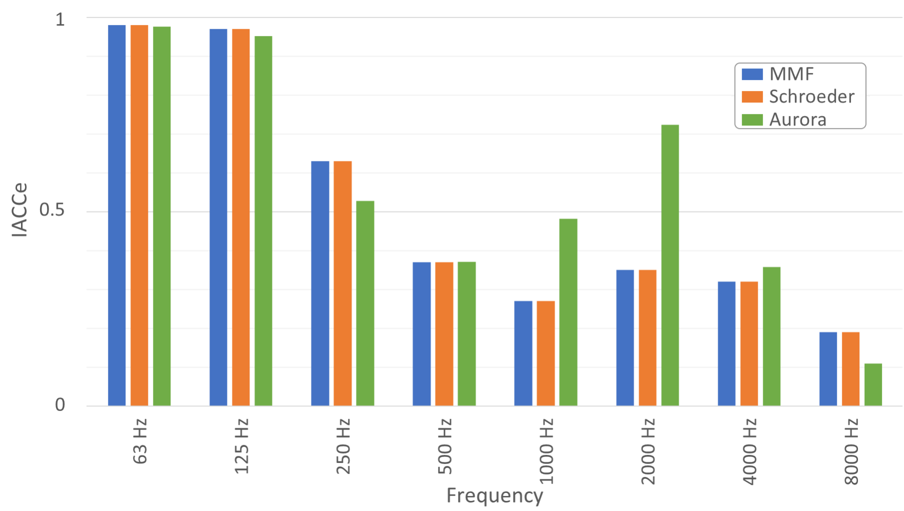


Figure 9: IACCe filtered by octave bands

Figure 9 shows the $IACC_e$ results. Differences are observed between the data provided by Aurora and those obtained by this application in all the bands analyzed. However, due to the structure of the designed analysis, this parameter is not affected by the chosen smoothing method. The auto correlation analysis is prior to that stage. Resulting the largest difference, by far, in the $2kHz$ band.

This parameter is the one that presents

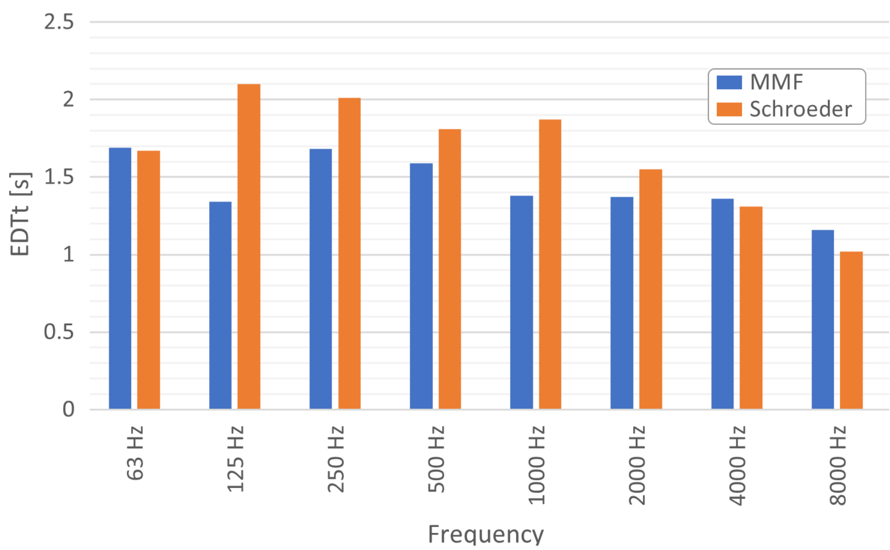


Figure 10: EDT_t filtered by octave bands.

EDT_t is a parameter that is not contemplated by the Aurora software, it is for this reason that only the results obtained through our application are shown in Figure 10. An early decay time estimate is observed with a time shorter than that indicated by the usual parameter.

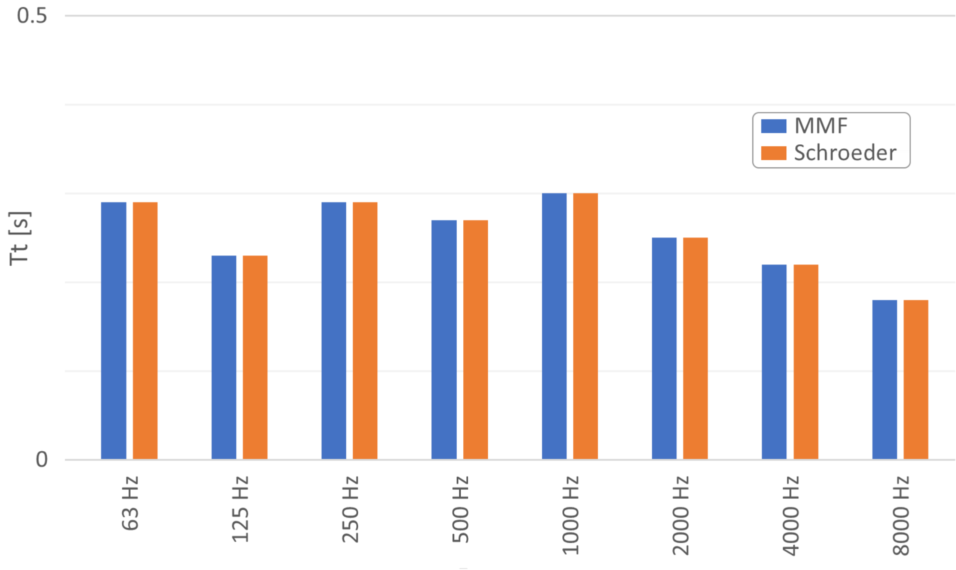


Figure 11: T_t filtered by octave bands

As can be seen in Figure 11, the values of the transition time are not affected by the smoothing of the signal because this parameter is calculated over the raw ETC. The transition times of the analyzed room are very short. Most of the energy is concentrated in the first moments of the impulse. Up to the $1kHz$ band, the time oscillates at $0.3s$, and from then on it gradually decreases. This characteristic marks the differences with the other early decay calculation.

5. Conclusions

After the comparisons were made, it was possible to identify through the clarity parameters that there are clear differences between the noise correction employed by this software and the one used by Aurora. It is possible that Aurora contemplates a single cross point for all frequency bands, while the one proposed in this work identifies the noise tail for each of them. It is for this reason that different jumps are observed between the values of each band for this type of parameters, in which the calculation of the length of the noise tail is a determining factor. The same effect is observed in the $IACC_e$ measurements.

The main problem found in the calculation of the different reverberation times is the absence of a method to evaluate the results obtained. This is implemented in Aurora by displaying an empty data cell instead of the incoherent values. An evaluation method for this type of error could be implemented for a future version of the software. Possibly a signal to noise ratio calculation would be helpful in this task.

When observing the results obtained in T_20 and T_30 between the two types of smoothing implemented by this software, it could be verified that the MMF is a more robust tool to determine the decay curve of the highest bands. The result obtained with the Schroeder equation shows an almost flat slope for the last band used, making it clear that a calculation is being carried out on a noisy signal.

When the EDT and the EDT_t are analyzed, it is observed that all the values of this parameter are significantly reduced. This is mainly caused by the non-uniformity of the decay curve in this particular room. When considering the main concentration of energy in the calculation of the early decay, the analysis focuses on the first slope of the line. This is much more pronounced at the beginning, and it is in this situation that it is interesting to discuss which portion of the impulse response best interprets the behavior of a room.

Possible extensions of the software are: the addition of more noise correction algorithms (for example Chu's or Hirata's), more smoothing methods (for example the moving mean filter) and the ability to visualize each band's decay curve.

REFERENCES

1. Farina, A. Simultaneous measurement of impulse response and distortion with a swept-sine technique. *Audio Engineering Society*, Paris, France, 19 - 22 February (2000)

2. Lundeby, A., Vigran, T.E., Bietz, H., Vorländer, M. Uncertainties of measurements in room acoustics, *Acustica*(81), 344-355, (1995).

3. Beranek L., Mellow, T. Acoustics: sound fields and transducers. *Academic Press*, London, UK, (2019).

4. Howard, D., Angus, J. Acoustics and psychoacoustics. *Focal Press*, Oxford, UK, (1996).

5. International Organization for Standardization. ISO 3382: Measurement of room acoustic parameters. (2008)

6. Ando, Y. Architectural acoustics. *Springer*, New York, NY, (1998).

7. Bidondo, A., Vazquez, J., Vazquez, S., Arouxet, M., Heinze, G. A new and simple method to define the time limit between the early and late sound fields. *Audio Engineering Society*, Los Angeles, CA, 29 September - 2 October (2016).

8. Asociación Española de Normalización y Certificación. UNE-EN 61260: Filtros de banda de octava y de bandas de una fracción de octava. (1997)

9. Bernaek L. Concert halls and opera houses. *Springer*, New York, NY, (2004).

6. Appendix

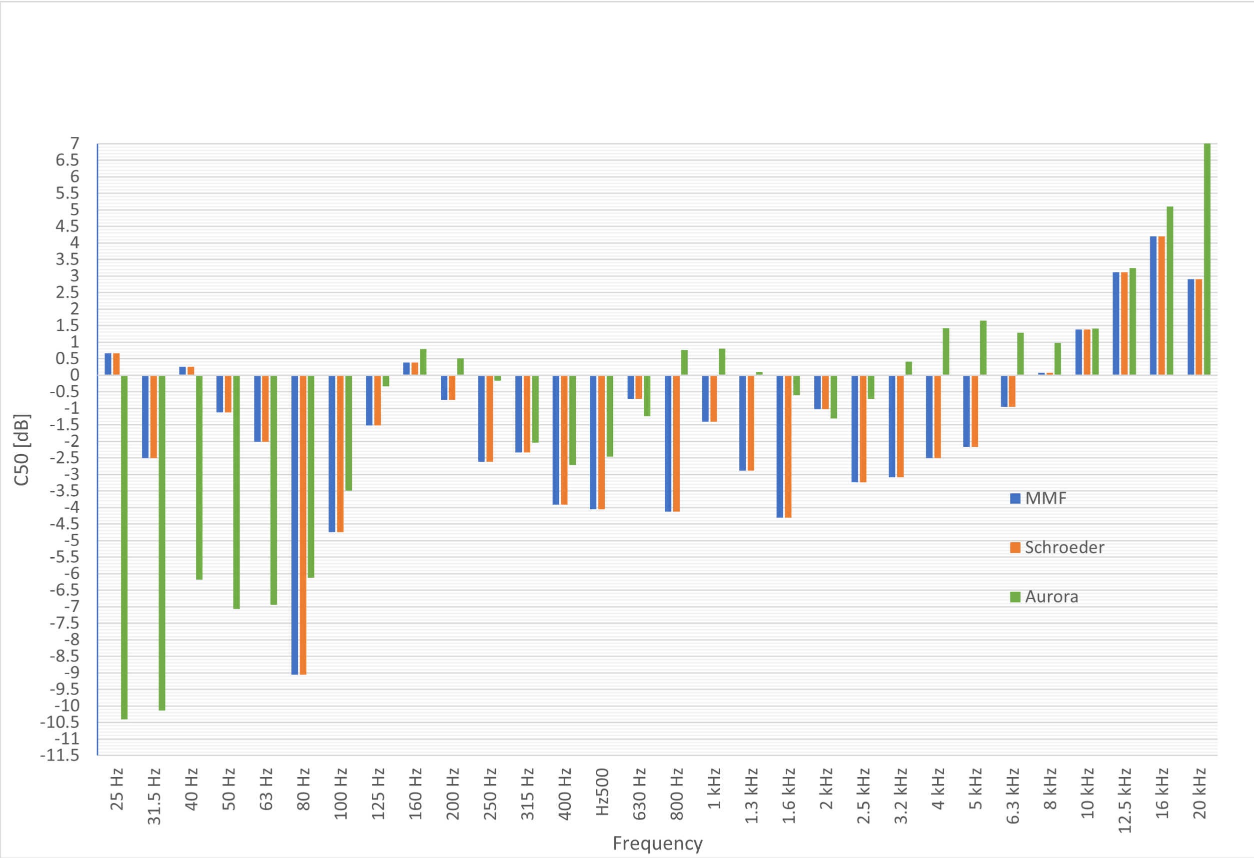


Figure 12: C50 by third octave bands.

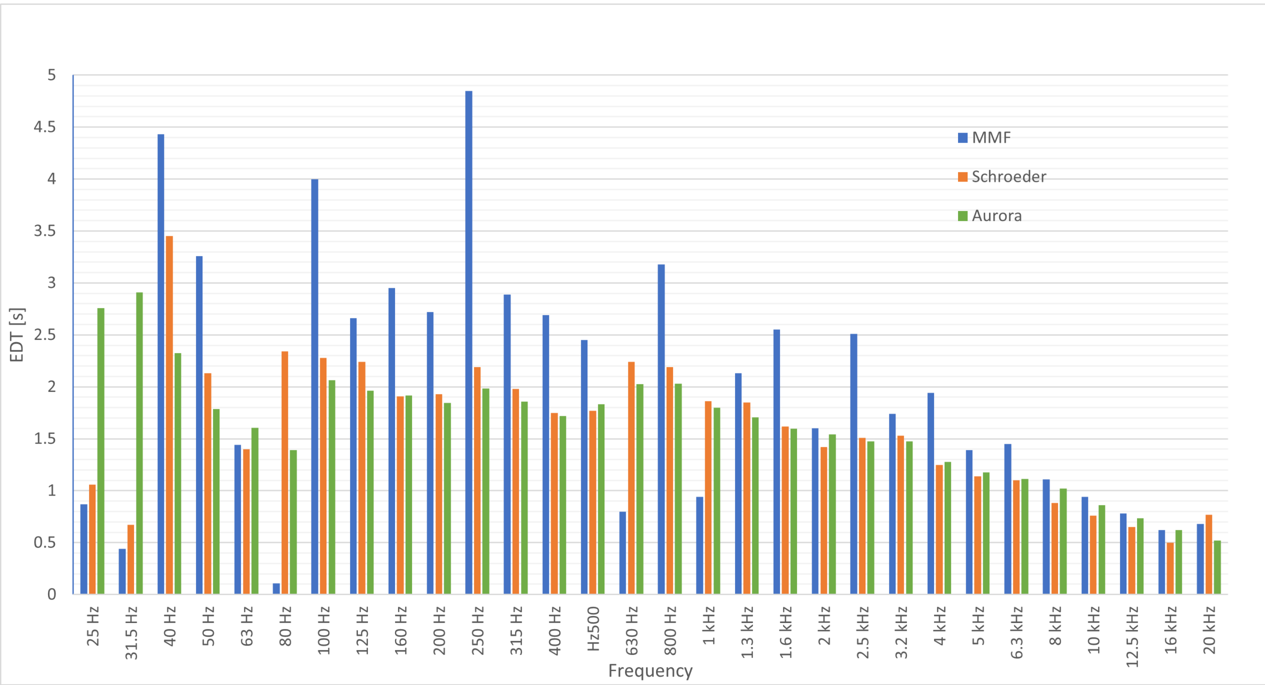


Figure 13: EDT by third octave bands.

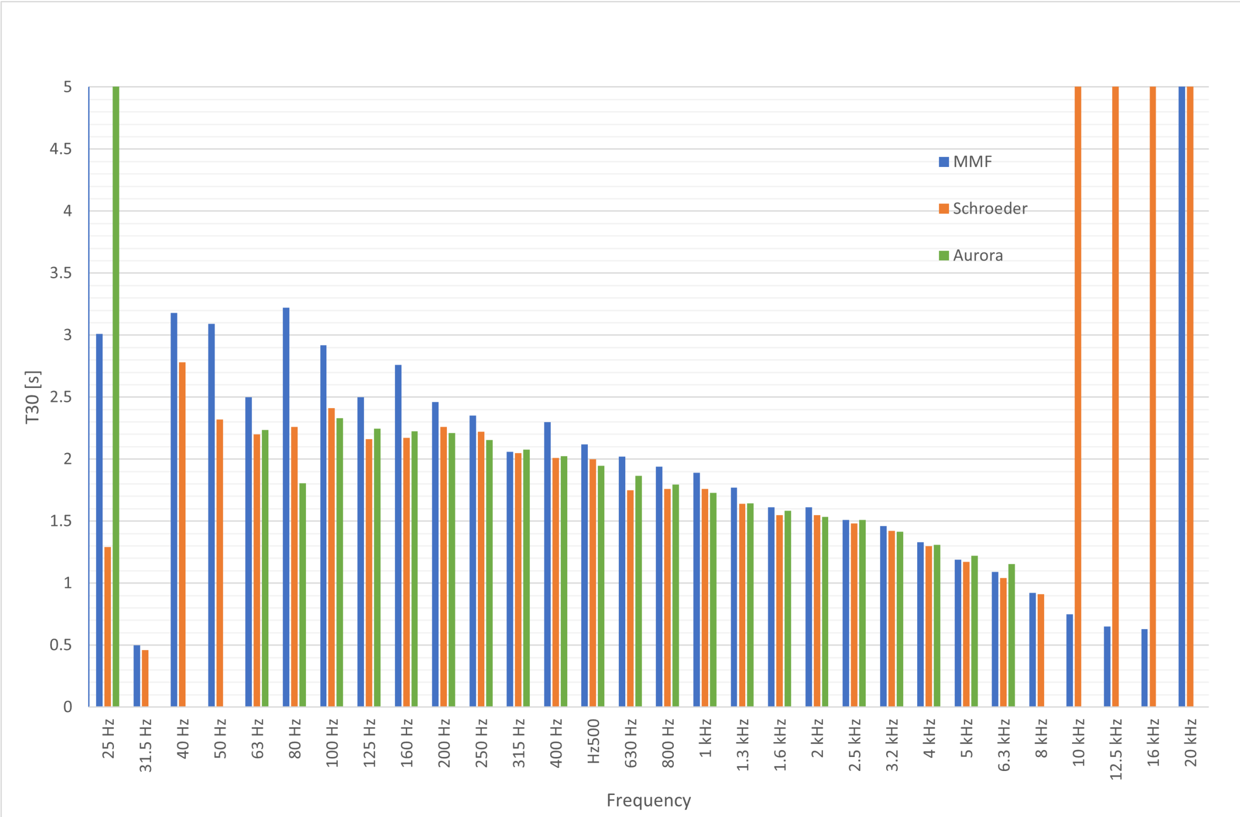


Figure 14: T30 by third octave bands.

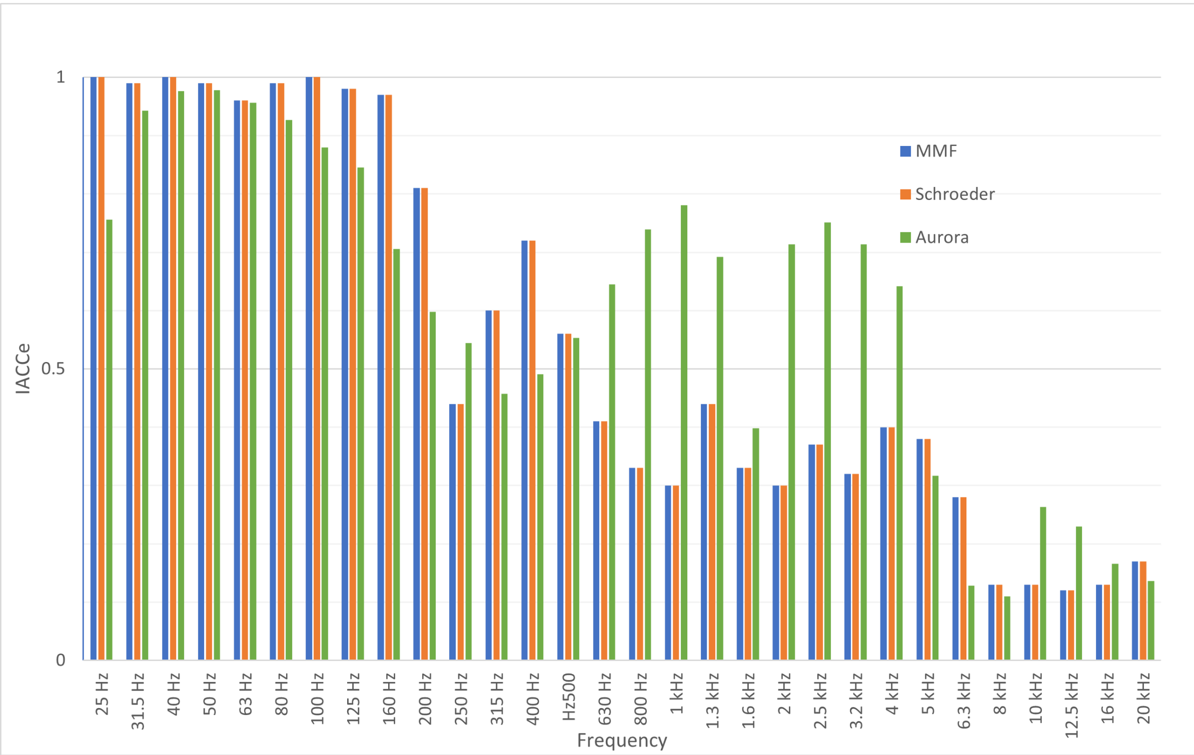


Figure 15: IACCe by third octave bands.