UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Superior d'Enginyeries Industrial,
Aeroespacial i Audiovisual de Terrassa

BACHELOR FINAL THESIS

Real time acoustic analysis and correction

Document:

Report

Author:

Alexandr Ramos Sundukov

Director:

Albino Nogueiras Rodríguez

Degree:

Bachelor's degree in Audiovisual Systems Engineering

Examintaion session:

Spring, 2025

Summary

Text breu (entre 250 a 500 paraules) en què s'informa del contingut i la naturalesa del treball, on s'hi fan constar especialment els objectius, els mètodes, els resultats i les conclusions del treball. El resum ha de ser en català o castellà i anglès.

Índex

List of figures

List of tables

List of abbreviations

ESEIAAT Escola Superior d'Enginyeries Industrial, Aeroespacial i Audiovisual de Terrassa [UPC25].

RTA Real Time Analysis

SMAART System Measurement Acoustic Analysis Real-time Tool

FT Fourier Transform

FFT Fast Fourier Transform

DFT Discrete Fourier Transform

RFFT Real-valued Fast Fourier Transform

IEC International Electrotechnical Commision

RMS Root Mean Square

IIR Infinite Impulse Response

FIR Finite Impulse Response

Chapter 1

Introduction

1.1 Objecte

Resultat final que es vol aconseguir. En aquest cas, l'objecte d'aquesta plantilla és donar les pautes d'estructura i contingut de la Memòria del TFE.

El cos tant d'aquest document "Memòria" com dels altres documents integrants del TFE (Pressupost, Annexos i Plec de condicions) serà amb lletra Times New Roman o Arial d'una mida d'11 punts, marge lateral esquerra de 3 cm, dret de 2,5, superior i inferior de 2,5 i espaiat senzill.

L'alumne/a ha de revisar l'ortografia i gramàtica de tots els documents del TFE; ha d'utilitzar les unitats del Sistema Internacional; ha d'utilitzar un nombre coherent de decimals; i ha d'identificar els eixos dels gràfics inclosos al llarg del text.

Es recomana que la memòria no superi una extensió màxima de 60-70 pàgines. Tant les taules com figures han d'estar enumerades i tenir un títol. Si s'han obtingut d'algun altre document consultat, s'haurà de dir la font d'on s'ha tret [UPC25].

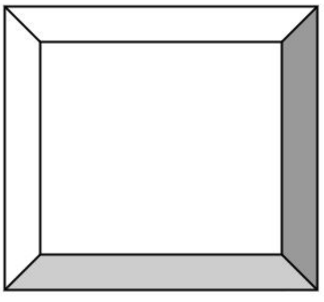


Figure 1.1: Imatge d'exemple

Table 1.1: Risks assessment

1	X2	X	X
...	...	...	...
...	...	...	...
...	...	...	...
...	...	...	...

1.2 Abast

Paquets de treball i lliurables necessaris per arribar a la solució.

1.3 Requeriments

O especificacions bàsiques. Restriccions sobre la solució final.

1.4 Justificació

Plantejament de la necessitat del treball des d'una visió global i aproximant-lo a una visió més específica. Serveix per centrar i contextualitzar el treball.

1.4.1 Subsection

1.4.1.1 superseccion

divisiones.py

```python
"""
Biblioteca con definiciones importantes para la división de números.
Se incluyen las funciones divideSiDivisible() y cocienteModulo().
"""

def divideSiDivisible(nume, deno):
"""
Si nume es divisible por deno, devuelve la división
entera. Si no lo es, devuelve None.
"""

if not nume % deno:
return nume // deno
```

```


def cocienteModulo(nume, deno):
"""
Devuelve el conciente entero y el resto de
la divisi
ón entera (mod) de dos números.
"""

return nume // deno, nume % deno

```

src/divisions.py

```
def divideSiDivisible(nume, deno):
        """
        Si nume es divisible por deno, devuelve la división
        entera. Si no lo es, devuelve None.
        """

        if not nume % deno:
        return nume // deno
```

Chapter 2

Background and/or status of the matter

Nowadays, there are many solutions that can fit to solve our problem. Some are very expensive, and others have shortcomings. In this chapter, we will have a look at some of the most popular solutions.

2.1 Smaart

Smaart, an acronym for *System Measurement Acoustic Analysis Real-time Tool* [**SMAART**], is a software-based solution commercialized by Rational Acoustics. It is probably the most used and well-known solution for professional acoustic analysis, used in big venues, concert halls, stadiums, touring productions, as well as in professional audio studios and speaker development laboratories. Common uses are:

- **Speaker Alignment:** When we have multiple sound sources, this software helps us find the phase and delay between them. For example, it can be used to find the time and phase alignment between a subwoofer and a full-range speaker.
- **RTA, Frequency and Phase Response** used to view live spectrograms, phase deviation, or energy in frequency bands. One example of use is identifying resonances at specific frequencies.
- **Coherence Analysis** to evaluate the quality of the measured data. A common use is to detect reflections and background noise.
- **Delay Time** between different sources or signals. Widely used to synchronize different elements of the system.
- **Room and architectural acoustics** to identify the frequency and phase response of a room, as well as reverberation and echoes.

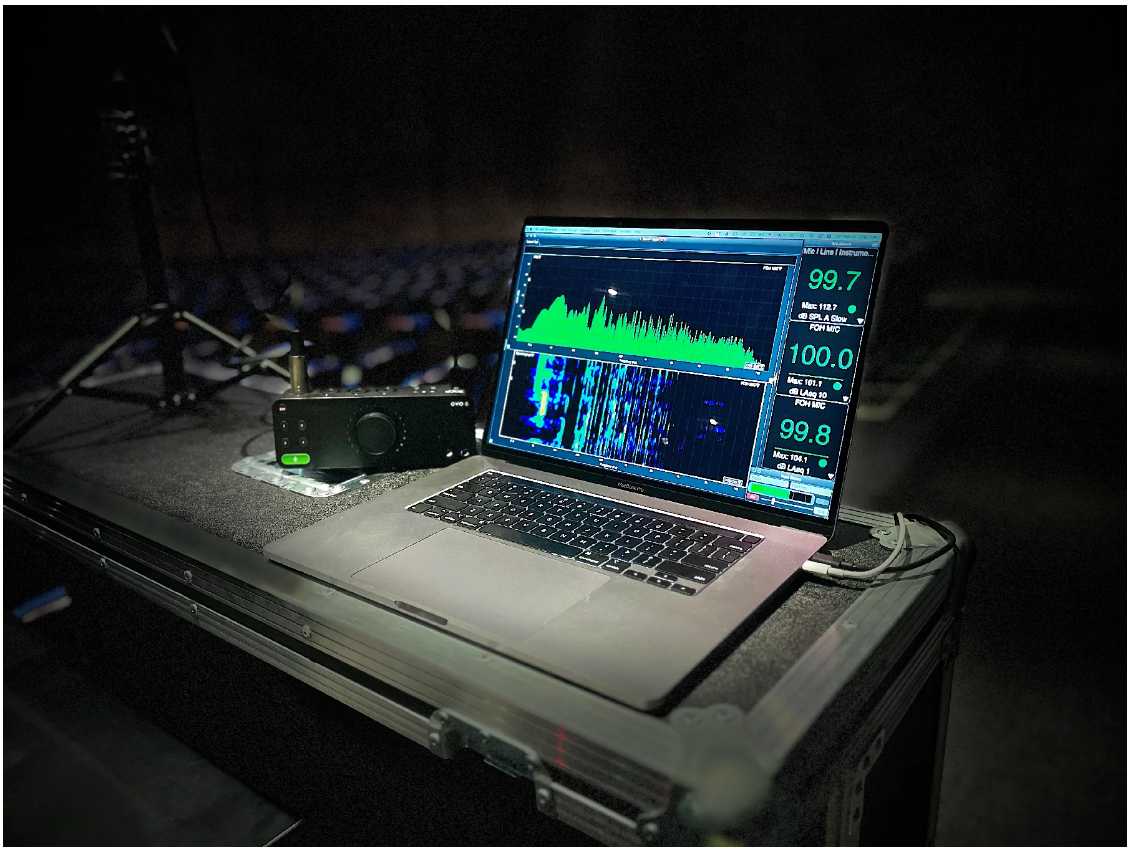


Figure 2.1: Notebook using Smaart, where we can see some of the tools it includes. The notebook is connected to the EVO 8 (a USB interface that acts as an external sound card).

The strongest points of this program are:

- **Flexibility:** As a software-based solution, it can run on any Windows or Mac computer (meeting the minimum required specs), and can be used with most external audio sound cards, allowing the connection of unlimited types of microphones or direct signals.
- **More than one channel:** This software can analyze and display information from more than one input channel at the same time, allowing comparisons between different channels. This is used to compare an original signal with the signal captured by a microphone inside a room with a sound system, helping to detect room acoustics or sound system issues. Another common use is to measure the sound in different places of the same room simultaneously.
- **Widely used:** It is very common to see professionals in the sector using this software, or at least being familiar with it. It has become a kind of standard, which leads other companies to ensure maximum compatibility with it. For example, Audix makes the Audix TM-1 Plus microphone [**AudixTM1**], which includes a file that can be imported into SMAART to apply microphone correction during analysis.

On the other hand, it requires a license, external hardware such as sound cards and microphones, and it does not have any correction capabilities—only analysis.

2.2 Dirac

Home correction solution

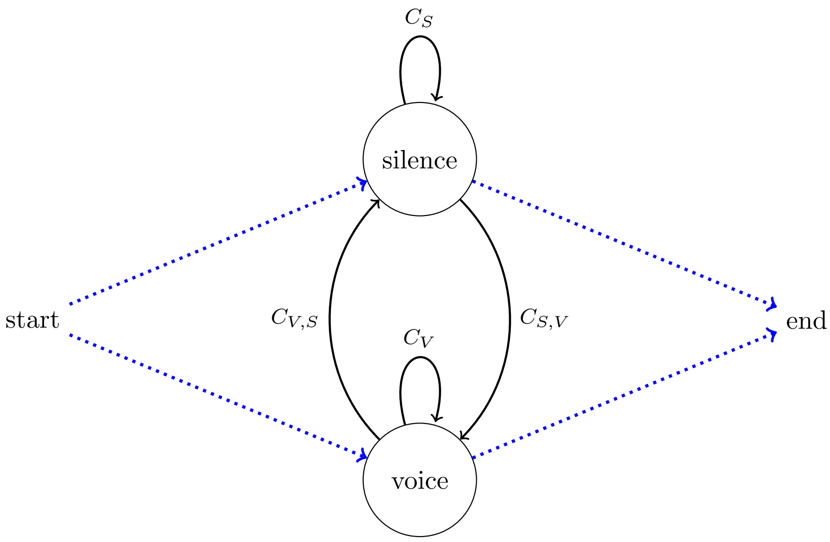
2.3 REW

Free software for room measurement

2.4 Trinnov

Hardware base solution with correction -

******************************** Exemple d'esquema *************************************



Chapter 3

Methodology, consideration and decision on alternative solutions

Passos que cal seguir per aconseguir els objectius del treball, incorporant les tècniques i eines necessàries per resoldre el problema.

Define how program should be.

Chapter 4

Development of the chosen solutions

El nom d'aquest apartat s'ha de triar en funció del treball que es porti a terme. Aquest apartat pot constar de diversos apartats i subapartats, depenent del criteri de l'autor o autora i de les consideracions de la tesi que es desenvolupa.

4.1 Graphic interface

About user friendly graphic interface.................................

4.2 Settings page

About parameters that can be set on this page, an why we need them.

4.3 Signal Path

How is adquired data from soundcard and how I move this thata inside the program

4.4 Acoustics analysis

The acoustic analysis is devoloped with python librari LIBROSA[McF+25]

4.4.1 Spectrogram (FT)

How implemented the spectogram

4.4.2 RTA

Usually, any form of analysis that is performed in real time can be considered **RTA** (*Real-Time Analysis*). This includes a wide range of operations such as spectrum monitoring, transfer function measurements, phase and coherence analysis, and more—all happening as the signal flows. However, in my experience, in common usage, when someone refers to ”RTA”, they are often specifically referring to the classic 31-band graphical spectrum display. Also, the data collected on this page is especially important because it will be used to set the correction parameters. For all of that, this page has been named **RTA**.

Also, there is another conflict. When someone defines the 31 bands and their bandwidth, it is common to use the definition provided in **IEC 61260**. However, this standard does not mathematically respect the logarithmic spacing between bands. The 31 bands are defined as 1/3 of an octave per band.

Table 4.1: Center frequencies for true 1/3 octave bands and IEC 61260 bands

1/3 octave	20	25.18	31.7	39.91	50.24	63.25	79.62	100.24	126.19	158.87	
IEC 61260	20	25	31.5	40	50	63	80	100	125	160	
1/3 octave	200	251.79	316.98	399.05	502.38	632.46	796.21	1002.37	1261.91	1588.66	
IEC 61260	200	250	315	400	500	630	800	1000	1250	1600	
1/3 octave	2000	2517.85	3169.79	3990.52	5023.77	6324.56	7962.14	10023.74	12619.15	15886.56	20000
IEC 61260	2000	2500	3150	4000	5000	6300	8000	10000	12500	16000	20000

```ipython
"""
In order to obtain the true 1/3 octave values, the following line of code was used with IPython3.
The results were rounded to the second decimal place.
"""

import numpy as np
print(np.round(np.logspace(np.log10(20), np.log10(20000), 31), 2))

```

On the other hand, this standard is widely used in many professional devices and software. One important example is the DBX 231s graphic equalizer, which is commonly used in analog processing chains.

Figure 4.1: Image of the front panel of the DBX 231s [**DBX˙31s**], where we can observe that the center frequency bands are the same as those defined in IEC 61260.

In order to achieve the greatest possible compatibility and coherence with industry standards, I prefer to use the IEC 61260 standard.

The signal path on this page is very similar to the one used on the "FT" page. We have a buffer with two blocks of input data ("In from external device" and "In from system").

First, we copy the buffer data to the "delay buffer", where, if needed, the data will be adjusted to make it coincide with the applied delay. If necessary, the adjustment will use data from previous blocks stored in the same "delay buffer".

I created this buffer with the capacity to store 1 second of data, which can be used to apply a maximum delay of (1 - "Block Size in seconds") seconds.

Once the data in the "delay buffer" is adjusted, we can start applying algorithms to perform the analysis.

The algorithm is based on the calculation of RMS (*root mean square*) to obtain the energy for each frequency band, which is divided using filters for each band.

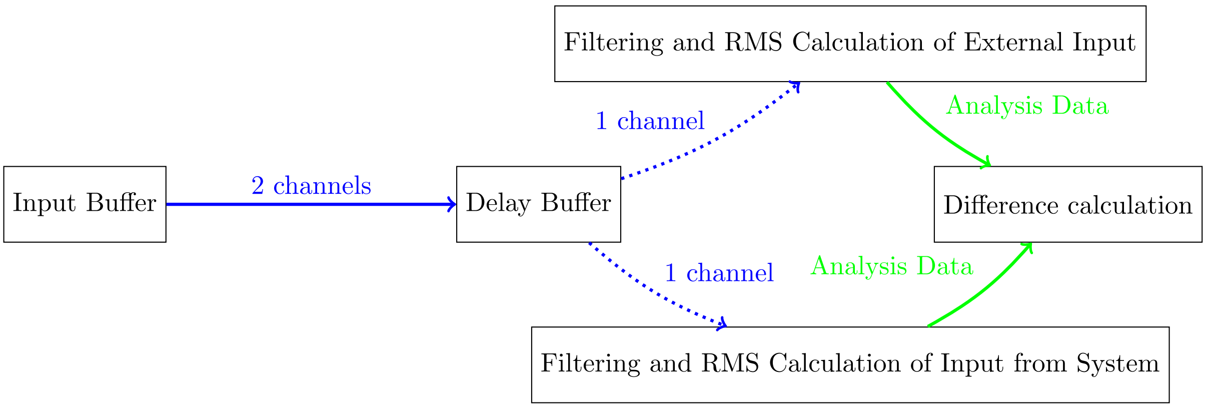


Figure 4.2: Diagram of the architecture of the RTA page

In this case, I'm not using any kind of windowing. As a starting point, I'm using 4th-order IIR (*infinite impulse response*) Butterworth band-pass filters for each band. All these filters are created using the `scipy.signal` library [**scipy˙signal**], which returns SOS (*Second-Order Section*) parameters.

Figure 4.3: Second-Order Sections for IIR filters with their parameters

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

When the program applies each filter to the signal block, it also calculates the RMS and converts it to a logarithmic scale, which will be plotted on the graph and used to calculate the difference graph.

Figure 4.4: Root Mean Square to calculate energy from filtered signal

$$RMS = \sqrt{\frac{1}{N}\sum_{n=0}^{N-1} x^2[n]}$$

At the end, there is: a pause button, which blocks the update function and can be used to pause the graphics; a save button, to store the current values of the difference graph for later use in the correction window; and a time averaging section that works exactly the same as the averaging section from the FT page, except that it does not include frequency averaging (since it doesn't make sense to apply frequency averaging between bands).

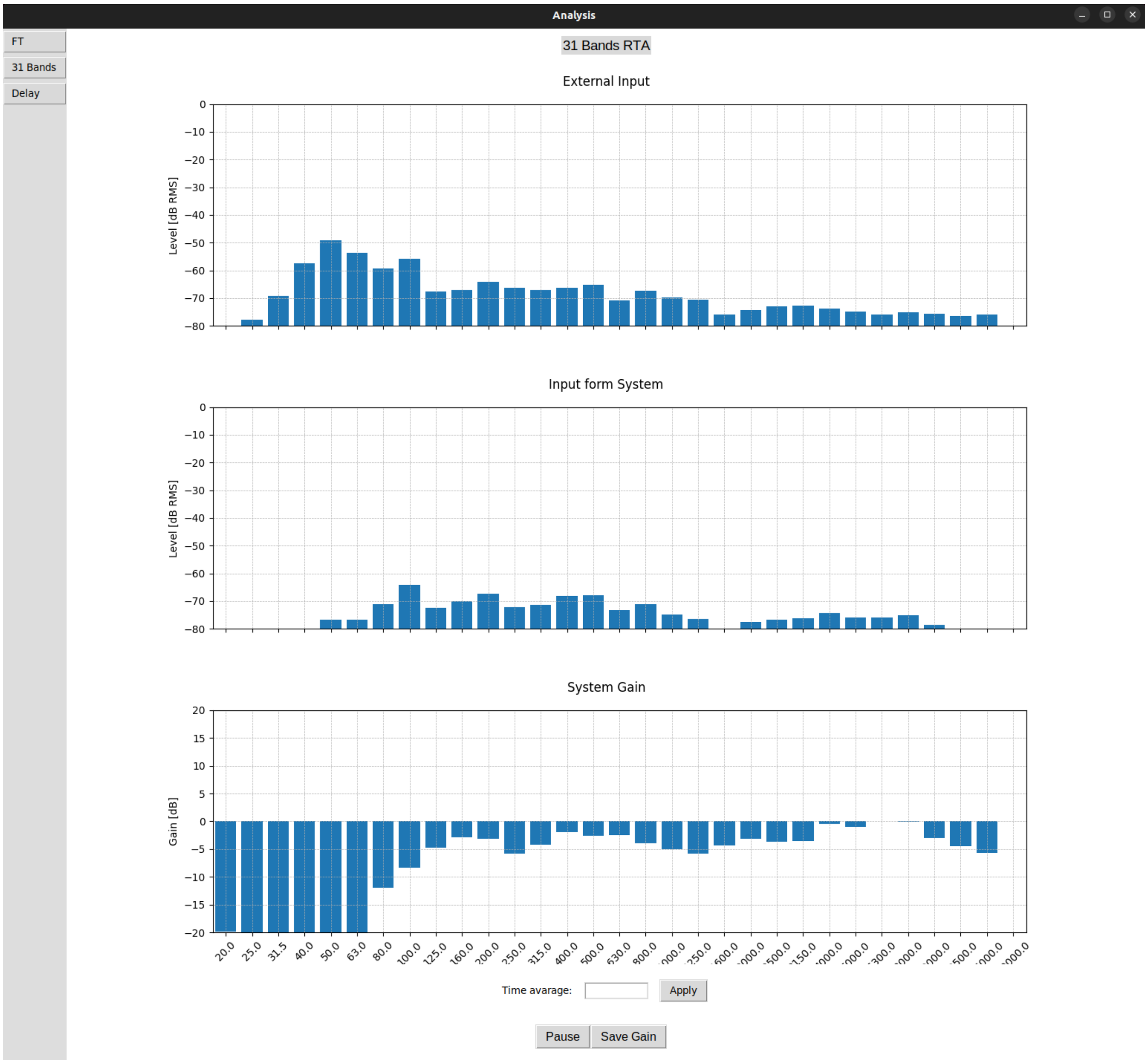


Figure 4.5: Analysis window - RTA page.

4.4.3 Delay

Explain delay page

4.5 Acoustic correction

Acoustic correction = DSP, allways have to be somethuing on the output buffer, by default, zeros.

4.5.1 Bypass

How Bypass works, and why it works bad.

4.5.2 31 Bars

Implementation of correction

4.6 Integration of monitoring mechanism

Home page and information that it appears / start, stop streams buttons...

4.7 Others

More problems that I didn't expect, losing time solving them or at least trying to. No more time to implement additional functionalities...

Chapter 5

Results

About the final program results

5.1 Final tests

About testing program on real situations

5.2 User experiance

About user experiance.

Chapter 6

Conclusions

Fa la funció de síntesi final i s'elabora a partir de la interpretació dels resultats assolits. La conclusió sol ser breu i s'ha de relacionar directament amb els objectius del treball.

També s'hi han d'incloure les recomanacions de continuació del treball i la planificació i programació del treball futur proposat.

Bibliography

[McF+25] Brian McFee et al. *librosa/librosa: 0.11.0rc1*. Version 0.11.0rc1. Feb. 2025. DOI: 10.5281/zenodo.14908061. URL: https://doi.org/10.5281/zenodo.14908061.

[UPC25] UPC. *The School of Industrial, Aerospace and Audiovisual Engineering of Terrassa*. Spring of 2025. URL: https://eseiaat.upc.edu/en?set_language=en.