Active Noise Cancellation using LMS algorithm : Theoretical Study and implementation on STM32 micro-controller

By Khalil Kaaniche supervised by Mr. Omar Bouzourraa

October 10,2022

QUEEN CASSIOPEIA

Contents

List of Figures

1 Introduction

Active noise cancellation is a technique based on adaptive filters used to minimise the unwanted noise by generating an inverted phase wave, resulting in a cleaner output. This method is used in many applications :

- High-End Headphones
- Reducing motor's noise
- Attenuating air conditionning noise
- Anti-snooring devices

This document will present briefly the active noise cancellation using LMS algorithm and will contain three main parts :

- Theoretical Study : This part will cover the theoretical notions necessary for the good implementation of the algorithm (FIR filters , Leaky LMS , Normalized LMS ....) and will contain the block diagrams used in our project.
- Simulation : It will contain all the results obtained on Matlab / Simulink in order to test the algorithm and to get familiarized with it.
- Practical Implementation : This section will present the mock-up used for a practical demonstration of active noise cancellation. It will also contain the results obtained during the conducted tests.

Unless otherwise stated, we'll use throughout the documentation the following symbols :

- $x(n)$: the input in the iteration n
- $e(n)$: the error signal to mminimize
- $d(n)$: the desired signal
- $y(n)$: the output of the FIR filter
- μ : Step size
- α : Leakage factor
- $\mathbf{x[n]}$: the row vector containing the input signal samples

$$\mathbf{x}[\mathbf{n}] = [x(n), x(n-1), x(n-2)....x(n-L+1)]^T \tag{1}$$

- $\mathbf{w[n]}$: the row vector containing the filter coefficients

$$\mathbf{w}[\mathbf{n}] = [w_0(n), w_1(n)....w_{L-1}(n)]^T \tag{2}$$

2 Theoretical Study

2.1 Adaptive filters

Adaptive filters are able to adjust their transfer functions depending on the adaptation algorithm according to the error signal in order to minimize the latter. They are built in two parts:

- A "FIR" (finite impulse response) filter
- An adaptation algorithm that will constantly adapt the coefficients of the FIR filter depending on the calculated or measured error

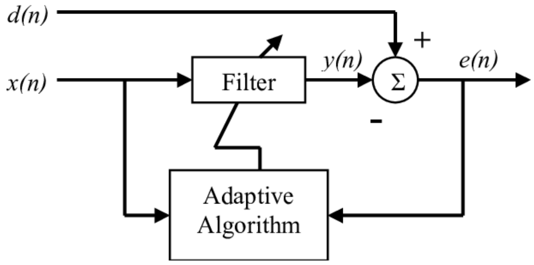


Figure 1: Adaptive filter block diagram

In the block diagram shown in Figure 1 and, as mentioned in the introduction, the signal *x(n)* is the input, *d(n)* the desired signal, *y(n)* is the output of the FIR filter and *e(n)* is the calculated error.

2.1.1 FIR filters

A finite impulse response (FIR) filter is a filter whose response to any finite length input is of finite duration, because it settles to zero in finite time since it doesn't have any feedback. The input signal is delayed by a number of time units equal to the filter coefficients, which we can clearly see in the figure 2 shown below.

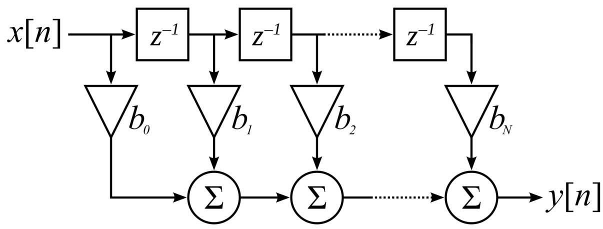


Figure 2: Direct form discrete-time FIR filter of order N

In this figure, b_0, b_1, ... , b_N are the N+1 coefficients of the N order FIR filter. The index n denotes the sample, the input $x[n]$ and the output $y[n]$ is given by the equation below:

$$y[n] = \sum_{m=0}^{N} x[n-m] \times b_m \tag{3}$$

Some advantages of the use of FIR filters in digital signal processing are :

- Simple implementation
- Stability
- Good performance

however, when using them we should keep in mind that they have some drawbacks :

- The need to a large amount of memory
- Impossible to simulate analog filters
- Introducing relatively important delays

2.1.2 Adaptation algorithms

As mentioned ,the adaptation algorithms take care of modifying the filter coefficients during each iteration so that the filter output *y(n)* approaches the desired signal *d(n)*. In other words, the error *e(n)* is minimized at each iteration. There are several algorithms but we will focus mainly on the LMS (least mean squares) algorithm. The " Least Mean Squares Algorithm " is an adaptation algorithm that seeks to minimize the root-mean-square error. We shall abstract from the mathematical background and the theoretical part related to this algorithm to keep only the essential for the implementation.

During our project we mainly implemented 3 variants of the cited algorithm :

Leaky LMS

$$\mathbf{w}[\mathbf{n+1}] = \alpha \mathbf{w}[\mathbf{n}] + \mu e(n)\mathbf{x}[\mathbf{n}] \tag{4}$$

Sign-sign LMS

$$\mathbf{w}[\mathbf{n+1}] = \alpha \mathbf{w}[\mathbf{n}] + \mu sign(e(n))\mathbf{sign}(\mathbf{x}(\mathbf{n})) \tag{5}$$

Normalized LMS

$$\mathbf{w}[\mathbf{n+1}] = \alpha \mathbf{w}[\mathbf{n}] + \frac{\bar{\mu}}{\epsilon + \sum_{i=0}^{L-1} x^2(n-i)} e(n)\mathbf{x}[\mathbf{n}] \tag{6}$$

2.2 Bloc Diagram

For the determination of the block diagram, it is necessary to understand the physical phenomenon of the active noise cancellation.

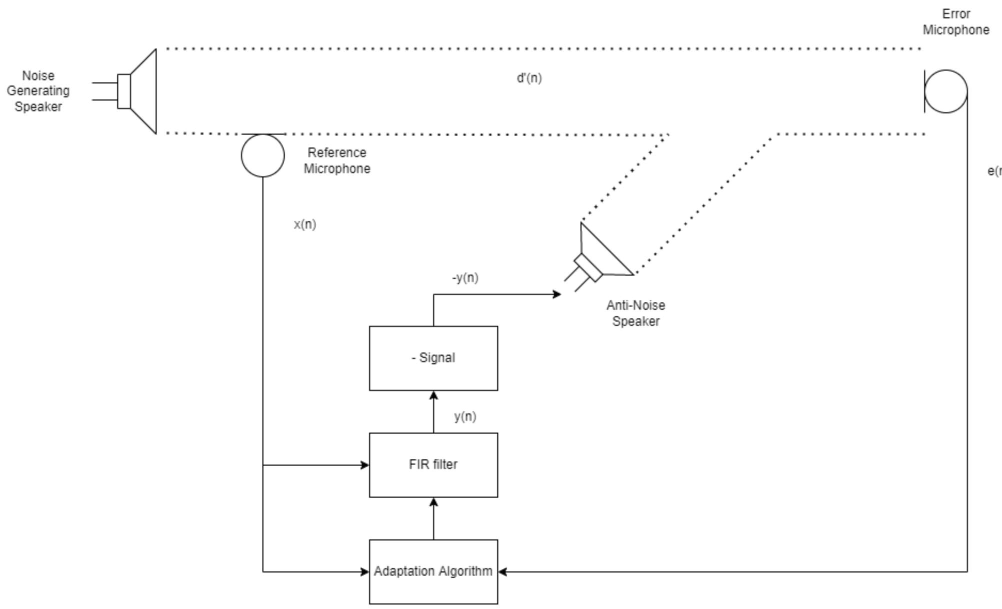


Figure 3: Simplified diagram of the model

First we will define some notions that we will use from this section :

- Noise speaker : speaker on the left of the diagram used to generate the noise.
- Anti-noise speaker : speaker on the right used to generate the anti-noise.
- Reference microphone : used to capture the generated noise.
- Error microphone : used to capture the remaining noise to be cancelled.
- Primary path : propagation path of the noise wave to the error microphone.
- Secondary path : propagation path of the anti-noise wave to the error microphone.
- Feedback path : propagation path of the anti-noise wave to the reference microphone.

A block diagram similar to the figure 3 won't be possible since there is always a delay in the process, and it's physically impossible to generate an exact opposite waveform of the noise. Moreover, the reference microphone is able to capture a part of the anti-noise signal which will corrupt our measure. Also, the theoretical signal y(n) coming out of the filter is not identical to the one arriving at the summation junction (introduction of a phase shift, DAC and amplification of the signal, characteristics of the speaker, the acoustic path of the mockup used etc...) so we need to compensate this difference. Which will lead us to the block diagram shown below.

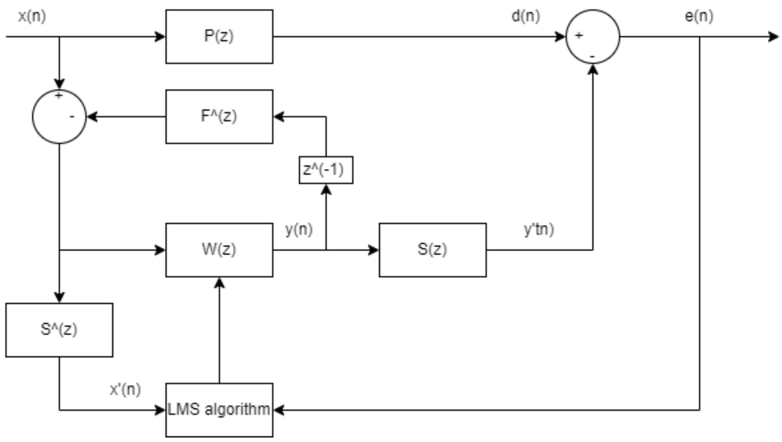


Figure 4: Active noise cancellation bloc diagram

with :

- x(n) reference signal
- y(n) output signal of the FIR filter
- d(n) input signal (the noise found at the summing junction)
- e(n) error signal captured by microphone at the summing junction
- P(z) transfer function modeling the primary path
- S(z) transfer function modeling the secondary path
- F(z) transfer function modeling the feedback path

- $\hat{S(z)}$ an estimation of S(z) made offline.
- $\hat{F(z)}$ an estimation of F(z) made offline.

3 Simulation

3.1 LMS algorithm implementation on Matlab

Before implementing the active noise cancellation on micro-controller and using the available hardware, we did some tests on Matlab in order to get familiar with the LMS algorithm and to do some primary tests. We can see through the figures below that the algorithm can be very efficient if the tuning operation is well done.

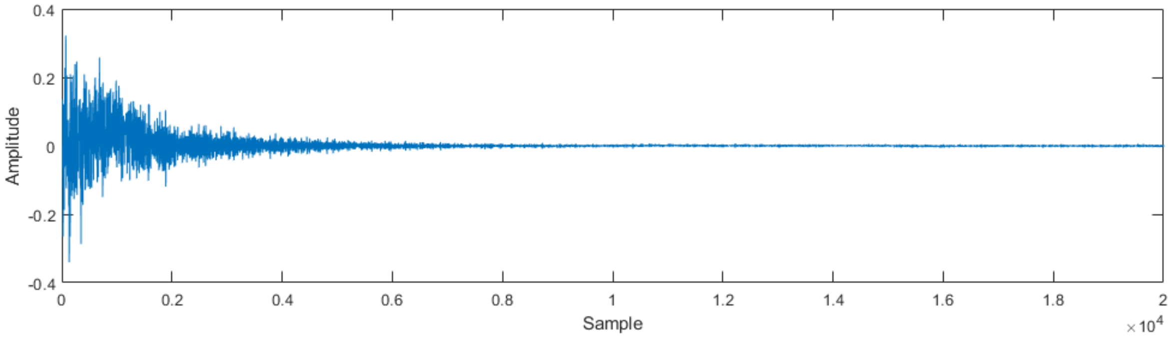


Figure 5: Error Plot

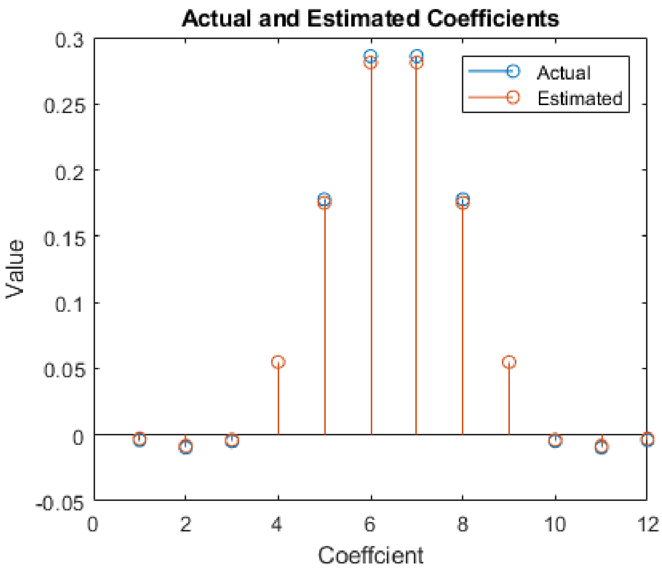


Figure 6: Estimated vs Actual coefficient of a FIR filter

3.2 LMS for transfer function estimation using simulink

In order to test some more complex block diagrams , we used Simulink to facilitate the implementation.

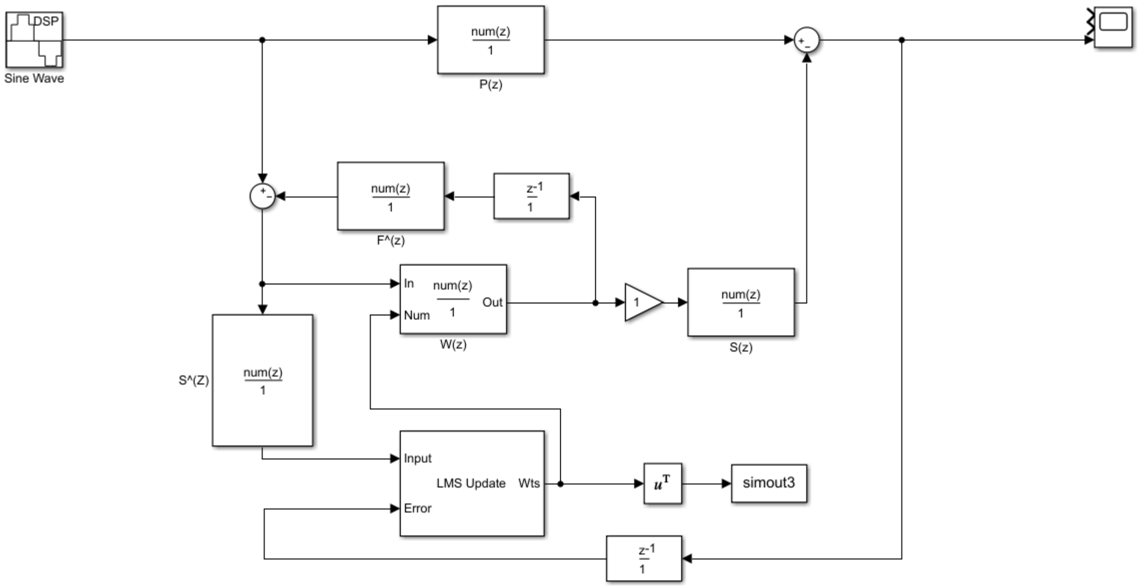


Figure 7: Simulink Block Diagram

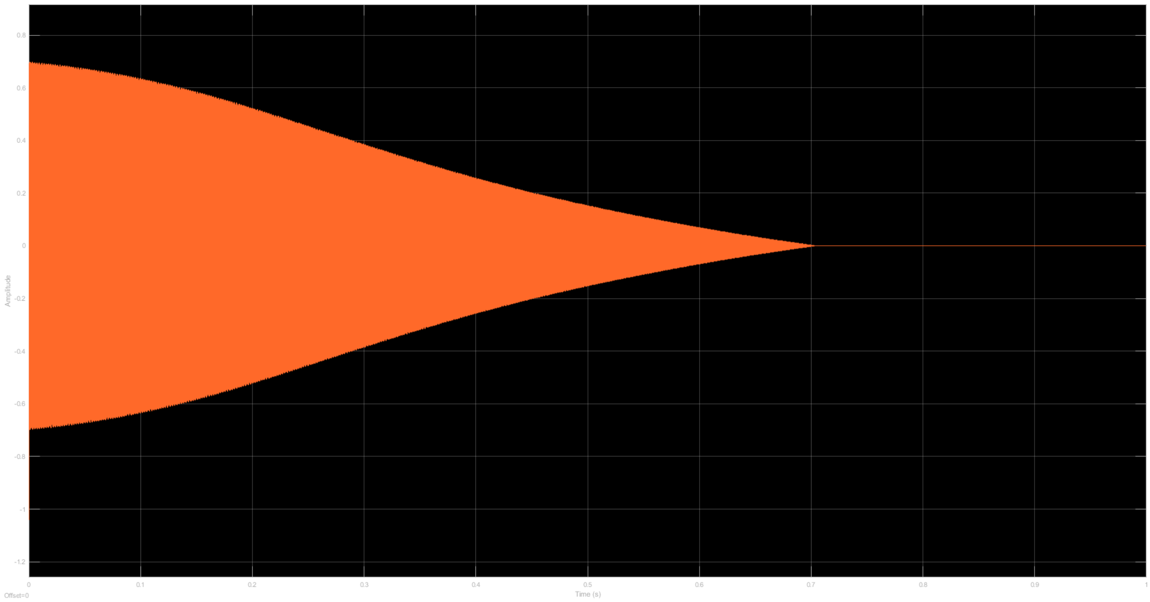


Figure 8: Error Plot

4 Practical Implementation

4.1 Hardware Setup

To get to the practical stage of active noise cancellation, we will need two microphones and two speakers. For a less noisy rendering, we opted to use an external amplifier based on a digital sound protocol I2S for each speaker to avoid using the DAC and a pre-amplification stage that could cause unwanted noise. Moreover, we opted for the use of a development board embedding an ultra powerful STM32H747 micro-controller in order to take full advantage of its power for this first version of the project.

4.1.1 Speakers SM400204-1

The speakers used have the following characteristics:

- Sound Pressure Level : 99 dB (@10 cm)
- Frequency Range : 220 - 20 000 Hz
- Nominal Power : 3 W
- Max. Power : 4 W
- Impedance : 4 Ω

4.1.2 Amplifier MAX98357

The MAX98357A is an I2S amplifier - it does not use analog inputs, it only supports digital audio inputs. This amplifier is designed to drive moving coil speakers only. The impedance of the speakers must be 4 Ω or higher (Which is our case). We will be using 5V from a regulator in order to power the used amplifiers. It has the following specs :

- Output power : 3.2W at 4Ω, 10 % THD, 1.8W at 8Ω, 10% THD, with 5V supply
- PSRR : 77 dB typ @1kHz
- I2S supported sample rates : 8kHz to 96kHz

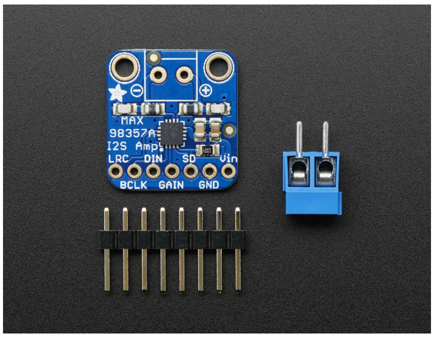


Figure 9: Picture of the MAX98357 Amplifier

Pinout : The figure 9 is showing that the amplifier contains 7 pins :

- LRC , BCLK and DIN : I2S pins (More details in the appendix at the end of this document)
- Vin and GND : used for the powering of the amplifier.
- Gain : used to change the static gain.

- **15dB** if a 100K resistor is connected between **GAIN** and **GND**
- **12dB** if **GAIN** is connected directly to **GND**
- **9dB** if **GAIN** is not connected to anything (this is the default)
- **6dB** if **GAIN** is connected directly to **Vin**
- **3dB** if a 100K resistor is connected between **GAIN** and **Vin**

Figure 10: Different Gain Configuration

- SD : used to switch the channel to output.

- If **SD** is connected to ground directly (voltage is under 0.16V) then the amp is **shut down**
- If the voltage on **SD** is between 0.16V and 0.77V then the output is (Left + Right)/2, that is the stereo average.
- If the voltage on **SD** is between 0.77V and 1.4V then the output is just the Right channel
- If the voltage on **SD** is higher than 1.4V then the output is the Left channel.

Figure 11: Different SD Configuratio

4.1.3 Microphone SPH0645

The used microphone has the following specs :

- Directivity : Omnidirectional.
- Sensitivity : -26 dBFS $\pm$3dB @ 94dB SPL .
- Signal To Noise Ratio : 65 dB(A)
- Frequency Range : 20 Hz - 10 kHz

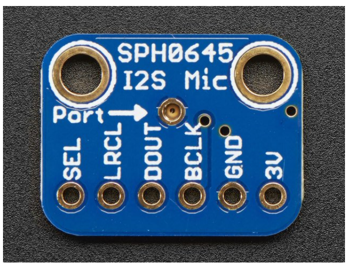


Figure 12: Picture of the SPH0645 Microphone

the figure 12 shows the following pins :

- 3V and GND : used for the powering.
- LRCL,DOUT,CLK : used for the I2S protocol.
- SEL : the channel selection pin. By default, this pin is low, so that it transmits on the mono left channel. If you connect it to a high logic voltage, the microphone will instantly start transmitting on the right channel.

4.2 Mockup

The figure below shows a 3D preview of the model made with SolidWorkS software. All parts have been designed to be cut out of 3mm MDF with a laser cutting machine in order to get a neat prototype . Also we covered the lower part of the mock-up with a foam to limit the reverb effect which had caused us a problem for a first version of the model made in PVC pipes.

Figure 13: 3D model of the mockup

The distance between the noise speaker and the debut of the secondary path is 392 mm.

4.3 Software Setup

A first part of the work was to enrich the digital signal processing library named EDL(Embedded DSP Library) created by the company Queen Cassiopeia SAS and to add the function containing the used variations of the LMS algorithm, as well as to carry out a test phase. In addition to that , we implemented a function for FIR filters since we need it in our projects. We also added some functions related to sound processing like compression and delay but they will not be developed in this document. Then, we created the three final projects that will be used for the last test phase. The first one is a project used to test the mock-up, the used components as well as the wiring and it follows the following flowchart.

flow.

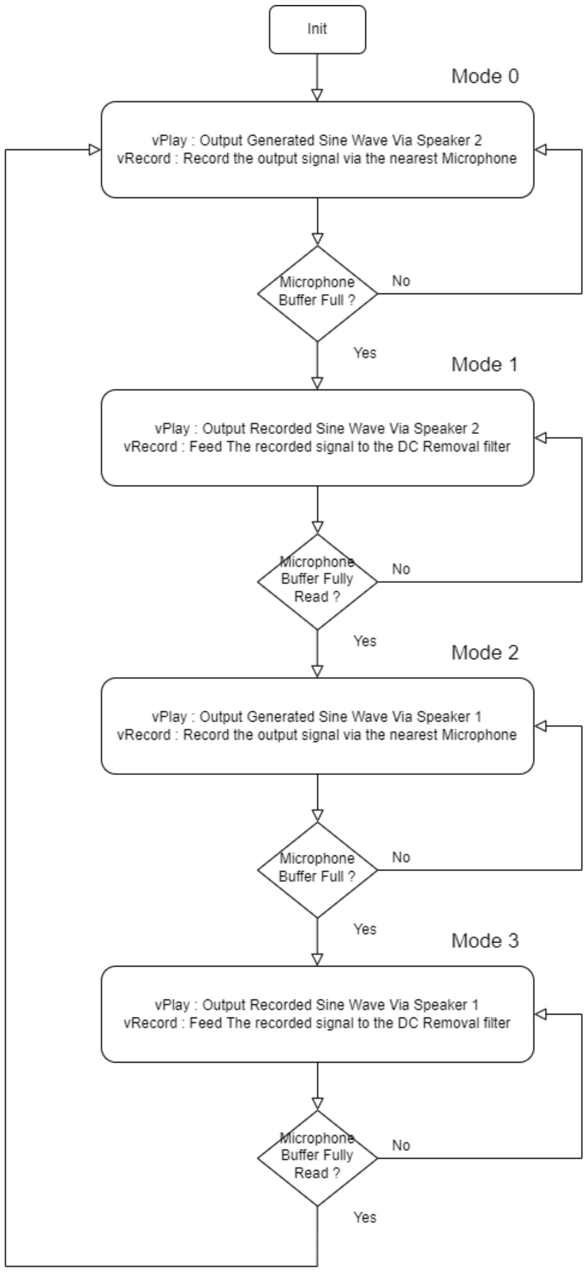


Figure 14: Project 1 Flowchart

As we have seen in the figure 4, in order to fully implement the ANC algorithm, we have to estimate the two functions S(z) and F(z). That was the aim of the second project.
In order to make the tuning easier, we decided to use the UART protocol in order to establish a communication with the PC so that we can send the values of the step size and the leakage factor while plotting the results of the algorithm. We use the same protocol to write the coefficients that we got after the process of tuning so they can be later used in the final project. the following figure shows the STM32 Cube Monitor flow.

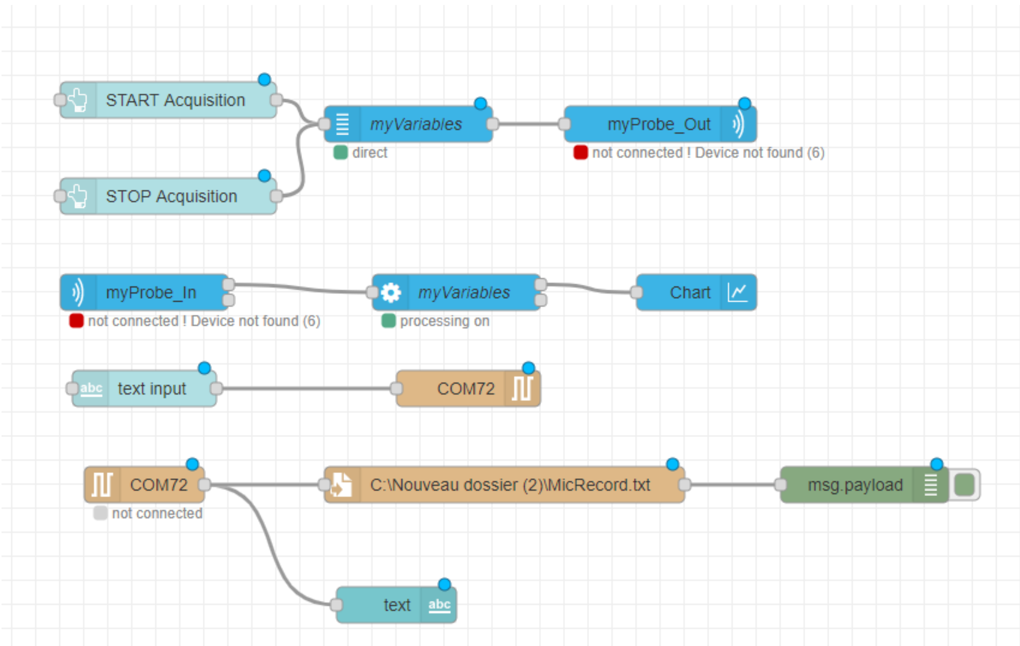


Figure 15: STM32 CubeMonitor Flow

and the flowchart of the second project is shown in the figure below :

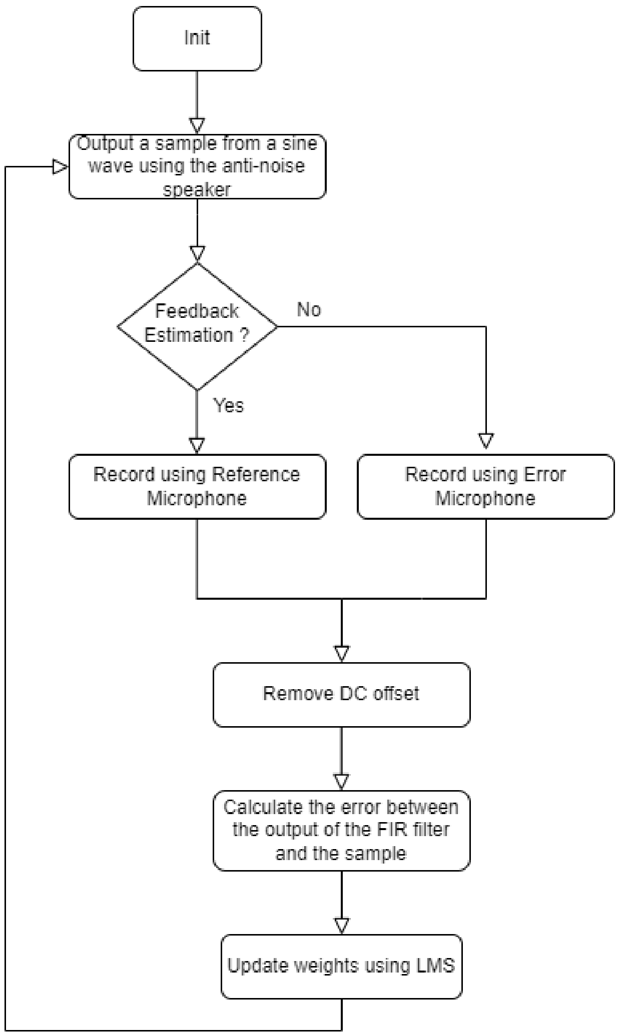


Figure 16: Project 2 Flowchart

After the estimation, we'll conclude with the implementation of the final block diagram with the help of the Embedded DSP Library. In the figure below , we'll find the design of our project's system. A part from the blocks used in the theoretical block diagram, we added two DC removals block (High pass filters) to eliminate the microphone's offset.

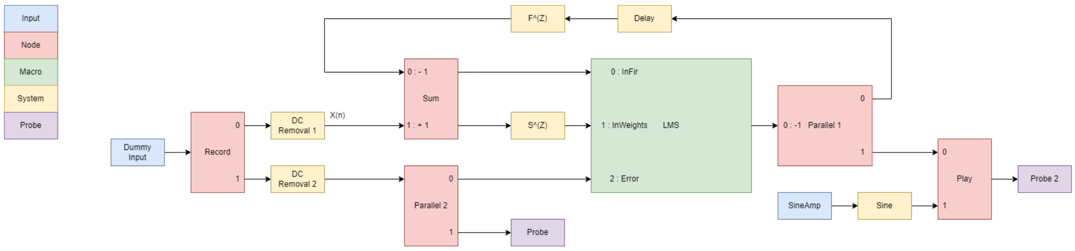


Figure 17: Project 3 System's design

4.4 Results

First of all , for the estimation part , we feed the speaker a sum of 6 sine waves with different frequencies and amplitudes. After different tests , we choose the normalized LMS algorithm since it's easier to tune.

4.4.1 Secondary and Feedback Path Estimation

We tried multiple configurations for the step size , and also we tried changing the order of the FIR filters for the estimation and we found out that these are the best parameters for each case. In several cases, increasing the FIR order will require additional computing power, despite that it won't reduce the error.

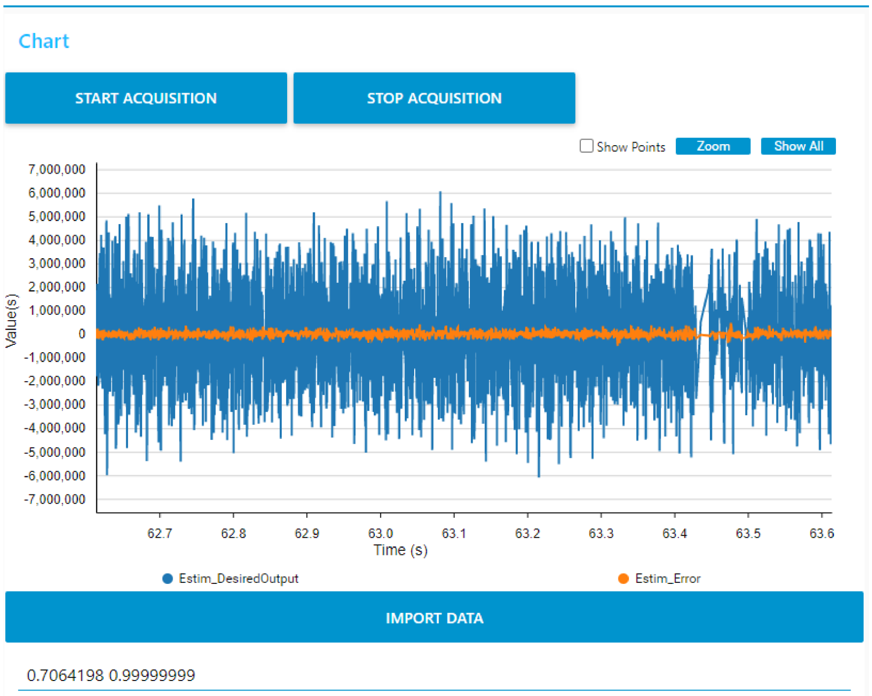


Figure 18: Plot of the Secondary Path estimation using Cube Monitor

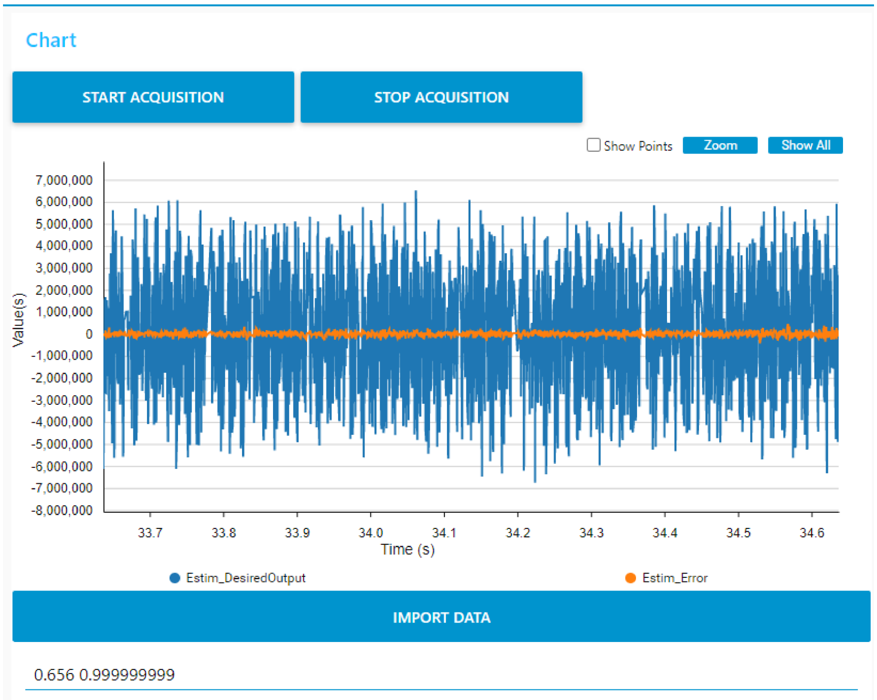


Figure 19: Plot of the Feedback Path estimation using Cube Monitor

In each figure, the signal represented in blue is the desired output and the one in orange is the calculated error. The box below the plot enables us to write the Step Size and the Leakage factor with a space between them so the data can be parsed on the micro-controller easily. As mentioned previously , the coefficients were sent to a different file for each path estimation in order to use them in the final algorithm.

4.4.2 Active Noise Cancellation

The figure below shows the final result of the algorithm after tweaking the different parameters.

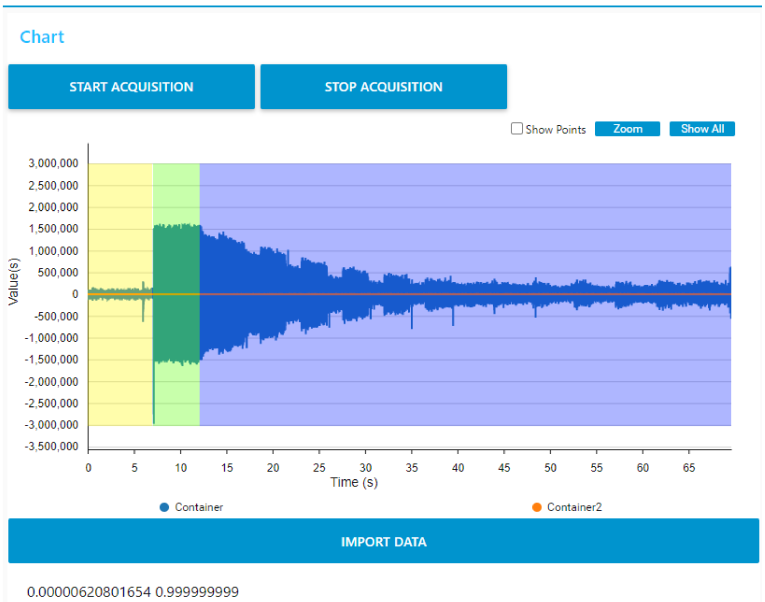


Figure 20: Plot of the Feedback Path estimation using Cube Monitor

The chart presents the error's microphone divided into 3 main parts highlighted :

- In yellow : when the noise speaker isn't working yet.
- In green : when the the noise speaker is working but the anti-noise is not.
- In blue : when the Active noise algorithm started

5 Conclusion

We can conclude after the different tests that the LMS algorithm is well and truly functional for active noise cancellation, however it presents a certain complication for the tuning part even if it's easier for the normalized LMS. The noise is not entirely suppressed, so for the time being, our project is not suitable for headphones or earphones, but can be used for noise suppression in ventilation ducts for example which can be a more error-tolerant application.
For future improvements, we can try to:

- Use better components as the microphone or the speakers used have some limitations (for example the frequency range).
- Test other variations of the LMS algorithm.
- Take more time for the tuning part.
- See if a more modern approach based on deep learning or machine learning can give better results.

6 Appendix

Hardware specs

Speakers

Sound Pressure Level (expressed in decibels) It is the sound pressure measured at one meter from the loudspeaker, when it is supplied with a power of one watt unless otherwise stated in the datasheet.The higher it is, the louder the sound emitted by the speaker.

Frequency Range It corresponds to the range of frequencies that the speaker can reproduce. It is present in the documents generally in the form of curve of response.

Nominal and Max Power (expressed in watts) It is the power that the loudspeaker can support in a punctual way for the max value and in a continuous way for the nominal.

Impedance (expresses in Ω) It is the resistance of the coil of the speaker added to the impedance due to the movements of the coil in a permanent magnetic field. It should be taken into account when connecting the speaker(s) to the amp.

Amplifiers

Output Power (expressed in Watts) It is defined by the average power supplied by the amplifier. It's relative to the impedance of the connected speaker like we've seen in our amplifiers datasheet.

Power Supply Rejection Ratio (expressed in dB) It characterizes the ability of the system to reduce the impact of the power supply change on its output (the sound produced in this case)

I2S Supported sample rates (expressed in Hz) It shows us the different sampling frequencies of the I2S protocol supported by the amplifier.

Microphones

Directivity it characterizes its sensitivity according to the origin of the sound, according to its central axis. The figure below shows different directivities of a microphone:

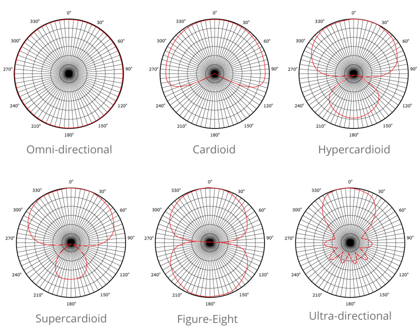


Figure 21: Picture of different microphone directivities

Frequency Range (expressed in Hz) It corresponds to the range of frequencies that the microphone can capture.

Sensitivity (expressed in dB) it indicates the performance of the microphone: a high sensitivity eliminates background noise and other extraneous sound, for a clear and precise rendering. To calculate the sensitivity of a microphone, we measure the ratio between the output voltage supplied (millivolts) and the sound pressure received (pascals). This gives the signal-to-noise ratio of mV/Pa. According to the manufacturers, the sensitivity can also be displayed in decibels (dB). The output voltage is then divided by a reference value of 1 V (the result is always negative). Thus, the lower the number of dB displayed, the higher the sensitivity:

- 1mV/Pa = -60 dB
- 10mV/Pa = -40 dB
- 100mV/Pa = -20 dB

Signal To Noise Ratio (expressed in dB) It is the ratio of the MEMS microphone's internal noise level to a standard reference pressure, in this case 94 dB SPL (1 Pa) at 1 kHz.

I2S Protocol

It is a standard connection protocol for digital audio devices. It allows controlling 1 or 2 audio channels of PWM audio signal. It gives the user the flexibility to choose the resolution and sampling rate.

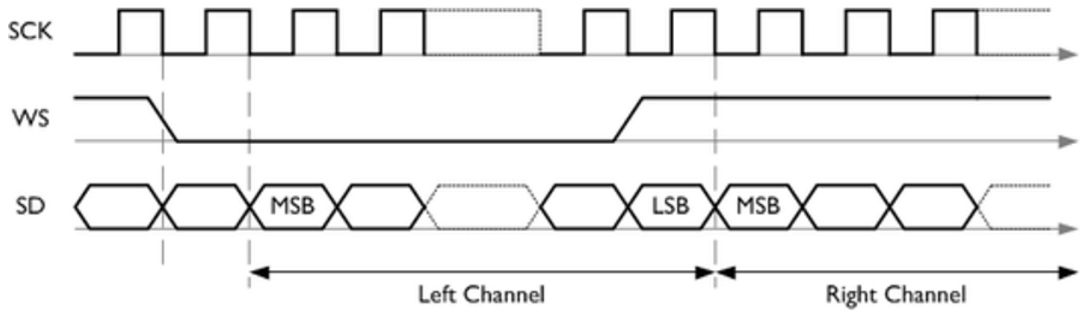


Figure 22: I2S communication protocol

It needs 3 lines apart from the GND:

- SCK (Serial Clock) : also called BCLK for bit clock. Its frequency is determined by :
$$Sampling frequency \times bitsperchannel \times number of channels. \tag{7}$$
For example, for a CD quality (44.1 kHz and 16 Bit):
$$44.1kHz \times 16 \times 2 \tag{8}$$
- WS (Word Select):also called LRCLK (Left Right Clock) or FS (Frame Sync) is a line alternating between high and low state to be able to alternate between channels.
- SD (Serial Data): it can be SDATA,SDIN,DOUT,DACDAT,ADCDAT and Contains the data sent or received according to the device with which we communicate.

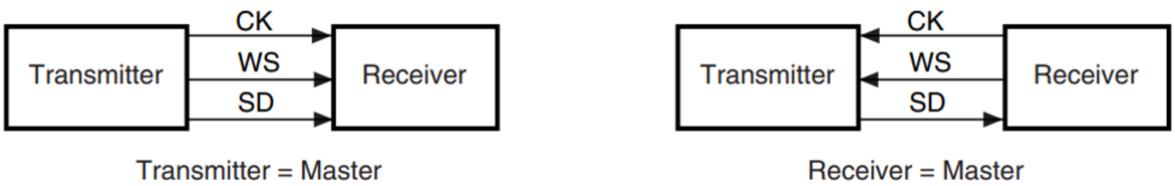
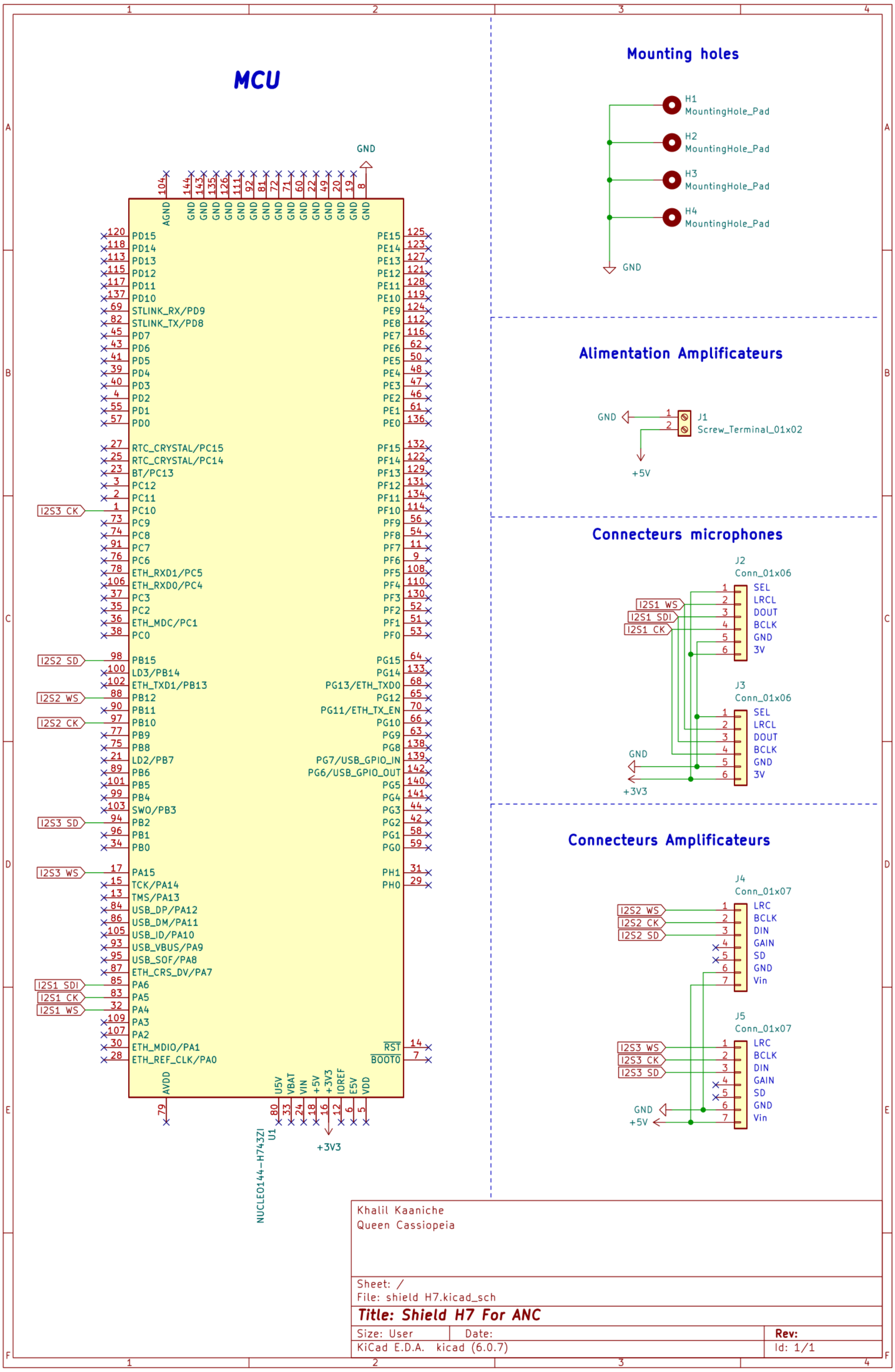


Figure 23: I2S Configurations

The diagram on the right illustrates the case where the receiver is the micro-controller and the transmitter is the microphone, while the one on the left illustrates the case where the transmitter is the micro-controller and the receiver is the speaker amp.

Electrical Schematic



7 References

- Active Noise Control with Simulink Real-Time - MATLAB and Simulink. (n.d.). Retrieved October 20, 2022, from $https : //www.mathworks.com/help/audio/ug/active-noise-control-with-simulink.html?s_eid = PSM_15028$
- Diniz, P. S. R. (2010, October 29). Adaptive Filtering: Algorithms and Practical Implementation (Softcover reprint of hardcover 3rd ed. 2008). Springer.
- Sidhu, S. S. (n.d.). Implementation of active noise cancellation in a duct [A Thesis Submitted In Partial Fulfillment of the Requirements for the Degree of Bachelor of Applied Science]. Simon Fraser University.
- DB UNLIMITED. (2018). SM400204-1 datasheet. $https : //dbunlimitedco.com/images/product$ $Drawings/SM400204 - 1.pdf$
- Maxim Integrated. (2019). MAX98357A/B datasheet. $https : //datasheets.maximintegrated.c$ $MAX98357B.pdf$
- Knowles. (2015). SPH0645LM4H-B datasheet. $https : //cdn-shop.adafruit.com/product-files/3421/i2S + Datasheet.PDF$
- STM32H745/755. (n.d.). STMicroelectronics. Retrieved October 20, 2022, from $https : //www.st.com/en/microcontrollers - microprocessors/stm32h745 - 755.html$