Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Master Thesis in ICT for Internet and Multimedia

Automotive Granular Synthesizer (AGS) for Electric Vehicles: Development and Affective Impact Analysis

Master Candidate

Giorgio Povegliano

Student ID 2089066

Supervisor

Prof. Canazza Targon Sergio

University of Padova

Co-supervisor

Dott. Alessandro Fiordelmondo

University of Udine

Co-supervisor

Dott. Anna Zuccante

University of Padova

Academic Year
2023/2024

To my family,
friends,
and Stella

Abstract

With the rapid expansion of electric mobility, more Electric Vehicles (EV) are on the roads, bringing new challenges, notably the "quiet vehicle problem." EV are significantly quieter than traditional Internal Combustion Engine Vehicles (ICEV), especially at low speeds, raising safety concerns for pedestrians and other road users who may not hear them approaching. This thesis, developed in collaboration with FIAMM Energy Technology and the CSC at the University of Padua, addresses this challenge by creating a Simulink model to generate customizable artificial engine sounds in real-time.

The project employs granular synthesis, a technique not widely used in the automotive field. This method led to the development of the Automotive Granular Synthesizer (AGS), which allows users to input audio files and adjust parameters to create specific sounds that enhance pedestrian safety while aligning with vehicle brand identity.

The thesis begins with an examination of the quiet vehicle problem and introduces the AGS as a solution. It explores the algorithm's principles, details the tunable parameters within the Graphical User Interface (GUI), and outlines the model's workflow. Finally, it presents an experimental study focusing on the emotional impact and detectability of the sounds generated by the AGS.

Contents

List of Figures

List of Acronyms

EV Electric Vehicles

HEV Hybrid Electric Vehicles

ICEV Internal Combustion Engine Vehicles

AGS Automotive Granular Synthesizer

GUI Graphical User Interface

CSC Centro di Sonologia Computazionale

FIAMM Fabbrica Italiana Accumulatori Motocarri Montecchio

IEA International Energy Agency

ADAS Advanced Driver-Assistance Systems

SPL Sound Pressure Level

AVAS Acoustic Vehicle Alerting System

GEVO-2023 2023 Global EV Outlook

AsGS Asynchronous Granular Synthesis

RPM Revolutions Per Minute

ECU Electronic Control Units

CAN Controller Area Network

Tmin Minimum Grain Size

Tmax Maximum Grain Size

LPF Low Pass Filter

HPF High Pass Filter

BPF Band Pass Filter

NVH Noise, Vibration and Harshness

ANOVA Analysis of Variance

HSD Tukey's Honestly Significant Difference

1 Introduction

This project originates from a collaboration between the Centro di Sonologia Computazionale (CSC) at the University of Padova and Fabbrica Italiana Accumulatori Motocarri Montecchio (FIAMM), focusing on developing an innovative solution for acoustic warning systems in Electric Vehicles (EV).

Figure 1.1: CSC logo

Figure 1.2: FIAMM logo

1.1 Problem Statement and Opportunities

While the quietness of EV reduces urban noise pollution, it also poses a significant safety challenge for pedestrians and other road users, particularly at low speeds, where the absence of traditional engine noise makes these vehicles harder to detect. In recent years, the rapid adoption of EV and the implementation of new safety regulations (see Chapter 2) have prompted automakers to invest heavily in the development of advanced acoustic warning systems designed specifically for electric and hybrid vehicles. Recognizing the unique challenges posed by the quieter operation of these vehicles, greater emphasis has been placed on creating innovative sound solutions that not only enhance pedestrian safety but also improve the overall driving experience.

As stated in Section 2.3.1, wavetable synthesis is one of the most commonly used technique for addressing the problem and generating vehicle sounds. However, this conventional approach offers opportunities for exploring alternative synthesis methods, each bringing unique benefits that could surpass traditional techniques. One such alternative is granular synthesis, which, although primarily explored within academic contexts for EV sound design, presents considerable potential for creating more dynamic and expressive soundscapes. The primary challenges of granular synthesis include its inherent complexity and the stochastic nature of the approach, which can make it difficult to ensure consistency and reproducibility in sound generation. Despite these challenges, the potential advantages of granular synthesis, such as the ability to create more varied and engaging sound environments, make it a promising area for further research and development in the field of electric vehicle acoustics. Given the necessity for more dynamic and expressive soundscapes in EV, this project seeks to address these challenges by exploring the potential of sound granulation.

1.2 Project Goals and Outcomes

To meet these needs, a granular synthesis approach was developed using Simulink, a graphical programming environment based on MATLAB. Simulink was selected for its ability to provide visual feedback and facilitate the analysis of signal outputs from the synthesis algorithm. This visual representation allows engineers to better understand and refine the complex processes involved in granular synthesis, making it an ideal tool for developing and fine-tuning the acoustic characteristics needed for electric vehicle sound design. Additionally, Simulink enables the direct generation of C code from its graphical blocks, which can then be compiled and implemented on the microcontrollers responsible for managing sound actuation in the vehicle.

The result of this project is an application designed for engineers to develop and refine sounds for electric and hybrid vehicles using granular synthesis. This application features a user-friendly interface that facilitates the adjustment and tuning of various synthesis parameters. Furthermore, it was crucial to assess the quality and detectability of the sounds generated by the granular synthesis algorithm. To achieve this, an impact analysis was conducted through an online survey. The detectability is discussed extensively through an experiment inside the thesis of my colleague Edoardo Di Pietrantonio [36], with whom I collaborated on this project.

This thesis is structured as follows: Chapter 2 presents the state of the art in sound design for electric vehicles, providing insights into current trends and technologies. Chapter 3 offers an overview of the granular synthesis algorithm, detailing its principles and functionalities. Chapter 4 focuses on the application itself, outlining the development process and features of the interface. In Chapter 5, the principles of sound design and psychoacoustics are explored, emphasizing their importance in creating effective auditory experiences. Chapter 6 presents the findings from the online survey, analyzing the affective impact of the generated sounds on users. Finally, Chapter 7 concludes the thesis by summarizing the project's objectives, outcomes, and key findings, providing a reflection on the significance of the research.

2

State of the Art

2.1 The Evolution and Adoption of Electric Vehicles

In the constantly evolving automotive landscape, the past 20 years have witnessed a significant shift towards the development and adoption of EV [8, 9, 18]. Globally, EV sales are now growing 3 to 8 times faster than those of combustion engine vehicles [27]. In 2023, EV sales increased by nearly 15 million units compared to 2022, reaching a total of 40 million units. In contrast, EV sales in 2021 were less than half of the 2023 figure [24](see Figure 2.1).

This transition is driven by several factors, including environmental concerns, advancements in battery technology, and the rising cost of oil. Stricter emissions regulations and growing consumer awareness of sustainability have further accelerated the shift towards electric mobility [48]. Projections from the 2023 Global EV Outlook (GEVO-2023) report have already been met [23], and by the end of 2035, electric cars are expected to account for nearly half of global car sales, surpassing 60 million units worldwide [24] (see Figure 2.2).

2.1.1 Investment and Innovation

In response to this shift, companies across the automotive sector are investing heavily in the future, focusing on the development of increasingly efficient, safer, and more technologically advanced vehicles. This wave of investment

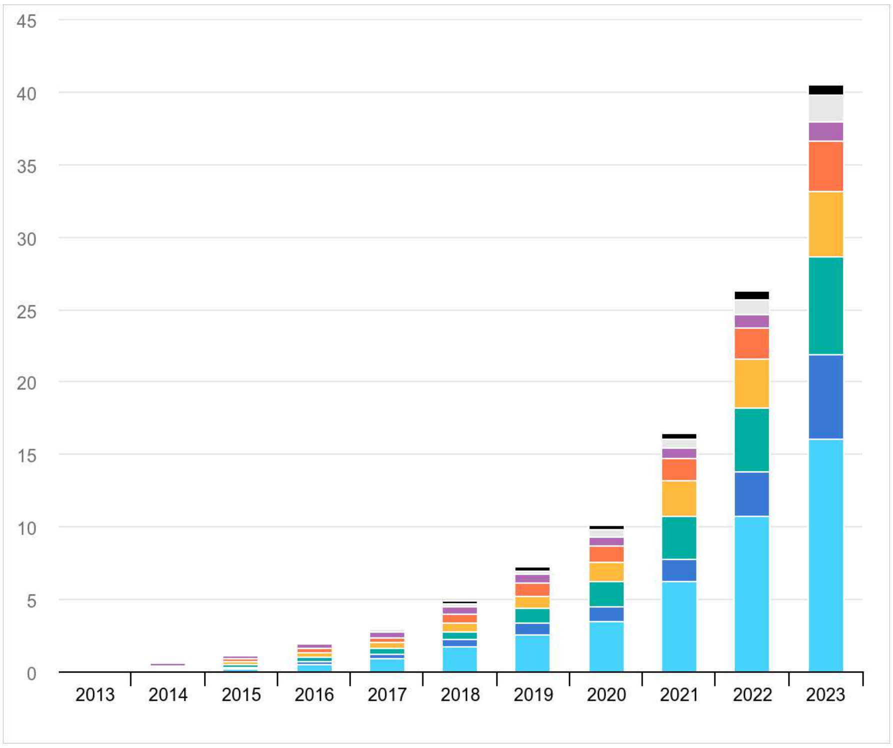

Figure 2.1: IEA trend in global EV sales over recent years. Updated with data as of April 30, 2024 [26].

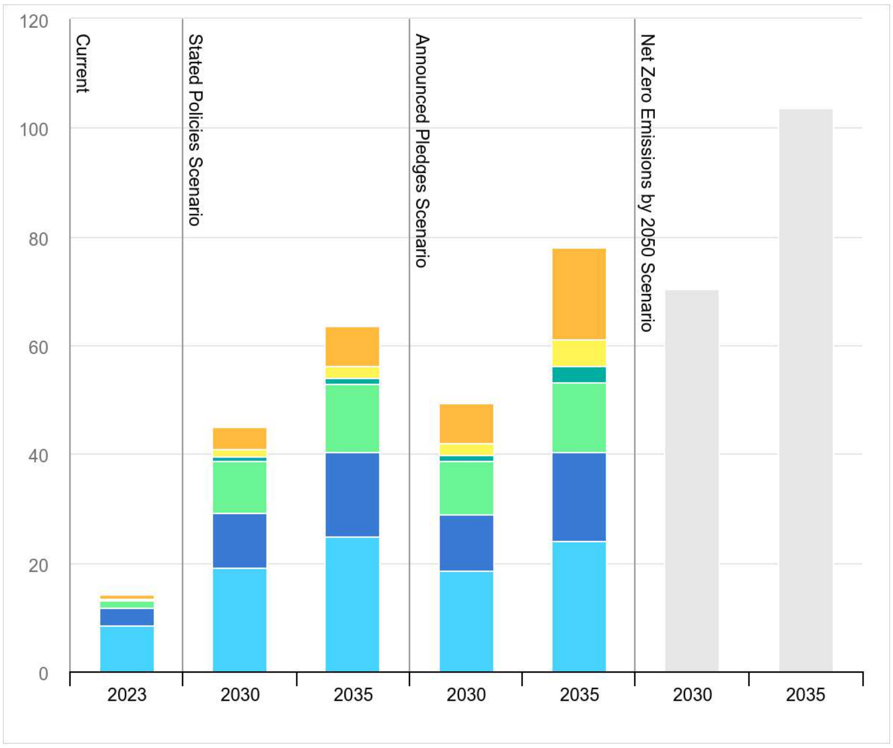

Figure 2.2: IEA projection in global EV sales under different scenarios. Updated with data as of April 30, 2024 [25].

spans the entire automotive ecosystem, from established automakers to new entrants, and from battery manufacturers to software developers working on smart vehicle technologies. Competition is driving rapid innovation, transforming the industry and leading to a more sustainable, electrified future. For instance, Volkswagen, GM, and Ford have announced multi-billion-dollar investments in electric mobility, with plans to electrify significant portions of their model lineups [10, 6]. Volkswagen, in particular, plans to offer 70 new electric models by 2028 and aims for 40% of its sales to be electric by the end of 2030 [6].

Infrastructure improvements are also playing a crucial role in accelerating the adoption of EV. The global number of charging stations is increasing rapidly, with investments from both governments and private companies expanding the network to make charging as convenient as refueling at gas stations. By 2030, the IEA expects there to be around 145 million EV on the road globally, up from about 11 million in 2020 [22].

2.1.2 Academic and Technological Advancements

Parallel to these technical advancements, there has been a growing interest in the academic field. A significant amount of research has been conducted to explore various aspects of electric vehicles, from improvements in battery technology and energy efficiency to the socio-economic impacts of widespread EV adoption. Researchers are also investigating the environmental benefits of EV, seeking ways to further reduce their carbon footprint and enhance sustainability [35]. Additionally, studies are examining the integration of EV with renewable energy sources, smart grid technology [18], and the development of autonomous driving capabilities [60]. As technology advances, modern vehicles are increasingly equipped with features that prioritize the safety of both passengers and pedestrians [9]. These innovations have greatly enhanced road safety, incorporating Advanced Driver-Assistance Systems (ADAS), automated emergency braking, and pedestrian detection technologies, among others. However, as we move toward a future dominated by electric vehicles, significant challenges remain that must be addressed to ensure their safe integration into our roadways.

2.2 The Quiet Vehicle Problem And Regulations

Despite all the aforementioned advantages, EV still have some downsides. In this study, we will focus on one key issue: while the quiet operation of EV contributes to a reduction in noise pollution, it also poses new challenges in terms of vehicle safety and driver engagement.

2.2.1 Challenges of EV Quietness

At low speeds, Hybrid Electric Vehicles (HEV) and EV are quieter compared to ICEV. Research indicates that the Sound Pressure Level (SPL) of an EV or an HEV in EV mode can be up to 20 dB(A) lower than that of a comparable ICEV when idling [13]. This difference is substantial, as the human ear can discern an SPL difference as small as 3 dB(A) [12]. However, as vehicle speed increases, the SPL differences between HEV/EV and ICEV diminish, becoming insignificant (less than 3 dB(A)) at speeds above 20 km/h [13]. In this context, the relative quietness of electric vehicles raises concerns regarding the safety of the streets, particularly for visually impaired people who solely rely on the sound emitted by the engine to identify an incoming car. This is identified in the literature as the quiet vehicle problem, and it also affects pedestrians and cyclists who cannot identify cars that are maneuvering at low speeds (10-20 km/h). In fact, this is supported by various studies indicating that EV are much more likely to collide with cyclists or pedestrians due to their low-noise engines [31, 16].

2.2.2 Balancing Safety and Noise Pollution

To address this problem, the EU has issued Commission Delegated Regulation (EU) 2017/1576, which mandates that all new types of electric and hybrid cars be fitted with a new safety device as of 1 July 2019, the Acoustic Vehicle Alerting System (AVAS). The device will automatically generate a sound from the start of the car up to approximately 20 km/h and during reversing. The sound-emitting device will be obligatory in all new electric cars as of 1 July 2021 [40]. Regulations on EV sounds have been in place in Japan since January 2010, while in the USA, regulations have been in effect since December 2010. However, the U.S. National Highway Traffic Safety Administration issued its final ruling in February 2018 [58]. The challenge is to ensure safety for the more vulnerable street users while still reducing noise pollution in already polluted cities. To address this, EVs and HEVs must produce a continuous noise level of at least 56 dBA (within 2 meters) when the car is traveling at 20 km/h (12 mph) or slower, with a maximum noise level of 75 dB(A) [58].

2.2.3 Sound Design for Brand Enhancement

The Quiet Vehicle Problem, while still a key safety issue, also presents untapped business opportunities to reinforce vehicle branding. Automakers can design the sound emitted by EV to reflect their brand identity, much like has been done in the past with combustion engines. Consider sports cars like Ferrari or Lamborghini, as well as Jaguar and Aston Martin, which deeply rely on sound as a core element of their identity on the streets. Given these observations, the sounds of EV and HEV vehicles need to be evaluated not only from the perspectives of safety (detectability) and soundscape (reducing noise pollution) but also from a sound design standpoint. The perceived sound quality is a key criterion that should be exploited to reinforce brand identity, enhancing both the driver's experience and the subjective impression of pedestrians, who could ultimately be potential customers.

2.3 Context and Challenges in EV Sound Design

Due to the new regulations on sound emissions for EV, companies are facing the challenge of finding the most effective solutions to address this issue. The sound generated by EV must meet several criteria: it needs to be both detectable and recognizable, striking a balance between being neither too loud nor too soft, while also remaining pleasant for both drivers and pedestrians.

It is important to note that the rapid adoption of electric vehicles is a relatively new phenomenon. Although there are now over 40 million electric cars in use globally, up from 26 million in 2022 [41], this number still represents only a small fraction of the total 1.47 billion vehicles on the road [11]. Consequently, there is a lack of established reference points for the average consumer. Customer expectations for the sound produced by EV are not yet fixed or well-defined, and the frame of reference for evaluating these vehicles is continuously evolving [14]. As the EV market matures and more people become familiar with these vehicles, the criteria and expectations for what constitutes a desirable EV will likely shift.

2.3.1 Wavetable Synthesis in EV Sound Design

Currently, one of the most widely used technique in the field of automotive sound is wavetable synthesis. This method involves generating sound by using pre-recorded waveforms, or "wavetables," which can be manipulated in real-time to produce a range of auditory effects. Wavetable synthesis allows for the creation of complex, dynamic sounds that can be tailored to meet specific regulatory requirements and brand identities. The principle underlying this technique is far from complex, and is represented in Figure 2.3.

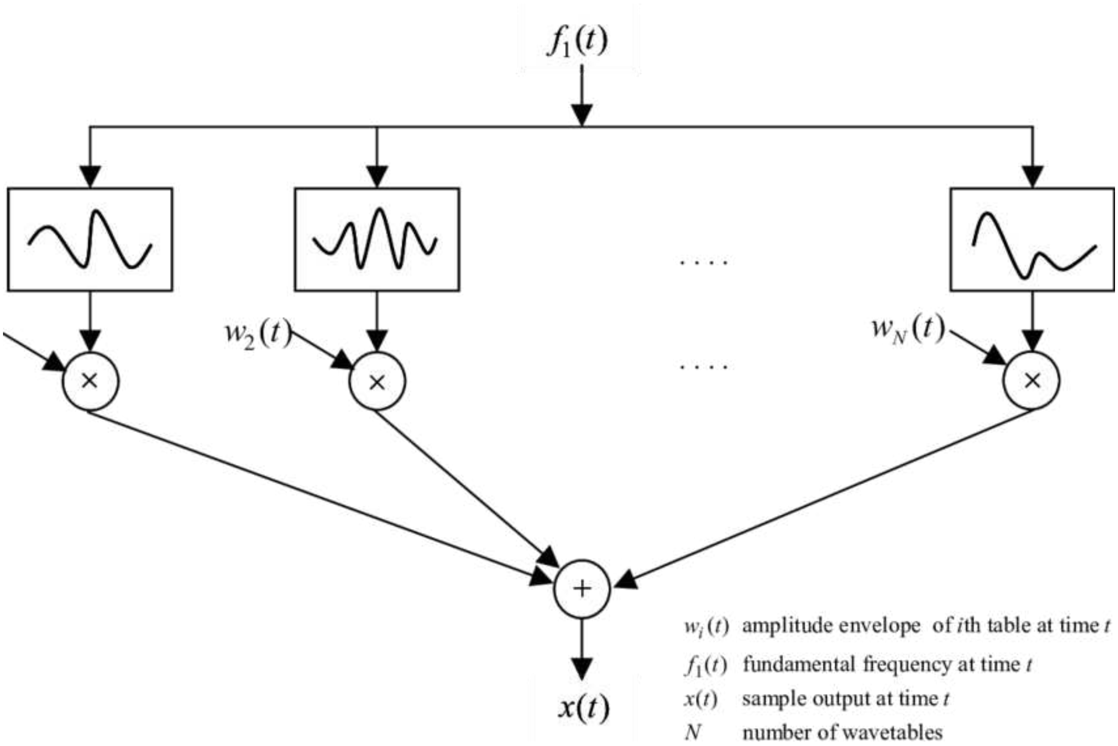

Figure 2.3: Schematic representation of the wavetable synthesis technique

Wavetable synthesis is particularly popular in the automotive industry because it offers flexibility, precision, and reproducibility in sound design. Typically, the algorithm uses various car-related parameters to alter the final sound. A notable feature of wavetable synthesis is the ability to adjust the starting point of the waveforms, which changes the timbre of the sound based on factors such as the car's speed. Additionally, the throttle can be mapped to affect the pitch of the wavetables, or its envelope, among other adjustments. The range of possible combinations is very large, allowing for the creation of sounds that can be tailored to align with a brand's identity.

2.3.2 Enhancing Sound Realism

In addition to wavetable synthesis, which forms the foundation of the sound design, an increasing number of companies are incorporating additional auditory details to enhance the realism and emotional appeal of these sounds. For instance, some manufacturers are adding simulated pops and bangs that resembles a traditional cars exhaust system, which evoke the experience of internal combustion engines [37, 21, 4]. These sounds are not just cosmetic; they serve to bridge the gap between the familiar auditory cues of conventional vehicles and the new, quieter world of electric propulsion.

Examples of such enhancements include artificial engine revs that respond dynamically to acceleration, providing drivers with an intuitive sense of speed and power [21, 37]. Another example is the integration of turbocharger whines or gear shift sounds, which can make the driving experience more engaging by replicating the auditory feedback that enthusiasts have come to expect from high-performance vehicles [37]. Companies like BMW and Porsche have pioneered the inclusion of these elements in their electric models, allowing drivers to choose from different sound profiles that can range from futuristic hums to more traditional, aggressive engine tones.

2.3.3 Crafting the Sound of the Future

An increasing number of companies are hiring specialized musicians and composers to craft iconic sounds for their electric vehicles. One of the most notable collaborations is with Hans Zimmer, the acclaimed film composer, who was engaged by BMW to design the soundscapes for the new BMW i4, iX, and the VISION M Next [4]. Similarly, Renault partnered with the composer and sound designer Jean-Michel Jarre, known for his pioneering work in electronic music, to craft the sound signature for their electric models [15]. Additionally, will.i.am collaborated with Lexus to promote their NX model, emphasizing innovative sound design as part of their marketing campaign.

These initiatives emphasize the importance of sound in enhancing user experience and highlight a growing trend where auditory branding is becoming crucial to a vehicle's identity. As electric vehicles gain popularity (see Section

Figure 2.4: BMW M GmbH and Hans Zimmer in the studio

2.1), developing unique soundscapes is vital for brands to stand out in a competitive market. This trend opens the door for innovative sound design solutions, the most notable being subtractive synthesis, additive synthesis, and granular synthesis. An approach utilizing the latter is presented in this thesis, offering a unique and largely unexplored method for auditory branding in electric vehicles.

3

Granular Synthesis

3.1 Granular Synthesis Overview

Before delving into the detailed implementation, it is important to provide an overview of granular synthesis and how it works, specifically the approach we used in the application. This will be covered in the following chapter.

3.1.1 Historical Background

The modern concept of sound particles originated in 1907 with Einstein's prediction of phononsinaudible packets of ultrasonic energy at very low amplitudes. Dennis Gabor, a student of Einstein, made the crucial breakthrough in the 1940s by introducing the particle model into the domain of audible sound [44].

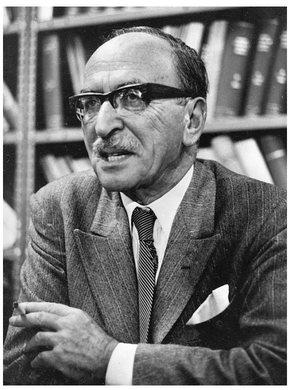

Figure 3.1: Dennis Gabor

In 1946, by applying the concept of quantum physics to the threshold of human hearing, Gabor became the first to conceptualize granular synthesis, laying the foundation for the modern scientific theory of microsounds. He documented his findings and the supporting mathematics in an article titled "Theory of Communication," followed a year later by "Acoustical Quanta and the Theory of Hearing." Initially, Gabor's work was aimed at reducing data requirements for audio communications, given the increasing use of low-band frequencies in telecommunications. Nonetheless, these theories made their way into the world of music composition with Iannis Xenakis, who came across Gabors work and decided to use this technique in his music. In 1959 that Xenakis modified a tape recorder to implement one of the first granular synthesizers.

This process involved splicing magnetic tape into tiny segments, rearranging the segments, and taping the new string of segments together. The first composition to use this technique is called Analogique A et B composed in 1959. From that point onward, granular synthesis gradually evolved into one of the most widely used techniques, particularly in electronic music composition. By the 1960s and 1970s, granular synthesis began to gain traction in electronic music. Pioneering composers like Karlheinz Stockhausen and Pierre Boulez explored granular synthesis to create intricate soundscapes and textures. The technique allowed for the manipulation of sound at a microscopic level, offering new possibilities for sound design and composition.

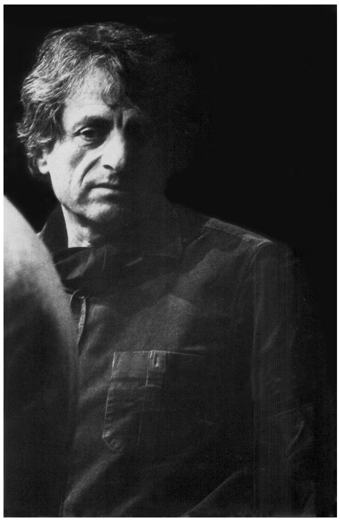

Figure 3.2: Iannis Xenakis

The advent of digital technology in the 1980s and 1990s further expanded the capabilities of granular synthesis. Digital synthesizers and software-based tools enabled more precise control over granular parameters, leading to more sophisticated and versatile sound design. Software like Max/MSP and SuperCollider provided composers and sound designers with platforms to experiment with granular synthesis in real-time. In contemporary music and sound design, granular synthesis is employed in a wide range of applications, including music composition [42, 45], film and game sound design, and interactive media. Its ability to manipulate sound at a microscopic level allows for the creation of intricate textures and dynamic soundscapes, making it one of the most creative synthesis techniques and enabling the production of previously unheard results.

Figure 3.3: Modern digital granular synthesizer

3.2 Granular Synthesis Explanation

Granular synthesis is a sound synthesis method that operates in the realm of microsound and involves generating thousands of very short sonic grains from larger acoustic events. It can be classified as a form of additive synthesis because the final sound is the result of the additive combination of these numerous grains [43].

In granular synthesis, a key concept is the "grain," which refers to a short segment of audio with a quasi-Gaussian bell curve in its amplitude envelope. Typically, the duration of a grain falls within the range of 1 to 50 milliseconds [43]. However, for certain applications, such as automotive sound design, this duration can be adjusted to fall outside this range to achieve specific auditory effects. There are various implementations of granular synthesis, and the one utilized in the proposed approach is Asynchronous Granular Synthesis (AsGS). This method will be discussed in the following section.

Asynchronous Granular Synthesis differs from traditional granular synthesis techniques primarily in how grains are triggered and arranged. In AsGS, grains are activated independently and at irregular intervals, rather than in a synchronized or rhythmic manner. This asynchronous triggering results in a more fluid and unpredictable sound texture, making it particularly well-suited for applications requiring dynamic and evolving audio or simulating unpredictable audio phenomena, such as the rumbling of an engine or environmental sounds.

In this context, randomization plays a crucial role in achieving a sustained sound with minimal fluctuations, which is essential for maintaining a consistent minimum sound level and ensuring that the car always remains audible. Without randomization, the sound could become overly predictable, leading to noticeable gaps or fluctuations that could compromise the vehicle's audibility, particularly at low speeds or during deceleration. This presents a significant challenge, as the unpredictability of random generation cannot be fully controlled. Addressing this issue is critical to ensuring the reliability and effectiveness of the system, and it will be explored in greater depth in Section 3.3.

3.3 Algorithm Explanation

3.3.1 General Structure

The workflow of the algorithm used in this granular synthesis implementation is depicted in the following Figure 3.4.

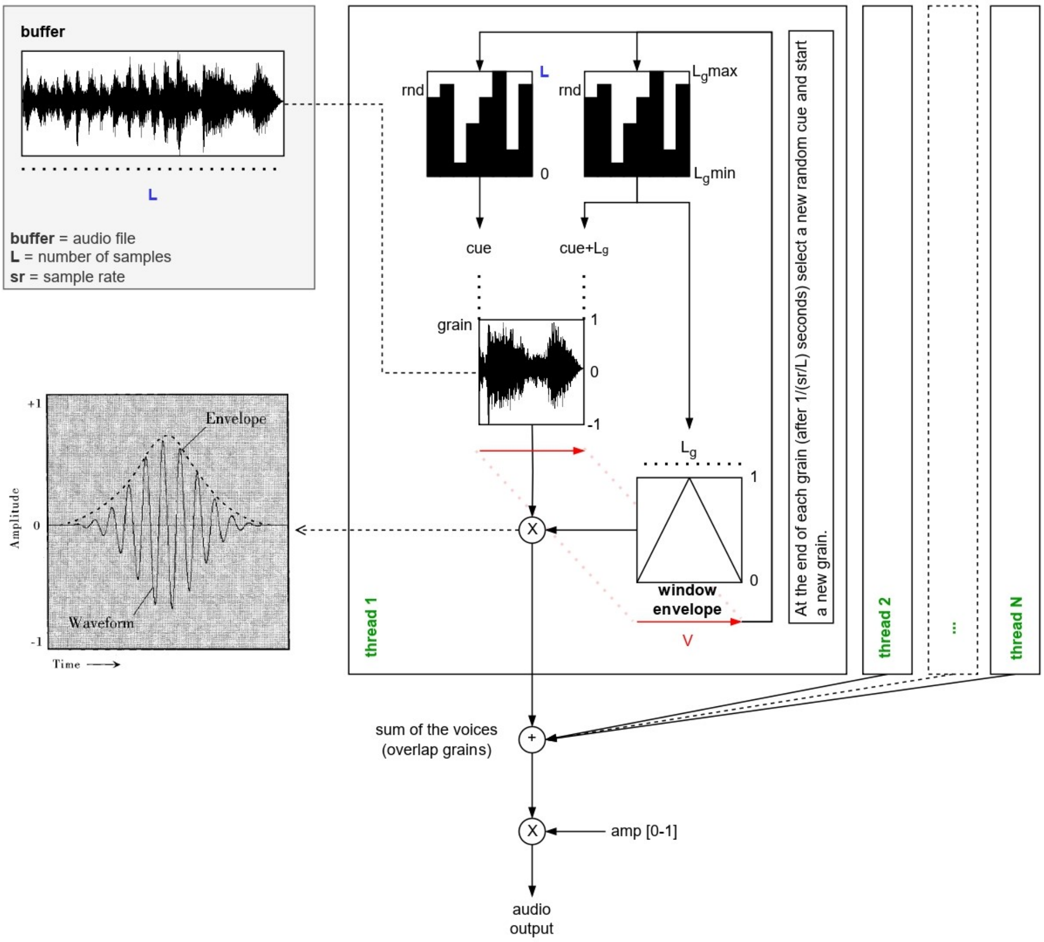

Figure 3.4: Scheme of the Asynchronous Granular Synthesis algorithm

The structures responsible for generating grains are called Threads, and a collection of N Threads is referred to as a Voice. In this setup, the optimal balance between computational efficiency and sound quality was determined through practical experience, leading to a configuration where the synthesizer uses three Voices, each composed of eight independently operating Threads. Each of the three Voices has its own buffer containing an entire audio file from which grains are extracted. This means that within each voice, eight Threads operate on the same audio file, while the Threads in the other voices work on two different audio files. In summary, this implementation uses 24 Threads to extract grains from three distinct audio files.

3.3.2 Threads

Inside each Thread, the randomization occurs in both the starting point and length of each grain. As depicted in the image, each time a new grain is extracted the algorithm generates two random numbers indicating the starting point of the grain within the entire audio file and its length. After the grain is extracted, a window envelope of the same length as the grain is generated and applied to it. At the end of the iteration the grain will look like the one depicted in Figure 3.5.

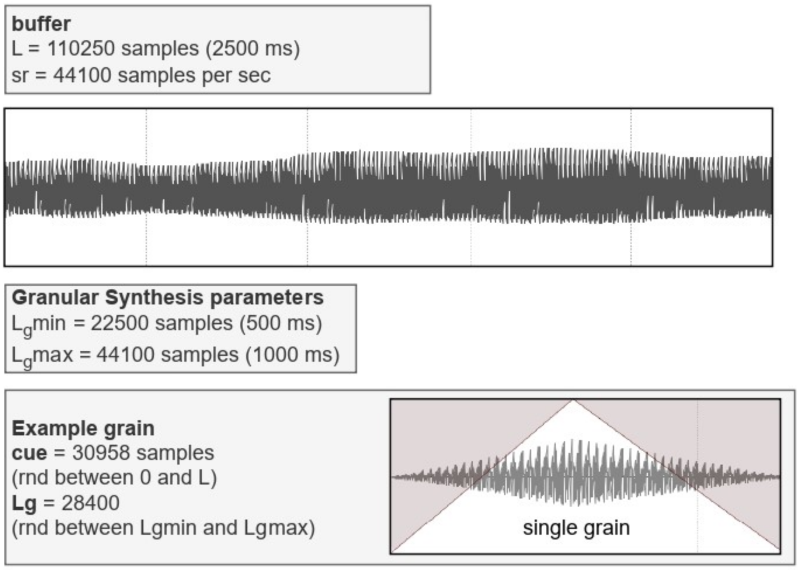

Figure 3.5: Example of grain extraction

Then, a new iteration begins, and another grain is extracted, continuing the process. After several iterations, the final output of each Thread will resemble the example depicted in Figure 3.6. In this way, the asynchronicity ensures that the output of each voice (with eight Threads operating on the same audio file) produces a sustained sound with limited fluctuations in amplitude.

The outputs of each of the 24 Threads will then be summed to represent the final output of the algorithm.

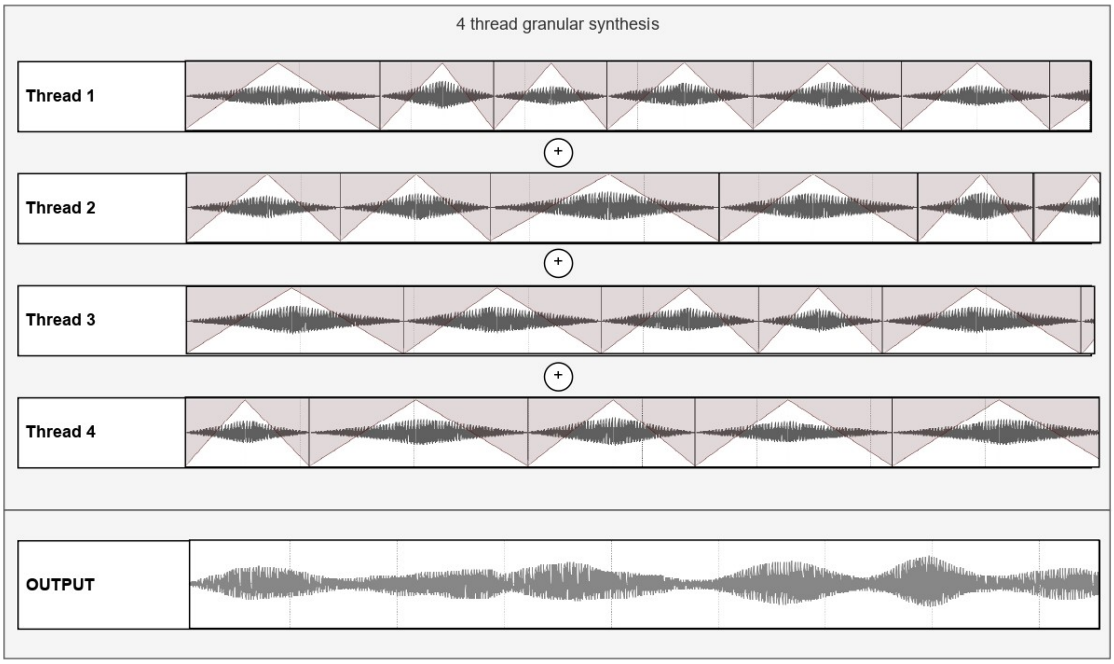

Figure 3.6: Output of 4 Threads summed to obtain the final output of the algorithm

3.3.3 Grains

Given that the algorithm operates in the digital domain, the basic unit is a sample, which represents a single point in the sound waveform at a specific moment in time. These samples are extracted into buffers of 1024 to form grains of various lengths. The mean duration of the grains depends on two factors: the car's engine Revolutions Per Minute (RPM) (which, assuming the car is electric and thus has a single gear, is directly proportional to the car's speed) and user input (which will be discussed in more detail in the following Chapter).

The user can select the minimum and maximum average grain duration, corresponding to the average grain length when the car is at a full stop and at full speed, respectively. The randomization in this algorithm affects the variation in the average grain length, with the amount of randomization being adjustable by the user. This allows for fine-tuning of how much the grain length fluctuates around the selected average. While the mean grain duration is deterministic and ranges from the minimum duration to the maximum duration selected by the user as the car accelerates, the degree of variation from this average can be controlled to produce different acoustic effects.

4 Application

This chapter offers a comprehensive overview of the GUI and the underlying implementation of the application developed for granular synthesis. It starts by introducing the design principles and key features of the GUI, emphasizing its user-friendly approach and functionality. The chapter then delves into the technical aspects of the implementation, discussing the Simulink model block by block to clarify its functionalities. Lastly it will be given a simple example of how to operate the application from its interface. By the end of this chapter, readers will have a thorough understanding of both the visual and technical components of the application and how they integrate to form a powerful tool for sound design.

4.1 Interface

To present all the features and tunable parameters of the AGS, it is essential to begin with an overview of the GUI. Details about the implementation will be covered in Section 4. The GUI, developed entirely using MATLAB App Designer, provides an intuitive interface that allows users to interact directly with the Simulink project and adjust parameters in real time. The interface, shown in Figure 4.1, is designed to be user-friendly and accessible, enabling users to dynamically modify settings and immediately observe the effects on the reproduced sounds through the computers audio output. The following sections will briefly describe all the tunable parameters and features of the synthesizer and their respective positions within the GUI.

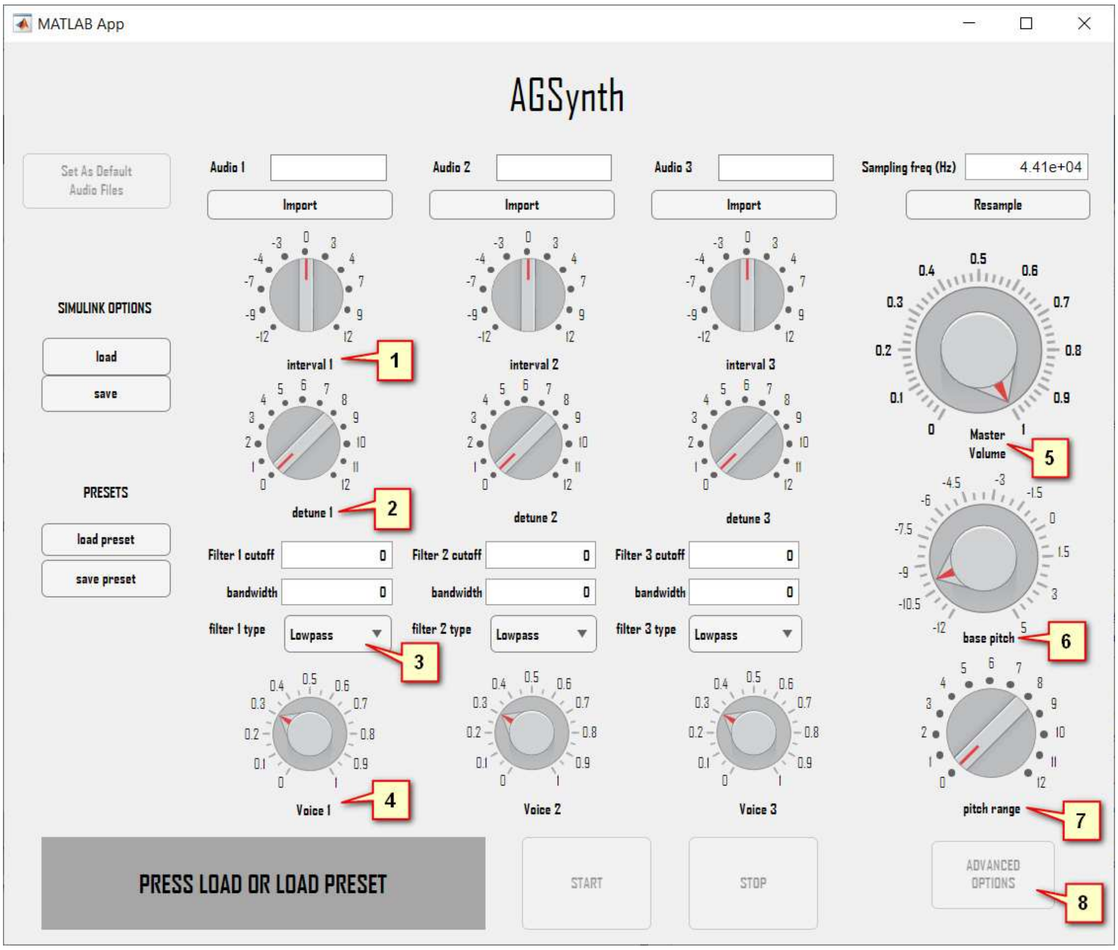

Figure 4.1: GUI of Automotive Granular Synthesizer, Voice parameters

4.1.1 Algorithm Parameters

The parameters of the Voices are organized in columns, as evident from the disposition of the numbers 1, 2, 3 and 4 displayed in Figure 4.1. Lets start presenting their tunable parameters:

1. **Interval:** enables to adjust the pitch of the input audio file for each voice. The intervals are measured in semitones and span two octaves, ranging from one below to one above the original pitch of the sound.

2. **Detune:** allows to select the amount of dynamic pitch shifting in semitones for each voice. The detuning reaches its maximum value when the car is at its highest RPM. For example, if Detune 1 is set to 7 semitones, Thread 1 will maintain its original pitch (selected by Interval 1) when the car is at its lowest RPM. As the car accelerates to reach its maximum speed and

the engine reaches its maximum RPM, the pitch will then shift upward by +7 semitones.

3. **Filter:** Allows control over filter parameters, such as bandwidth and cutoff frequency. The dropdown menu enables the selection of the filter type, with options including Lowpass, Bandpass, and Highpass (more details about filters will be provided in the following chapters).

4. **Voice Volume:** this knob represents the volume of each voice.

These parameters are specific to each voice, which is a collection of 8 threads operating with the same input audio file. Adjustments to the parameters of a voice (like pitch) will affect all 8 threads within that voice.

Lets now present the parameters that affect all the voices simultaneously. These controls are located on the right side of the interface and are shown in Figure 4.1.

5. **Master Volume:** allows to set the gain of all the voices at the final stage.

6. **Base Pitch:** simultaneously regulates the initial pitch of all threads, functioning similarly to the Interval knobs but applying to all threads at once. Its important to note that the effect of this parameter is combined with that of Interval. When both are used, the final pitch shift in semitones for each thread will be the sum of the two parameters.

7. **Pitch Range:** simultaneously regulates the dynamic pitch shift of all threads, functioning similarly to the Detune knobs but applying to all Threads at once. Like with Base Pitch and Interval, the effect of this parameter is combined with that of Detune, so the amount of pitch shift reached at the highest RPM and speed will be the sum of the two parameters (for each Voice).

8. **Advanced Options:** opens the advanced option tab, allowing to adjust some additional parameters that will now be presented.

The Advanced Options window appears as shown in Figure 4.2. Here its possible to adjust:

1. **Min Grain Size:** represents the average size of the grain when the car reaches its maxi- mum speed (and highest RPM).

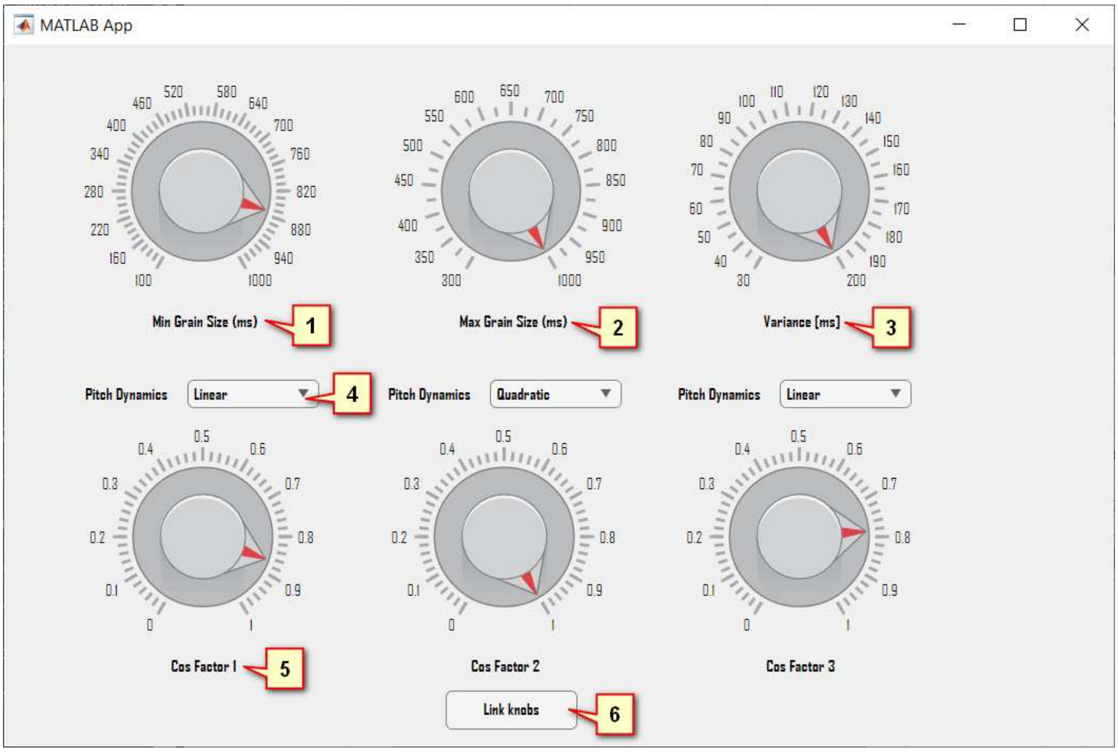

Figure 4.2: Advanced Options Tab

2. **Max Grain Size:** represents the average size of the grain when the car is stopped, i.e., when the engine is at its lowest RPM.

In the range between the maximum reachable speed and the lowest speed, an average grain value is computed. As previously stated, as the car accelerates, the average grain size changes smoothly from the maximum (selected using knob 2) value to the minimum value (knob 1). The opposite occurs when the car brakes to come to a full stop; the average grain size varies from the minimum value to the maximum value.

3. **Variance:** This parameter represents the variation in the average grain size at each iteration. As explained earlier, a degree of randomization in the length of the grains is necessary to ensure the algorithm's effectiveness. This control allows the user to set the maximum difference from the average grain duration. Ultimately, the final duration of each grain will fall within the range $AverageGrainSize \pm Variance$.

4. **Pitch Dynamics:** this setting controls the exponent of the pitch dynamic. Selecting *Linear* provides the default result, maintaining a steady pitch

progression and reaching the final pitch set in Pitch Range. Choosing *Quadratic* and *Cubic* allows the pitch to rise faster and reach higher levels, resulting in a more pronounced pitch variation over time. For more details refer to Section X.

5. **Cosine Factor:** allows you to adjust the attack of each grain. As explained in Section, a windowing process is applied to each grain. In our algorithm, we use the Tukey window, which will be discussed in detail in Section 3.3.2. Varying the Cosine Factor affects the attack duration of the window, ranging from a rectangular window ($\alpha = 0$) to a Hann window ($\alpha = 1$). Each voice has its own independent Cosine Factor.

6. **Link Knobs:** by pressing this button, the user can link the Cosine Factor of Voice 2 and 3 to the Cosine Factor of Voice 1. This allows the user to adjust the attacks of each Thread simultaneously using the first knob on the left.

4.1.2 Utility Features

After presenting the tunable parameters that offer flexibility in sound design, the basic commands that enhance the application's usability are introduced. These commands enable direct interaction with the Simulink model, allowing users to save and load the model within the interface, as well as save and load presets: user-defined patches that can be recalled as needed. The following section briefly presents the basic commands, highlighted in Figure 4.3.

1. **Load Button:** This button loads the Simulink model and runs the initial scripts. Once loaded, the Simulink parameters become available and can be adjusted directly within the interface.

2. **Save Button:** it saves the Simulink model and its parameters (equivalent of CTRL+S in Simulink).

3. **Load Preset Button:** it allows the user to select a preset from the Preset folder and it loads all the parameters into the Simulink model and the GUI.

Figure 4.3: GUI of Automotive Granular Synthesizer, utility features

4. **Save Preset Button:** it allows the user to save the current state of the app into a preset file in the Preset folder. The presets are saved in a numerical file whit .m extension. These files are not standalone and can only be imported using the application.

5. **Start Button:** this button initiates the simulation of the model. After a brief loading period (depending on the performance of the machine), the simulation's output will be audible, allowing for real-time adjustments to the parameters while immediately hearing their effects.

6. **Stop Button:** it stops the simulation.

7. **Status Display Panel:** this panel provides real-time updates on the current state of the application, assisting users by guiding them through its functions.

8. **Import Button:** lets the user select the desired audio file from the *Audio* folder.

9. **Set as Default Audio Files Button:** this button allows the user to designate the current set of audio files as the default. When selected, this default set, along with all associated patch parameters, will be automatically loaded when pressing the Load button.

10. **Resampling Button:** this button allows the user to resample all the audio files to the same sampling frequency. By default, audio samples are resampled at the frequency specified in the Sampling Frequency box, with the default value set to 44,100 Hz. The resampling script is implemented to provide greater flexibility when selecting audio files with different original sampling frequencies. By resampling, it is possible to reduce artifacts that may arise from mismatched sampling rates.

11. **Audio Text Box:** lets the user check the current audio file loaded for the specific voice. As previously stated, 3 different audio files can be loaded for each patch.

4.1.3 Controller Window

The Controller Window facilitates the management of car simulation parameters, enabling control over the car's behavior, including its speed and engine RPM. This window, as shown in Figure 4.4, appears when the simulation starts and closes when the simulation stops. The following features are available:

1. **Switch Control Type:** allows selection between Cruise Control mode and Manual Control:

    - In Cruise Control mode, the user controls the desired speed for the vehicle to reach.
    - In Manual Control mode, the user controls the virtual pedals, with the ability to press both pedals simultaneously.

2. **Speed Slider:** active only in Cruise Control mode, allows you to choose the desired speed for the car to reach.

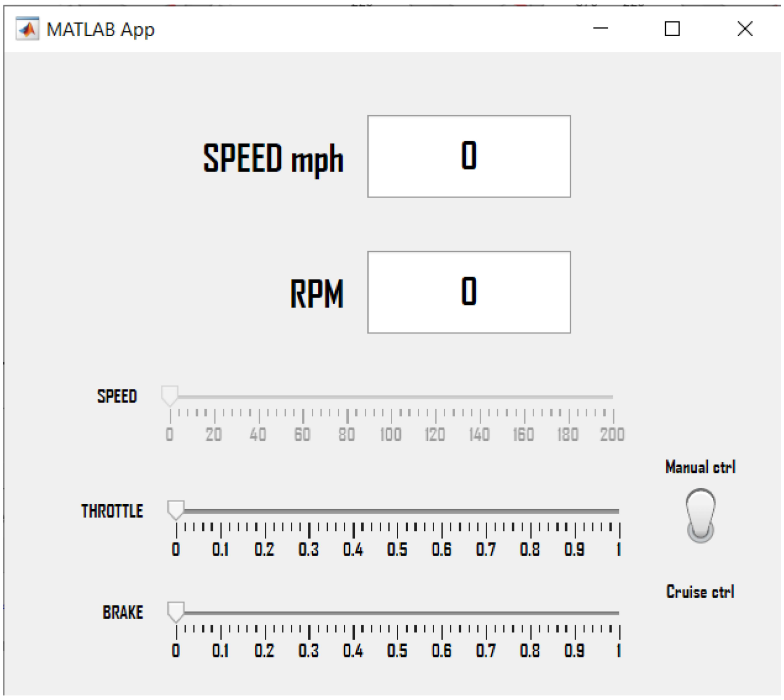

Figure 4.4: Controller Window

3. **Throttle and Brake Sliders:** active only in Manual Control mode, allows to press the imaginary pedals, a value of 1 means that the pedal is completely pressed.

4. **Speed Display:** shows the current car speed in mph.

5. **RPM Display:** shows the current engine RPM.

4.2 Implementation In-Depths

The implementation of the previously presented algorithm was carried out using Simulink and MATLAB, as requested by FIAMM. After presenting the synthesizers structure, this chapter provides insights into how the granular synthesis algorithm and all its features were developed.

4.2.1 Folder Structure

The folder containing all the files can be accessed on GitHub at the following link: `https://github.com/alessandrofiordelmondo/fiamm-sound`. This repository includes all the necessary source code, scripts, and documentation related to the project. Inside the Audio folder, various audio files are available. Users can expand the library by adding their own .wav files inside this folder, which can then be used with the synthesizer for a more customized sound design experience.

The Presets folder contains all the patches created by the developers. When users save their own presets, the corresponding files are stored in this folder. These saved presets can then be easily loaded, allowing users to revisit and modify their custom sound designs as needed. Each preset is saved as a .mat file, a MATLAB data format that stores the values of each individual parameter of the synthesizer, along with the associated audio file and sampling frequency. These files are not standalone; opening them outside the application will simply create new variables in the MATLAB workspace. They are intended to be loaded only through the GUI, specifically using the "Load Preset" button.

In addition to the Presets and Audio folders, the project contains various Simulink models, MATLAB scripts, and applications, which will be addressed in detail shortly. The primary task for the end user is to open the *app1.mlapp* file by double-clicking it in the file explorer. The application will then automatically utilize the necessary scripts for tasks such as loading, resampling, and storing data.

The *app2.mlapp* and *app3.mlapp* files represent the Advanced Options window and the Controller window, respectively. These windows open automatically when the "Advanced Options" button is clicked for the former or when the simulation is started using the "Start" button for the latter.

The Simulink model, *merge_variable_grain.slx,* contains the core implementation of the synthesizer. The GUI serves as an interface for the user to comfortably interact with the synthesizer, allowing for real-time parameter adjustments while the simulation is running. Additional details on the model will be provided in the following Section 4.2.2.

Other files in the project either serve as utilities that the application uses for common functions (such as resampling or saving and loading presets and audio files) or represent components of the main model that are complex enough to be stored separately (such as the implementation of a Thread). These files are not intended to be opened by the end user and will be clarified and explained in detail in Section X.

4.2.2 Model Structure and Implementation

This section aims to provide an in-depth explanation of the workflow involved in our implementation of the granular synthesis algorithm. The goal is to detail the processing chain and data flow within the model used to design the application AGS. To achieve this, we will reference the Simulink model underlying the application, named *"merge_variable_grain.slx"* and visible in Figure 4.5. Each block within the model will be examined, with its functionalities explained in the context of the main model and its role within the chain of surrounding blocks. Note that the blue box on the left merely displays the values of speed and RPM, which is not particularly relevant in this explanation, especially because there is another display that the user should utilize: the Control Panel described in Section 4.1.3.

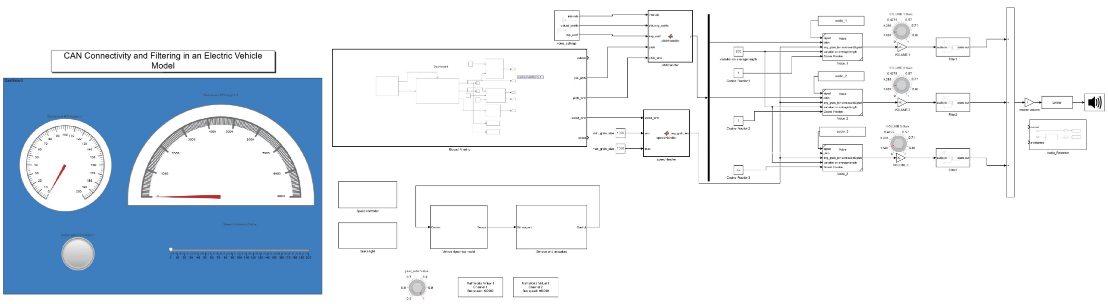

Figure 4.5: AGS Simulink Model

4.2.3 Phisycal Modeling of the Car: Sensors and Actuators

The algorithm is based on a model of a real car, which provides the essential framework for the synthesizer to function effectively. For the synthesizer to produce realistic and responsive audio output, it requires specific input data such as vehicle speed and engine RPM. This data is crucial because it directly influences the synthesis process, allowing the generated sounds to accurately reflect the car's behavior under different driving conditions.

In this context, speed and RPM serve as dynamic inputs that modulate various parameters of the granular synthesis. For instance, changes in the car's speed can affect the pitch of the synthesized sound and the length of the grain (as explained in Section 3.3.3).

These inputs are typically derived from the car's onboard sensors in a real-world application, but in a simulated environment, they can be generated by a virtual model of the vehicle. This model replicates the physics of driving, providing the necessary data for the synthesizer to create a realistic soundscape that responds to the car's and users actions in real-time. The accuracy and realism of the synthesized sounds depend heavily on the quality and precision of the input data, making it a critical component of the overall system. We wont focus extensively on this part of the model, as it goes beyond the primary scope of this thesis. However, a brief explanation of all the relevant blocks will be provided to give a clearer understanding of the system's operation.

The communication and data transfer between Electronic Control Units (ECU) within the vehicle are handled through a vehicle bus standard known as Controller Area Network (CAN). This standard is widely adopted in the automotive industry due to its flexibility, reliability, and efficiency in managing the exchange of data between various electronic systems within a vehicle.

The two main blocks that play a crucial role in simulating the car's behavior are the *Vehicle Dynamics Model* and the *Sensors and Actuators*. As can be observed in Figure 4.6, these two blocks are connected via a feedback loop. This configuration is essential because the blocks are interdependent: the output of the *Vehicle Dynamics Model* affects the *Sensors and Actuators*, and vice versa. For example, to accurately compute the speed of the car during acceleration, it is essential to know the amount of throttle applied. Similarly, for accurate braking simulation, the system must account for the amount of braking force being applied. The *Vehicle Dynamics Model* uses this information to calculate changes in speed and other dynamic parameters, which are then fed back into the *Sensors and Actuators* block. This block, in turn, adjusts the control signals based on the users actions via the actuators and generates the output variables needed for the granular synthesis algorithm through the sensors.

The *Vehicle Dynamics Model* depends on various fixed variables, the main ones are:

- Mass of the Chassis.
- Mass of the Wheels.
- Wheel Radius.
- Gear Ratio.

In this context, the *Gear Ratio* represents the ratio of the number of rotations of the motor (electric motor) to the number of rotations of the wheels. This ratio

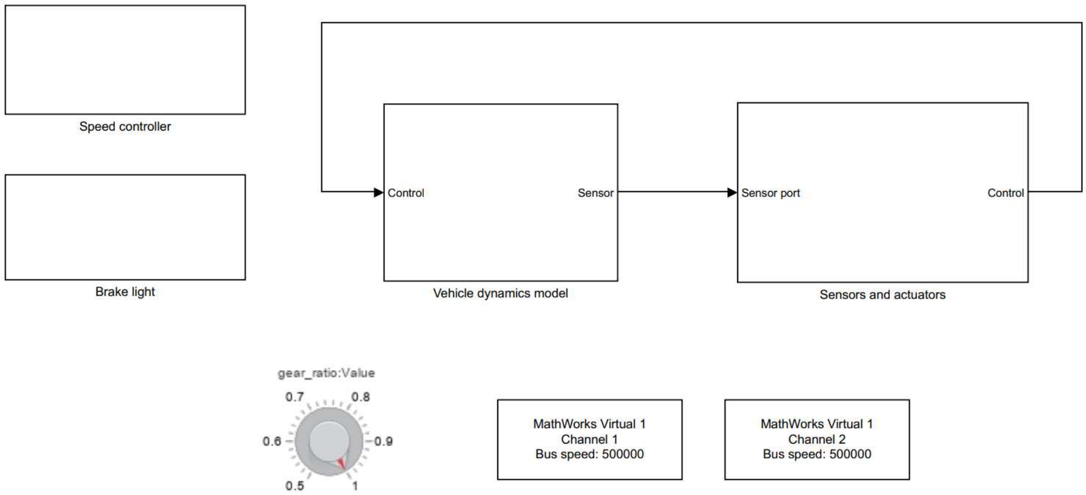

Figure 4.6: Blocks responsible for the phisycal modeling of the car

is crucial for translating the motor's rotational speed into the wheel's rotational speed. All the other constants are self-explanatory.

Sensors and Actuators block is responsible of simulating the acquisition of signals and physical values coming from the vehicle model, computing and transmitting the control action and logging data into CAN Bus. Its input are the actual Speed computed using the *Vehicle Dynamics Model* and Speed Command (selected by the user) coming from CAN Bus, while its output is the Throttle/Brake command that will be transmitted using the CAN bus.

Speed Controller block computes the proper Throttle / Brake CAN Bus command to feed into the *Sensors and Actuators* block. It implements both cruise control and normal driving mode. The controller of choice for cruise control is a simple discrete PI controller.

4.2.4 From Sensor Data to Input Data

To ensure the proper functioning of the algorithm, the data collected from sensors on the CAN bus must be converted into parameters suitable for granular synthesis. Specifically, this involves translating the car's speed and engine RPM into pitch and grain length parameters. The following Section 4.2.4 will detail this conversion process.

The block responsible for this task is called *Biquad Filtering*, as illustrated in Figure 4.7. This block takes as input the speed, RPM, and throttle command from the CAN bus and outputs variables that can be input to the granular synthesis

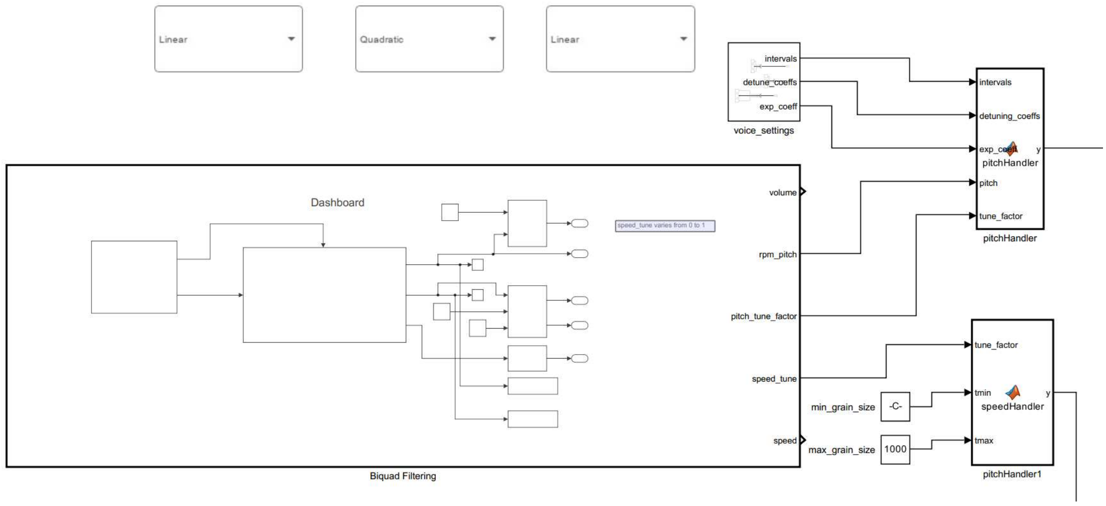

Figure 4.7: Blocks that transform sensor data into input data for the algorithm

algorithm, such as pitch and speed. Additionally, it provides a volume variable, which is intended to automate the volume of the final output, louder sounds when the car is moving quickly, for example. However, in this application, the volume is set to a fixed level and is not utilized dynamically.

Within the *Biquad Filtering* block, two additional blocks–*speedToTune* and *rpm-ToPitch*, visible in Figure 4.8–are responsible for converting and adjusting the output data to ensure it is meaningful for the pitch handlers outside the *Biquad Filtering* block:

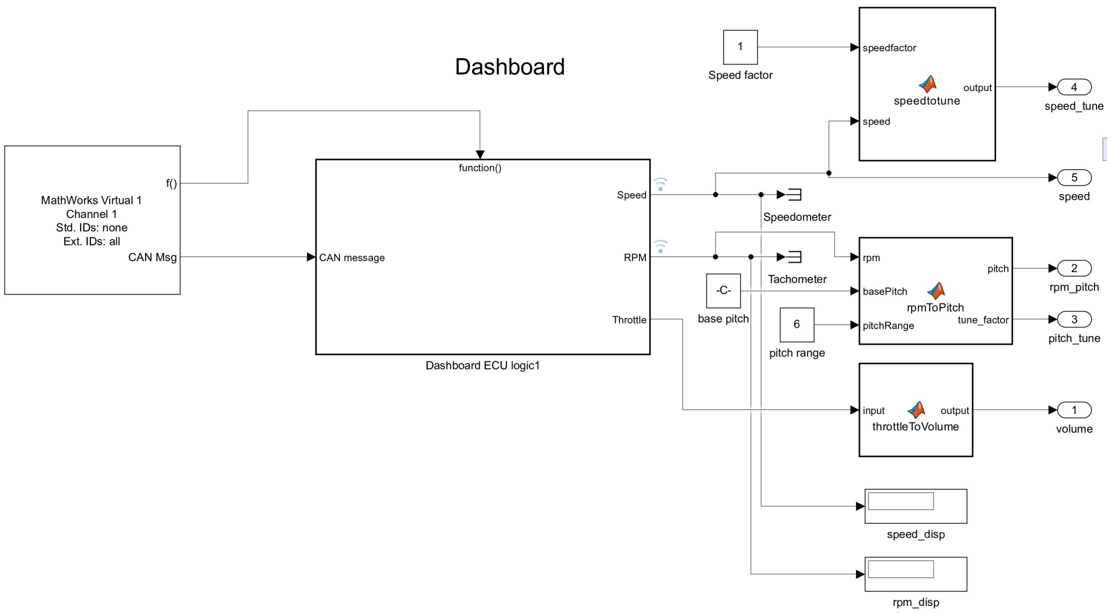

Figure 4.8: The blocks that make up the Biquad Filtering block

- *speed_tune*, which is a variable ranging from 0 (when the car is at full stop) to 1 (when the car is at full speed).
- *rpm_pitch* is a variable that ranges from a base pitch to a certain range that is specified in the GUI. (See section 4.1.1)
- *pitch_tune_factor* is a variable ranging from 0 to 1 that increases with linearly with the RPM.

Outside the *Biquad Filtering* block, there are three additional blocks (the ones on the right in Figure 4.7) that complete the data conversion by integrating user-specified inputs. The *Voice Settings* block stores the *Intervals*, *Detune*, and *Pitch Dynamics* selected for each Voice through the GUI. These values are used as inputs by the *Pitch Handler* block, along with the pitch output from the *Biquad Filtering* block. The final pitch of the grain is then determined by adjusting the initial pitch with the *Pitch Dynamic* value, adding the *Interval*, and applying *Detune* incrementally by multiplying it by the *Pitch Tune Factor*. This is the formula for computing the pitch that will then be fed inside the Granular Synthesizer:

$$y = pitch^{exp} * sign(pitch)^{mod(exp,2)} + interval + (detuning * pitch\_tune)$$

To account for negative pitch values that would be lost when raising to an even power, the formula includes a sign factor. This adjustment ensures that the sign of the original pitch is preserved when necessary.

The final pitch, denoted as y, will be input to the Granular Synthesizer. This value is computed in every iteration for each Voice, making y a dynamic vector containing three elements. Each element of the vector corresponds to a Voice and is continuously updated to reflect both the car's behavior (such as accelerating or braking) and the user's settings.

Let's now discuss another crucial variable that serves as an input to the granular algorithm: the average grain length. This value is calculated based on the car's speed. However, instead of using the absolute speed, the algorithm relies on *speed_tune*, a relative indicator that represents the proportion of the car's total speed. As explained earlier, *speed_tune* ranges from 0 to 1, where 0 corresponds to the car being at a complete stop, and 1 indicates that the car has reached its maximum speed. The block responsible for this conversion is called *speed_handler*. It takes as input the *speed_tune*, *Minimum Grain Size (Tmin)*, and *Maximum Grain Size (Tmax)*, which are two variables controlled by the user that represent the minimum and maximum average duration of the grains when the car is stationary and at maximum speed, respectively. The block operates by converting the *Tmin* and *Tmax* values from milliseconds to samples by multiplying them by the sampling frequency of the audio files. It then calculates the average grain length using the following formula:

$$avg\_grain\_len = Tmax - (speed\_tune) * (Tmax - Tmin)$$

This means that the average grain length is dynamically adjusted based on the car's speed, with shorter grains at higher speeds and longer grains at lower speeds. The average grain length is recalculated each time a new grain is extracted, ensuring that, before randomization is applied, the duration of each specific grain corresponds to the car's speed at that moment.

4.2.5 Granular Synthesis Implementation

With the car's behavior simulation and real-time data generation now explained, we can turn our attention to the granular synthesis algorithm. The two key inputs from the data generation blocks are the pitch and the average grain length. These inputs shape the sound produced by the granular synthesizer, making sure it reflects the car's real-time dynamics.

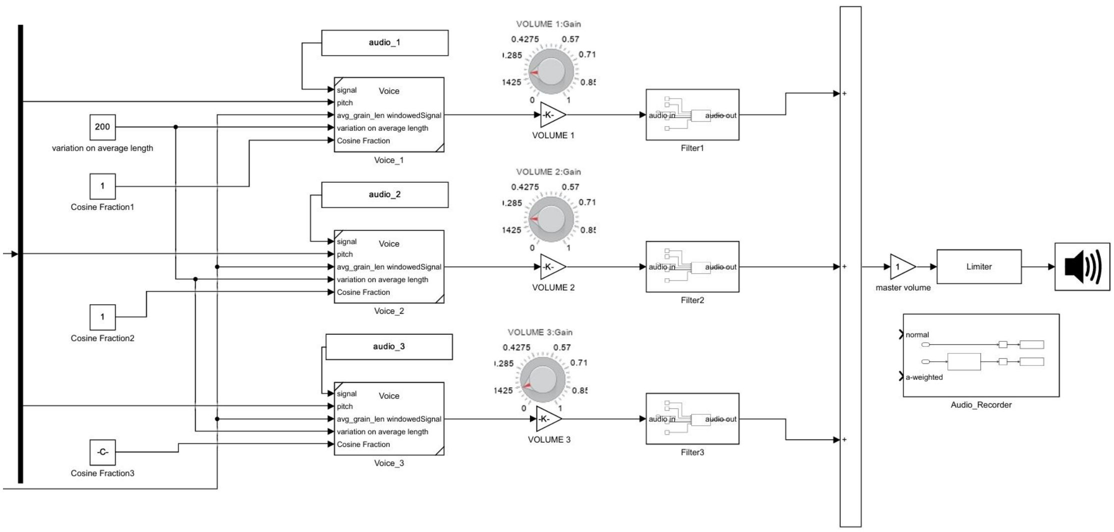

Figure 4.9: Granular synthesis implementation in Simulink

The blocks presented below are responsible for the process of generating and reproducing grains and can be seen in Figure 4.9. As said earlier, this implementation of granular synthesis is composed of 3 Voices, each operating independently and containing 8 Threads. Within the Voice block lies the implementation of the algorithm itself (4.10). To fully understand this implementation, it is essential to explore how randomization functions within the system. As discussed in Section 3.3.2, randomization plays a crucial role in AsGS, serving as the primary mechanism for achieving a continuous sound, which is the goal of this application. In this algorithm, randomization is applied to the starting point and, most importantly, the length of the grains, which in turn affects the start time of the next grain, as each Thread extracts and plays grains sequentially. Based on the users inputs for minimum and maximum grain length, as well as the car's speed, the average grain length input to the synthesizer is deterministic.

Handlers

Each Voice contains 8 *handlers*–one for each Thread, resulting in a total of 24 *handlers*. These *handlers*, shown in Figure 4.11, apply randomization as the first crucial step of the algorithm; In other words, the *handlers* act as random number generators, with certain precautions.

They take the average grain length and the user-defined variance around this length (in milliseconds) as inputs, then output the cue point (starting point) within the input audio file and the grain's length, which can vary within the range of the average grain length ± the specified variance. It is important to note that the grain length output by the handler is not an average but the actual length of the grain to be extracted in the current iteration. The cue point is also randomly selected to ensure that the entire length of the grain fits within the boundaries of the audio file. This process prevents the grain from extending beyond the start or end of the file, ensuring that the extracted grain is always fully contained within the audio data. For better understanding, a code snippet of the *handlers'* implementation is provided below:

```
% INPUT: total input audio length (sample_dim_f), average computed length
% of the grain (avg_len), variation on the length of the grain in ms
% (var_ms), seeds of the random number generator (rng_seed_f), sampling
% frequency (fs)

```

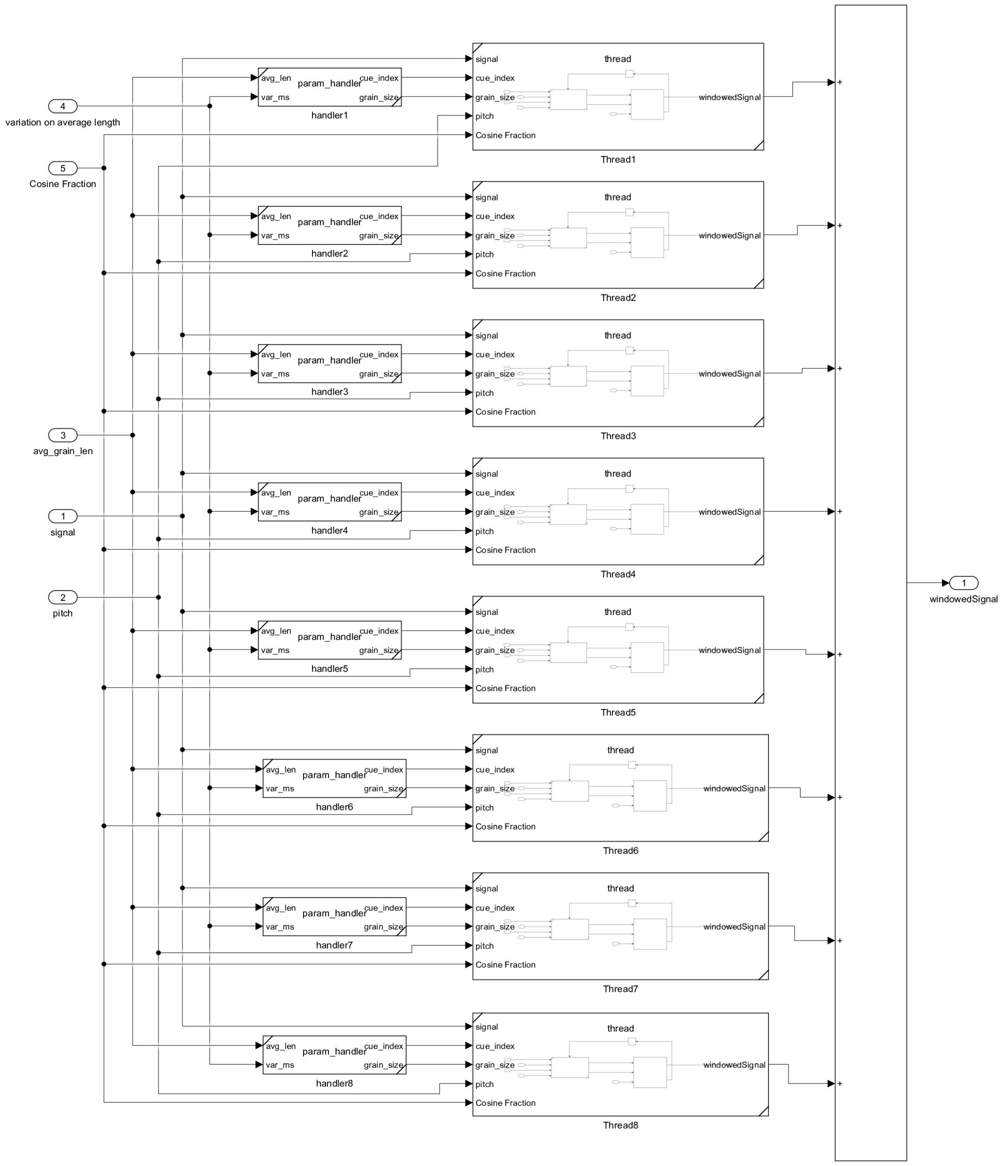

Figure 4.10: Implementation of a Voice

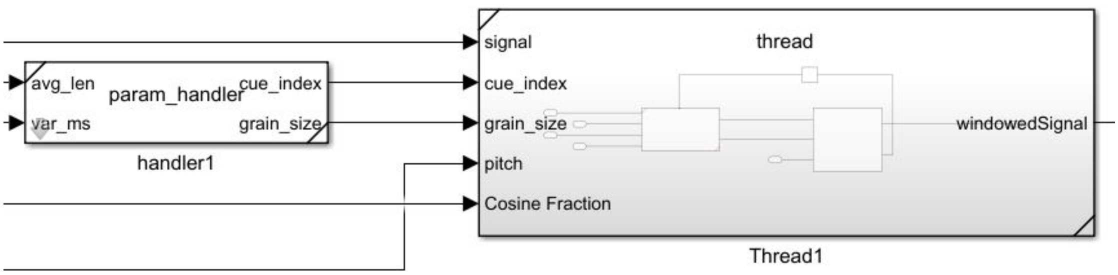

Figure 4.11: Implementation of a Thread

```
% OUTPUT: starting point of the grain (cue_index), size of the grain (grain_size)

% This function takes in input the length of the grain, adds some
% variation, and outputs the grain size and the cue point of the randomly extracted grain
function [cue_index, grain_size] = parametersHanlder(sample_dim_f, avg_len, var_ms, rng_seed_f, fs)

    persistent random_start;
    if isempty(random_start)
        random_start = 1;
    end
    avg_len = min(sample_dim_f, avg_len);  % Grains can't be longer than the input audio file
    rng(rng_seed_f);                       % Different seed for each handler
    var = ceil((var_ms/1000)/(1/fs));
    if random_start == 1                   % In the first iteration increase variance in any case
        var = 40000;
        random_start = 0;
    end
    % If the maximum grain duration is too big avg_len must be equal to the input audio length
    % otherwise min_grain_size could be equal to max_grain_size
    min_grain_size = min(max(avg_len - var, 2048), sample_dim_f - 1);  % Lower bound for short grains 2048 samples
    max_grain_size = max(min(avg_len + var, sample_dim_f - 1), 1);
    grain_size = randi([floor(min_grain_size), floor(max_grain_size)]);
    cue_index = randi([1, sample_dim_f - floor(grain_size)]);
end
```

Code 4.1: Code snippet of the function that selects grains

As observed, each *handler* operates with its own random seed, ensuring that the final grain lengths differ for each Thread at each iteration. Additionally, it is essential to consider that users may select audio files of varying lengths, which can be either very long or very short. Therefore, the grain length must be adapted to fit within the constraints of the selected audio file.

For example, if a user selects a very short audio file and specifies longer grain lengths, the algorithm must ensure that the generated grains do not exceed the length of the file. In such cases, both the grain length and cue point are adjusted so that the entire grain fits within the bounds of the audio file. Conversely, for a very long audio file, the algorithm has greater flexibility in determining grain lengths but still ensures that the randomly selected cue point and grain length remain within the file's boundaries.

Additionally, if the user selects a high variance for grain length, very short grains might be generated. To address this, a lower bound of 2048 samples is imposed on the grain length. Very short grains can cause issues during the simulation, as the software may not process and reproduce the sound correctly, leading to data loss and errors.

Another issue to address occurs when the average grain length is significantly larger than the audio file, even after subtracting the variance. In such cases, the selected length will always equal the length of the audio file, as accounting for the boundaries of the file results in the minimum and maximum grain sizes both being equal to the audio length. In this scenario, randomization becomes ineffective, as each iteration will output a grain of fixed length under these conditions. To resolve this, the average grain length will be adjusted to match the length of the input audio file. This adjustment ensures that the minimum extractable grain size will always be less than the maximum extractable grain size, thereby maintaining effective randomization.

After extracting the grain size according to the rules outlined above, the window for selecting the cue point so that the entire grain is contained within the audio file is $[1, audio\_length - grain\_size]$.

Thread

The subsequent step in the processing chain is managed by the block called *Thread*. As the name suggests, together with the *handler* block, it constitutes the complete implementation of a Thread as explained in Section 3.3.2. *Thread* is responsible for several tasks, including the extraction of samples from the audio file, the repitching process, and the windowing of the grain. *Thread* takes in input the whole input audio file, the cue point and grain length output by the *handler*, the pitch (which is the unaltered output of the block *pitchHandler* explained in the previous Section 4.2.4) and the Cosine Fraction. At each iteration, it outputs a buffer of 1024 samples of the windowed and repitched grain.

To provide a comprehensive understanding, the implementation of each of these processes will now be detailed by examining the inner workings of the *Thread* block, as shown in Figure 4.12.

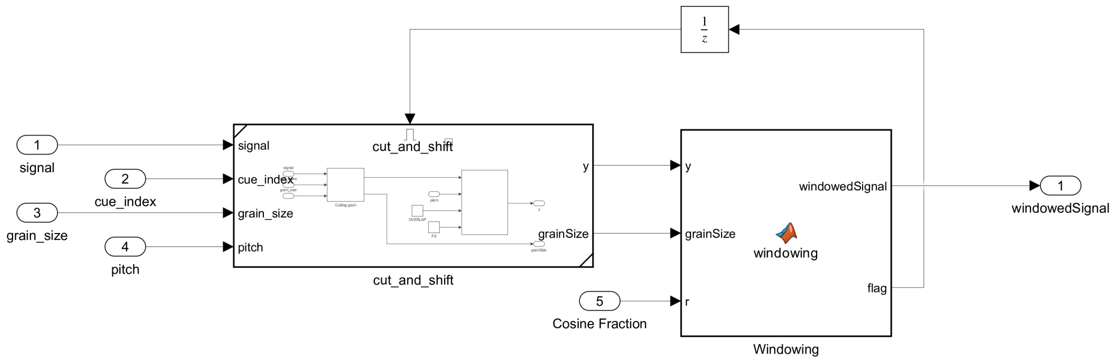

Figure 4.12: Inner workings of the *Thread* block

Extracting grains and pitch shifting

The first block is called *cut_and_shift*. As the name suggests, this block is responsible for extracting the grain from the audio file and then pitch-shifting it according to the input parameter. This subsystem is enabled by a flag that indicates when the previous grain has been fully output and a new one is needed (further details will be provided in the following Section). The flag ensures that the grain extractor operates only when necessary, preventing unnecessary processing and data loss. The extraction process is straightforward: given the starting point and length of the grain to be extracted, the function selects the samples and outputs them as variable-sized data, along with the size of the grain, which is represented as a scalar. Variable-sized signals can create limitations in Simulink because not every block supports them as input parameters. This is one of the primary reasons why, in this implementation, many MATLAB Function blocks are used (more on this in Chapter X Problems/Limitations).

Regarding the repitching of the grain, the function utilizes the MATLAB object *audiopluginexample.PitchShifter*. This audio plugin implements delay-based pitch shifting, adjusting the pitch according to the number of semitones specified as an input parameter. The *PitchShifter* plugin is well-suited for our implementation due to its optimization for real-time audio processing and its ability to preserve audio quality even after pitch shifting. For more details about the implementation and functionalities of this object, please refer to the MATLAB documentation, as this is not the primary focus of this thesis.

Windowing

At this point in the processing chain, an array containing the samples of the repitched grain is available for each Thread. The next step is to apply the window function and above all to manipulate the data so that the output is a buffer of fixed size, as requested by FIAMM. These operations are both implemented inside the same block called *Windowing*.

As explained earlier, the simulation must handle variable-sized signals due to the characteristics of the underlying algorithm. This requirement complicates the conversion between variable-sized and fixed-sized signals. Initially, it was considered that using the *Buffer* block would be the simplest and most efficient solution. However, the *Buffer* block cannot accept variable-sized signals as input, which would disrupt the fixed step size of the simulation. Maintaining a fixed step size is essential because the simulation must ultimately reproduce real-time audio accurately.

To address this challenge, a MATLAB Function block, along with persistent variables, was employed. Although this solution is not optimized for performance, it was the only viable method for managing the conversion between variable-sized and fixed-sized data within the constraints of the simulation. Persistent variables in the MATLAB Function block enable the storage of state information across iterations, facilitating the handling of variable-sized inputs. Despite its limitations, this approach ensures that the simulation maintains a consistent step size, allowing for real-time reproduction of the output audio.

The code for the windowing function is provided below, followed by a brief explanation.

```
% INPUT: extracted grain (y), size of the grain (grainSize), output audio
% buffer (spf), cosine fraction (r)

% OUTPUT: portion of the windowed signal of length spf (windowedSignal),
% flag that identifies the end of the grain

% If the grain size is not a multiple of spf, the function completes the
% buffer with the starting samples of the next grain

function [windowedSignal, flag] = windowing(y, grainSize, spf, r)
```

```
    % r is the cosine fraction:
    % If r == 0 -> rectangular window
    % If r == 1 -> hann window
    windowSize = grainSize;
    % Initialize persistent variables
    persistent it window windowedBuffer grain sz;
    % Initialize flag and sz
    flag = 0;
    if isempty(grain)
        grain = 0;
    end
    % Check if it is the first call or window needs to be initialized
    if isempty(it) || isempty(window)
        it = 1; % Initialize iteration count
        window = tukeywin(windowSize, r); % Initialize window
    end

    if isempty(sz)
        sz = 0;
    end
    % Check if windowedBuffer needs to be initialized
    if isempty(windowedBuffer)
        windowedBuffer = -ones(1024, 1);
    end
    windowedBuffer(windowedBuffer == -1) = [];
    % Append the buffer from previous grain (if there is any)
    if it == 1
        grain = grainSize;
        window = tukeywin(grain, r);
        y = [windowedBuffer; y]; % Append buffer to the beginning of
u
        windowedBuffer = [];
    end

    if it == 1 && sz == 0
        startIndex = (it - 1) * spf + 1; % 1
        endIndex = startIndex + spf - 1; % 1024
        windowedSignal = y(startIndex:endIndex) .* window(startIndex:
endIndex);
    elseif it == 1 && sz ~= 0
        startIndex = uint32((it - 1) * spf + 1); % 1
        endIndex = uint32(startIndex + spf - 1); % 1024
        windowedSignal = [y(startIndex:startIndex + sz - 1);y((
```

```
    startIndex + sz):endIndex) .* window(startIndex:(endIndex - sz))];
    else
        startIndex = (it - 1) * spf + 1 - sz;
        endIndex = startIndex + spf - 1;
        windowedSignal = y(startIndex:endIndex) .* window(startIndex:
    endIndex);
    end


    % Increment the iteration count
    it = it + 1;
    % Check if we've reached the end of the window
    if (it * spf - sz) >= grain
        % Reset the iteration count
        it = 1;
        % Store the remaining samples for the next iteration
        windowedBuffer = y(endIndex + 1:end) .* window(endIndex + 1:
    end);
        sz = size(windowedBuffer, 1);
        % Set the flag to indicate the end of the grain
        flag = 1;
    end
end
```

Code 4.2: Code snippet of the windowing function

As explained earlier, the output of this function must be of fixed length; for this application, buffers of 1024 samples are used, which is half the minimum length of a grain. In this implementation, the window associated with the grain is computed once at the time of grain extraction. It is then applied incrementally to the grain, processing it in chunks of 1024 samples at a time before outputting each buffer. This method preserves the integrity of the grain and ensures smooth transitions between subsequent segments of the grain. It takes several iterations for the function to output the entire grain, with a minimum of two iterations, but typically the number is much higher. For example, if a grain of 500 ms is extracted from an audio file with a sampling frequency of 44100 Hz, it would take exactly 22050 iterations to fully output the grain. There are no predefined assumptions about the length of the grain; it entirely depends on the user's choices and the parameters provided by the cars sensors. Given this, there is no guarantee that the grain length will be a multiple of 1024 samples; in fact, it is more likely that it wont be. The key requirement in this case is to ensure the smoothness of the output signal. Incomplete buffers must be avoided because they can cause discontinuities in the signal, resulting in audible artifacts or glitches in the audio output, as well as errors in data handling by the audio device writer. These artifacts not only degrade the listening experience but can also disrupt the stability of the application, especially in a real-time audio context like this one, where consistency is critical. As mentioned earlier, the solution was found in the use of persistent variables, which allow the storage of data between subsequent iterations. Specifically, these variables are:

- **Iteration (it)**: this variable stores the iteration counter, which is reset to 0 each time the current grain is finished and a new grain is extracted. It is used to calculate the starting and ending indices of the buffer within the grain for the current iteration, and to determine whether the grain has been fully output within that iteration. Additionally, it checks if the buffer is full by multiplying the iteration count by 1024 and comparing the result with the size of the grain.

- **Size of the grain (sz)**: it stores the size of the full grain. It is useful to determine when the grain ends and a new one needs to be extracted.

- **Grain (grain)**: this variable stores the full grain. A persistent variable is required because grains are of variable size, and dynamic allocation of such variable sizes is not possible directly within Simulink.

- **Window (window)**: it stores the full window of the length of the grain. For efficiency reasons it is computed only one time when the grain is extracted and then stored in a persistent variable.

- **Buffer (windowedBuffer)**: this persistent variable is essential for managing leftover samples from the grain that do not fit into a full buffer of 1024 samples. When a grain is not a multiple of 1024, the last buffer will contain only a portion of the grain. In such cases, after applying the window function to these remaining samples, they are stored in *windowedBuffer*. A new grain is then extracted, and the iteration counter is reset to 0. The initial samples of the new grain are appended to *windowedBuffer*, forming a complete buffer. The new grain is then processed in chunks of 1024 samples until it is finished, and this process repeats continuously.

A flag is output from this block, which is set to 1 whenever the extraction of the last buffer of the grain is complete and a new grain is needed. This flag is fed back to the *cut_and_shift* block after a unit delay, enabling the subsystem.

The window type chosen is the Tukey (tapered cosine) window. It has two input arguments: the length of the window L (always equal to the size of the grain to which the window will be applied), and the cosine fraction r. The Tukey window is a rectangular window with the first and last $r/2$ percent of the samples equal to parts of a cosine. For example, setting $r = 0.5$ produces a Tukey window where 1/2 of the entire window length consists of segments of a phase-shifted cosine with period $2r = 1$. In other words, the cosine fraction represents the attack of the grain. Varying the Cosine Factor affects the attack duration of the window, ranging from a rectangular window ($r = 0$) to a Hann window ($r = 1$). The Tukey window with varying r is illustrated in Figure 4.13.

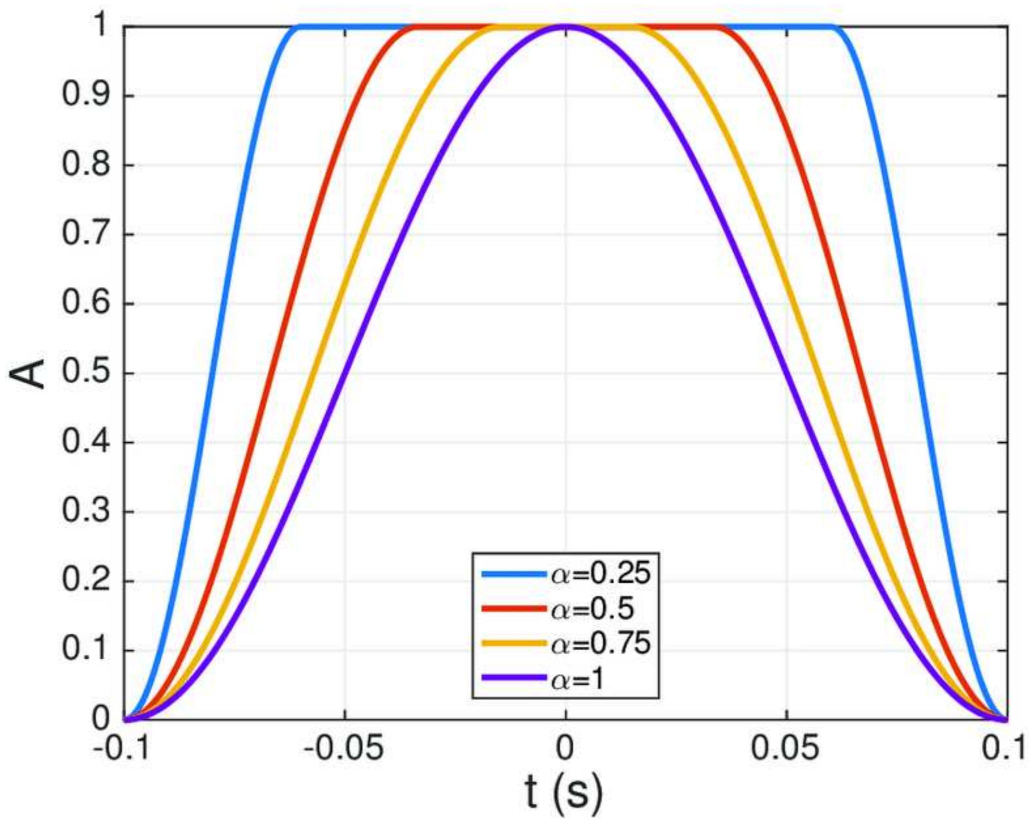

Figure 4.13: Tukey window (tapered cosine window)

At each iteration, the final output of each Thread is a buffer of 1024 samples containing a portion of the windowed grain. As explained in Section 3.3.2, these buffers must be summed to ensure the grains overlap and create a sustained sound. This summing occurs within each voice, where the outputs of the 8 Threads are combined to form a 1024-sample buffer representing the Voice's output. Three such buffers are generated, corresponding to each Voice, and these are output at the same frequency as the sampling frequency of the input audio file. Before summing the outputs of the three Voices together, additional processing is applied.

4.3 Filters and Voice Volume

To provide the user with greater flexibility in sound design, the volume of each Voice can be adjusted directly from the GUI. This adjustment is connected to a gain block at the output of each Voice, with values ranging from 0 to 1. It is important to note that the simulation requires three input audio files to function properly. However, if all three Voices are not needed, the user can simply turn down the corresponding volume knobs. This action sets the gain of those Voices to 0, effectively muting them.

Additionally, since each Voice's output is the superposition of the input audio file 8 times, users will often want to lower the volume. This is why the maximum value of the gain block is set to 1: increasing the sound volume beyond this point is typically unnecessary.

Another key feature of the AGS is the integration of filters directly within the Simulink model. These filters are well-suited to our implementation because their parameters can be adjusted during the simulation. However, some parameters, such as the IIR filter order (set to 8) and the stopband attenuation (set to 18), are fixed and can only be modified by accessing the model and changing the filter block directly. An example of the filter response is shown in Figure 4.14, illustrating a low-pass filter with a cutoff frequency of 10 kHz.

For filter type selection, a switch is incorporated into the design, as depicted by input block *c* in Figure 4.15. This switch is controlled by a constant, which the user can adjust via the GUI. Although all three types of filters process the output of each Voice simultaneously, only the output from the selected filter type is used based on the users preference. This design provides additional flexibility in sound design.

There are three types of filters available for each voice: Low Pass Filter (LPF), High Pass Filter (HPF), and Band Pass Filter (BPF). When the user selects LPF or HPF in the GUI, the only customizable parameter is the cutoff frequency, while

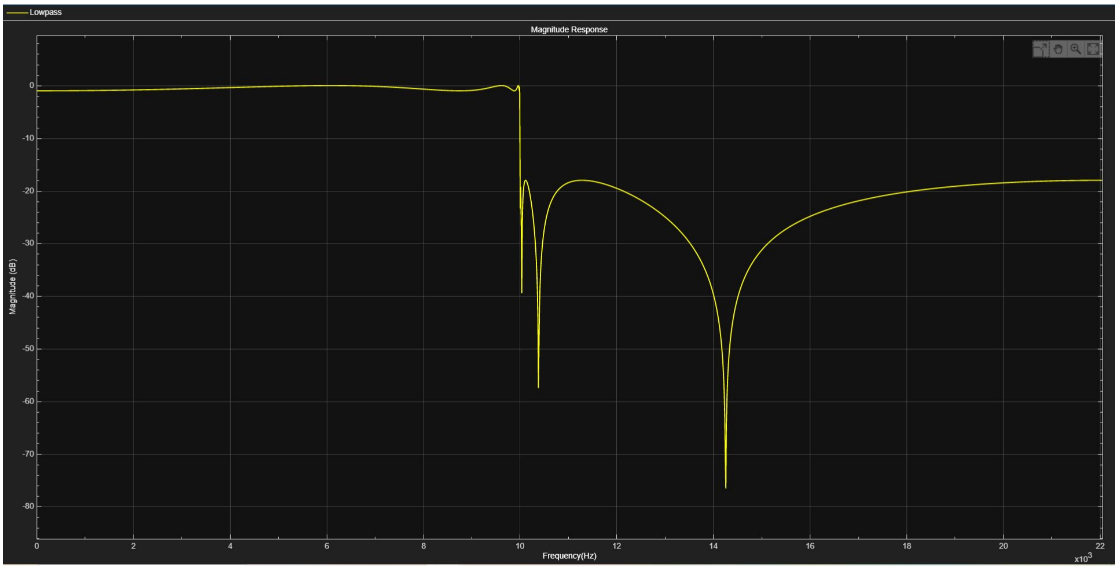

Figure 4.14: Frequency response of a low-pass filter with cutoff frequency of 10kHz

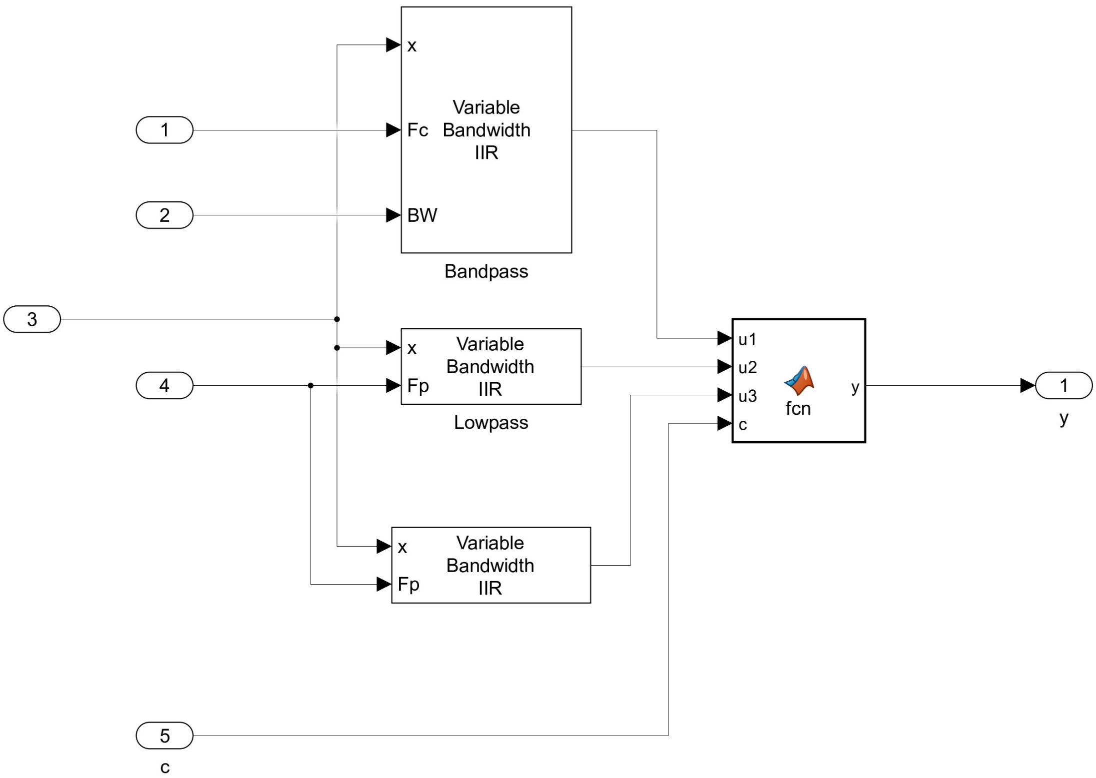

Figure 4.15: Simulink implementation of the three type of filters

the bandwidth option is automatically disabled. If BPF is selected, the cutoff frequency represents the center frequency of the filter, and the user can also adjust the bandwidth of the passband.

The filters are applied to the 1024-sample buffers output by each Voice, ensuring that the underlying structure of these buffers remains unchanged. Consequently, after filtering, there will still be three separate buffers, each containing 1024 samples, with the selected filter applied to the audio data within each buffer. These three buffers are then summed, and a gain block representing the main volume is applied.

Finally, the processed signal is ready for output through the *Audio Device Writer* block in the Simulink *Audio Toolbox*. By double-clicking this block within the model, the user can select the audio drivers and the specific output device for playback. The sample rate for audio reproduction is chosen directly within the GUI and determines the frequency at which the buffers from each Voice are output. It is crucial that the Voice output and audio reproduction are synchronized, especially in the context of real-time audio, to ensure accurate playback, without any data loss.

In this application, the input audio files are loaded and processed in mono, and the output audio also has a single channel. This is because, in this specific context, the sound is used to alert pedestrians and other road users, so there is no need for it to be in stereo as directional cues are not critical for this purpose. The primary goal is to ensure the sound is clearly heard in the surrounding environment, regardless of the listener's position relative to the vehicle.

4.4 Logical Scheme of the Processing Chain

To summarize the general structure of the model and clarify the data flow, the following schemes are provided. Figure 4.16 illustrates the car's physical model, offering a more detailed explanation of the blocks presented in Figure 4.6 and 4.8. Figure 4.17 presents a diagram of the Granular Synthesis algorithm implementation, which reflects the overall structure of the Simulink model shown in Figure 4.5. Finally, Figure 4.18 displays a schematic version of a single Thread within a Voice, referencing the blocks introduced in Figure 4.11.

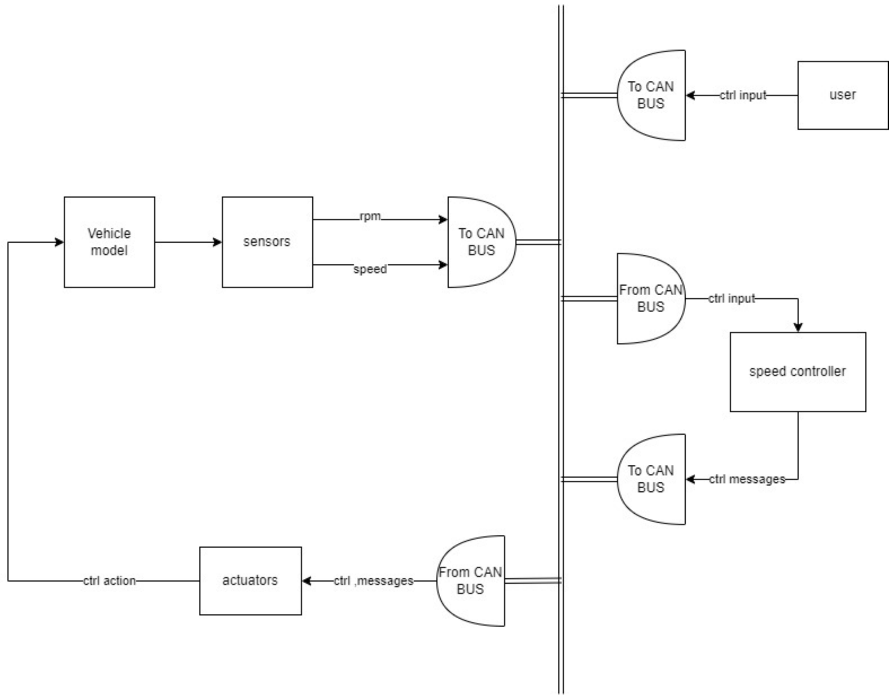

Figure 4.16: Logical scheme of the car model

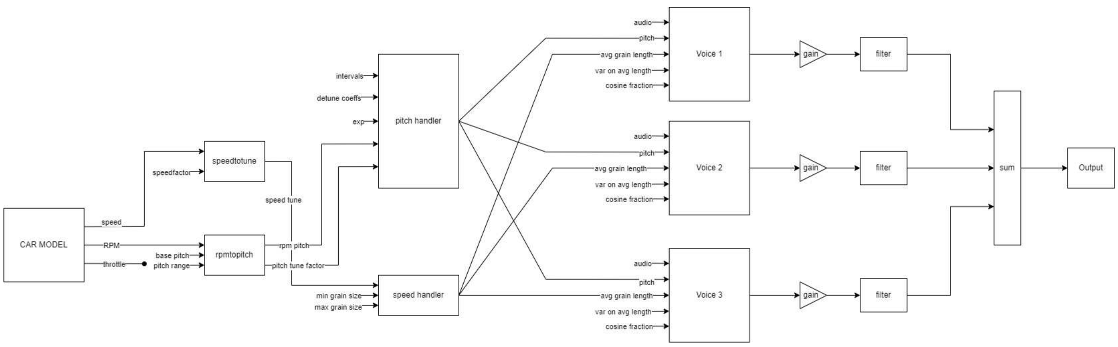

Figure 4.17: Logical scheme of the structure of the whole model

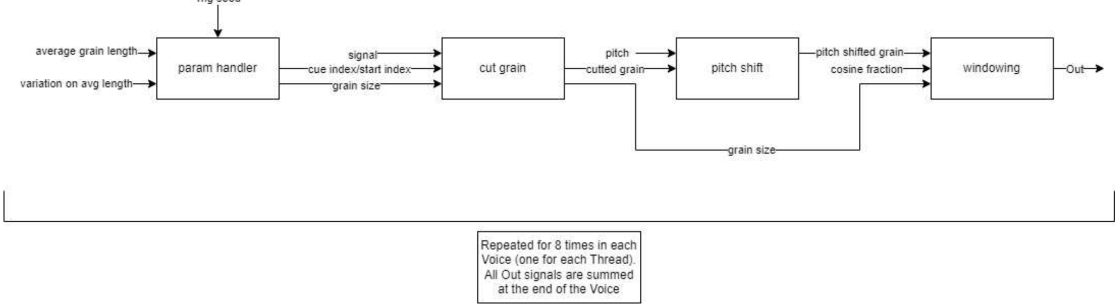

Figure 4.18: Logical scheme of of a Thread

4.5 Example of Application Workflow

Now that the structure of the model and the characteristics of the underlying algorithm have been presented, along with the various features of the application, an example of how to use the application will be demonstrated. This example will walk through the entire process, starting from loading the data and continuing through the tuning of parameters. The demonstration aims to provide a practical understanding of the workflow, illustrating how the different components and settings of the application interacts to produce the desired audio output.

Upon opening the file named *app1.mlapp*, the main interface of the application will appear. The gray status display panel, located at the bottom left, provides real-time feedback to assist users in navigating the application effectively. The initial step is to load all three input audio files. The user has two options:

- **Pressing the load button** will load the current parameters of the Simulink model into the application and also load the default audio files, whose relative path is stored in the file *default_audio.mat*. To change these default files, the user should load the desired files using the *Import* button and then select *Set as Default Audio*. The files will be resampled by default to the sampling frequency specified in the *Sampling Frequency* block located at the top right of the interface.

- **Pressing the Load Preset button** allows the user to select from a collection of default and previously saved custom presets, which will then be loaded into the model. These preset files contain all the parameter values for the synthesizer, along with the paths to the audio files and their sampling frequency.

If neither of these two buttons is pressed, the simulation cannot start (the *Start* button will remain greyed out), even if the user manually imports the input audio files. This is because pressing the load buttons triggers the automatic execution of an initialization script called *init.m*. This script not only loads and resamples the selected audio files but also sets the seeds for the random number generators and specifies the output buffer length.

After loading the input files, the user can start the simulation by pressing the *Start* button, which will allow them to hear the resulting sound through the output audio device. One of the key features of this application is its support for real-time parameter tuning, meaning that all parameters can be adjusted during the simulation, allowing the user to immediately hear the effects of these changes on the audio output. However, from within the GUI, the only operations that cannot be performed while the simulation is running are changing the input audio files and adjusting the sampling frequency, as these actions would require altering the simulation's step size.

4.5.1 Audio Recording and Additional Features

In AGS, a key feature is the capability to record audio output directly from the Simulink model, eliminating the need for external programs to capture the computer's audio. To utilize this feature, the user must connect the block labeled *Audio_Recorder* within the Simulink model, as shown in Figure 4.19. After running and stopping the simulation, the model will store the audio output in the MATLAB workspace as variables named *audio_output* and *audio_output_a*. The variable *audio_output* contains the unaltered audio output, while *audio_output_a* holds the output filtered with A-weighting.

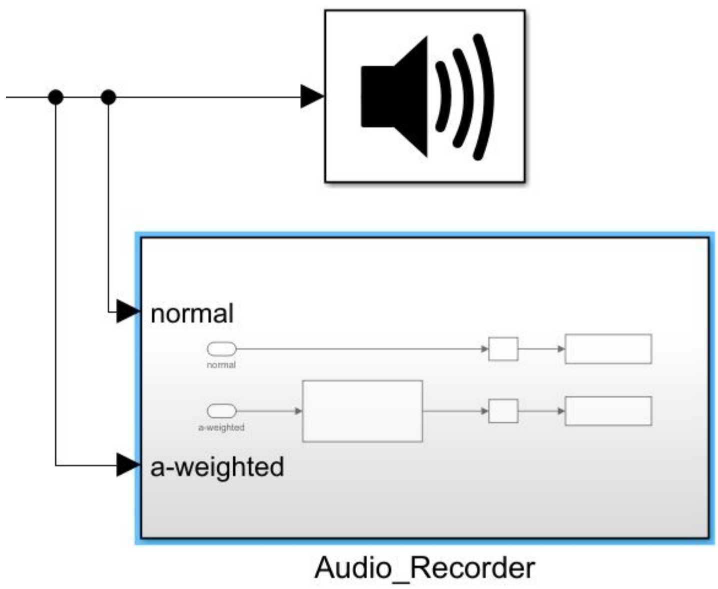

Figure 4.19: Audio Recorder

To save these outputs to the device, the user must execute the MATLAB script named *save_audio_file.m*, located in the main folder. This script will save the two variables as WAV files on the device with the same names as the variables.

An important note on audio recording: when recording is not necessary, the *Audio_Recorder* block should be disconnected from the main model. This is because, particularly during long executions, continuously saving the file can consume substantial resources and significantly slow down the process.

Having presented the principles behind the algorithm and its implementation, the next step is to evaluate the quality and detectability of the sounds it generates. This assessment will be the focus of Chapter 5, where we will examine the acoustic output in various scenarios to determine its effectiveness and suitability for its intended applications.

5 Principles and Applications

In this chapter, the principles and techniques involved in sound design using AGS are analyzed. The discussion begins by examining the theory of sound design from a psychoacoustic perspective, exploring how human perception of sound influences design decisions.

Building on this theoretical foundation, the chapter explores the practical application of these concepts within the AGS framework. It examines how various parameters of AGS can be adjusted to achieve the desired sound characteristics. This involves mapping specific features of the output sound, such as texture, dynamics, and loudness, to corresponding adjustments in grain size, pitch, envelope shape, and other relevant parameters within the AGS model.

5.1 Sound Design in Automotive

As stated in Section 2.2.3, sound selection is not only important for detection but also plays a crucial role in brand recognition and identity. The auditory signature of a vehicle, including its sound design, can significantly influence how a brand is perceived by consumers and how it distinguishes itself in a competitive market.

Sound selection is not merely about choosing a pleasant or distinctive noise; it involves a strategic approach to embedding a vehicles brand identity into its acoustic profile. For instance, the sound of a car engine or its alerting sounds can evoke specific emotions and associations related to the brands values and character. This is particularly important in the context of EV, where traditional engine noises are absent. In such cases, sound design becomes an opportunity, a critical tool for conveying attributes such as performance, luxury, or environmental responsibility.

Given these considerations, which underscore the importance of this topic, the following analysis will focus on evaluating sounds based on psychoacoustic principles.

5.1.1 Psychoacoustics

Hearing is a crucial means of obtaining information about our surroundings. It involves reacting to sound and encompasses two key components: the sound itself and the behavioral response it triggers. Psychoacoustics is the field that examines the behavioral effects of sound stimulation, which is the essence of hearing [59]. It explores not only the relationship between sound and perception but also how factors like frequency, intensity, and duration of sound shape our auditory experiences. This field offers valuable insights into how we interpret and react to various sounds, enhancing our understanding of everything from speech and music to environmental noise.

In this context, psychoacoustics provides objective metrics for assessing sound quality, helping to evaluate how well product designers' intentions align with consumers' perceptions of sound acceptability [32].

5.1.2 Capturing Sound Quality

Sound quality is closely related to the subjective emotional responses that a person has to a particular sound. It reflects the unique characteristics of the sound beyond just pitch and loudness [38], influencing how individuals experience and react to auditory stimuli. The most challenging aspect in this context is capturing the subjective essence of sound quality and translating it into objective, quantifiable metrics that can guide engineering decisions, such as tuning synthesis parameters. The main difficulty lies in the variability of individual perceptions and preferences, which complicates the creation of a standardized approach to sound quality across different brands and markets. This variability is influenced by personal experiences, cultural differences, and subjective opinions on what constitutes an "optimal" sound [28].

While there is an established measure for analyzing vehicle comfort known as Noise, Vibration and Harshness (NVH), which evaluates the auditory and tactile sensations experienced within a vehicle, this measure is not fully adequate for EV. NVH helps in identifying and addressing sources of discomfort to ensure a smoother and quieter ride. However, in the case of EV, the absence of engine noise provides more flexibility in designing synthetic sounds. Therefore, a more comprehensive approach to analyzing sound quality is needed to effectively guide the design of these sounds and adding engineering value to this process.

In the past, numerous studies and surveys have been conducted to gain a deeper understanding of drivers' and road users' perceptions of vehicle sound [50]. Many investigations focused on exploring the language used in sound evaluations and identifying the specific sound characteristics that lead to positive feedback. To capture its subjective nature, sound quality in the automotive field is described using semantic descriptors, which are adjectives that characterize the sound, such as *powerful*, *sporty*, *refined* and *harsh* [28]. Researchers use these semantic descriptors in various listening studies, where participants rate vehicle sounds based on these descriptors using a semantic differential scale. This method allows for a structured evaluation of how different sound characteristics are perceived and evaluated. The ratings are analyzed using statistical techniques, such as factor analysis, to identify underlying factors that explain variations in sound quality perceptions.

These studies reveal that sound quality in vehicles can be understood through two main dimensions [28]:

1. **Power/Strength:** Reflects the perceived intensity or impact of the sound.
2. **Comfort:** Addresses the pleasantness or comfort associated with the sound.

The qualitative adjectives **powerful** and **refined** are associated with these two aspects of sound and will be used extensively in this work. Recent research on EV sound semantics suggests that these adjectives alone are insufficient to fully capture the unique qualities of EVs [30]. Terms such as **futuristic** are also frequently used in the evaluation of electric vehicles and will be incorporated into this study.

In addition, it is essential to define quantitative acoustic features that can be mapped to these descriptive adjectives. These metrics are identified through jury listening tests on recorded engine sounds and form a strong foundation for the analysis intended in this study. They help to characterize the sound and relate these characteristics to user perceptions based on the previously specified adjectives. Analyzing prior research, as detailed in [28], identifies seven key acoustic features:

1. **Roughness (Rumble):** Intermittent and impure discomforting sensations during acceleration caused by closely coupled engine tones [56] are linked to both comfort and power aspects of sound quality. While many researchers associate higher levels of rumble with reduced vehicle comfort and refinement, others suggest that increasing rumble can enhance the sporty or powerful characteristics of a vehicle's sound. Rumble is generally identified as noise generated by resonant low frequencies, typically below 50 Hz.

2. **Linearity:** Extreme changes in sound pressure levels during vehicle acceleration can cause discomfort, degrading the comfort-related aspects of sound quality [1]. However, initial steep changes in low-frequency content can enhance the perception of sound strength and contribute to a more powerful sound quality [2, 49].

3. **Engine Firing Order:** It was found to influence the strength or powerful aspects of sound quality [53, 34].

4. **Sound Pressure Level of Low Engine Orders:** This feature influences subjective perceptions of sound quality. Some researchers found that higher sound pressure levels of low engine orders can negatively impact the comfort-related aspects of sound quality [49, 54, 5, 46]. Conversely, other researchers observed that these levels can enhance the perception of the sound's strength [29, 20].

5. **Loudness Level:** The loudness level, a fundamental psychoacoustic measure, quantifies the intensity of a sound. Research indicates that higher loudness levels generally reduce comfort, making sounds more annoying [53, 49, 5, 46, 7]. However, some studies suggest that increased loudness can enhance the perception of a sound's strength, contributing to a sense of power [57, 33, 17].

6. **Sharpness:** Sharpness measures the high-frequency content of a sound. Increased sharpness levels were found to reduce comfort and degrade

sound quality [7, 33, 49, 17, 5, 46]. Conversely, some research suggested that low sharpness levels might enhance the perception of sound strength [54].

7. **Impulsiveness:** Impulsiveness measures the irregularity in a vehicle sound, primarily perceived in the time domain. A more impulsive sound is generally perceived as powerful but can also reduce the comfort factor of the sound [47, 55, 19].

5.1.3 Mapping Sound Characteristics to Granular Synthesis Parameters

Having presented the affective adjectives, the semantic descriptors, and how they relate to the sounds, the next step is to map these sound characteristics to specific parameters within the granular synthesis (AGS) framework. This process involves translating the previously presented subjective sound qualities (*futuristic*, *powerful* and *refined*) into concrete synthesis parameters that can be manipulated to achieve the desired outcomes.

Input Audio

The initial and most critical decision in the sound design process is the selection of input audio. This audio serves as the foundational material from which all subsequent sound manipulations will be developed. The input audio fundamentally influences the final sound output, as it is the core element that the granular synthesis algorithm will process.

When choosing input samples, several factors must be considered:

1. **Source Quality:** The clarity, richness, and fidelity of the input audio directly impact the quality of the synthesized sound. High-quality recordings with minimal noise and distortion provide a better foundation for sound design. In this context, audios with higher sampling frequencies are preferable; however, the key is to avoid resampling when possible, particularly when there is a significant difference in frequencies.

2. **Sound Characteristics:** The spectrum of the input audio is the most critical factor in shaping the final output of the algorithm. Intrinsic sound

characteristicssuch as timbre, pitch, and dynamic rangeplay a fundamental role in determining the overall quality and impact of the synthesized sound. For instance, as discussed in Section 5.1.2, if the goal is to produce a sound perceived as aggressive, input audio with high loudness, rich high-frequency content, and a wide dynamic range (including sharp transients or small impulses) is preferred. In contrast, to generate a sound perceived as relaxing, input audio with lower loudness level, limited sharpness, and a more restricted dynamic range is preferred. A good example in this context could be the sine wave, which is characterized by its pure tone and smooth, consistent waveform, making it well-suited for producing calming and soothing sounds.

3. **Flexibility:** The input audio should be versatile enough to allow for extensive manipulation. A sound that can be easily adjusted in terms of pitch, duration, and filtering will provide more creative possibilities during the granular synthesis process.

4. **Fit with Desired Outcome:** The input audio should align with the desired acoustic profile of the final sound. For automotive applications, this means selecting sounds that match the vehicles design goals and branding. For example, if the aim is to emulate the sound of a real V8 engine, the input audio could be a field recording of an actual V8 engine.

Pitch

In AGS, pitch adjustment is essential for simulating the sensations of acceleration and braking in a vehicle. By altering the pitch, the synthesized sound can mirror the vehicle's dynamic changes in speed and RPM.

The degree of pitch shifting during acceleration significantly influences users' perceptions of speed. Larger pitch intervals can create the impression of faster acceleration and higher speeds, thereby enhancing the driving experience. Additionally, pitch adjustments can alter the harmonic content (sharpness) of the sound during acceleration, enriching the higher frequencies and contributing to a more powerful and dynamic auditory sensation. Conversely, by limiting the extent of pitch shift, the sound can maintain its original texture and timbre, providing a more consistent auditory experience that emphasizes refinement and control.

Grain Duration

Grain duration is a crucial parameter in granular synthesis, influencing the continuity, texture, and dynamic range of the synthesized sound. Short grain durations tend to create a more staccato or glitchy effect [43], resulting in a fragmented auditory texture that can introduce impulsive elements and enhance the perceived power of the sound. In contrast, longer grain durations produce a smoother and more continuous sound, contributing to greater fluidity and cohesiveness, making the audio output more refined.

Additionally, the selection of grain duration plays a significant role in determining how closely the synthesized sound replicates or diverges from the original input audio. Longer grains enable the preservation of more detailed aspects of the original sound, maintaining a closer resemblance to the source material. In contrast, shorter grains can lead to a more fragmented outcome that may deviate significantly from the original input audio [43].

Grain Envelope

In AGS, the user can adjust the Cosine Factor (α) to modify the attack and release characteristics of the window applied to each grain, as illustrated in Figure 4.13. By altering the shape of the window envelope, the frequency content, texture, and timbre of the final audio can be adjusted. Shorter attack times produce transients, which manifest as discontinuities in the final audio due to abrupt changes in volume or timbre between successive grains. These transients are perceived as very short bursts of sound or impulses and are more pronounced when the input audio has a wide dynamic range or evolving timbre. Such transients enhance the impulsiveness of the final sound, creating a sense of power. Conversely, increasing the Cosine Factor smooths the transition between successive grains, reducing or eliminating transients and resulting in a more cohesive sound, which is generally perceived as more relaxing. Longer attacks generally result in more smooth fluctuations in the volume of the final output.

Filtering

Filters provide additional flexibility in shaping the frequency content of the audio output. In AGS, filters are particularly useful as they are applied at the end of each processing chain for every voice. This placement allows for precise control over their effects on the final audio. By adjusting filters, users can fine-tune the spectral characteristics of the output, such as attenuating unwanted frequencies, emphasizing desired ones, or shaping the overall tonal balance. For instance, a HPF can be used to remove higher frequencies, reducing the sharpness of an overly aggressive sound. Conversely, a LPF can be employed to cut lower frequencies, reducing roughness to obtain a more refined sound.

Volume

In the context of software implementation, discussing the absolute loudness of the output audio is not particularly relevant, as this is typically addressed during the hardware setup. What truly matters, especially concerning regulatory compliance and noise pollution, is the SPL emitted by the car's speaker. While increasing loudness can enhance the perceived power of the sound, making it feel more powerful and less refined, this adjustment is generally handled at the hardware level. Within the software, the focus is on the relative loudness between different voices. Balancing these volumes can effectively alter the frequency content of the output audio. For instance, a user might introduce subtle textures by incorporating various types of noise at low intensities, adding depth and realism to the overall sound.

6 Survey

To assess the versatility of the AGS, an anonymous online survey was conducted. The primary objective of this survey was to demonstrate the synthesizer's ability to generate a diverse range of sounds, each capable of evoking distinct emotional responses from listeners.The survey was constructed using the PsyToolkit platform [51, 52], and efforts were made to recruit as many participants as possible, with particular emphasis on ensuring a diverse demographic background among them.

6.1 Methodology

Participants were asked to evaluate 9 different sounds based on three affective adjectives, as described in Section 5.1.2: **aggressive**, which relates to the strength and intensity of the sound; **refined**, which reflects the comfort and smoothness of the sound; and **futuristic**, which refers to the sound's innovative or forward-thinking qualities. The sounds were produced following the guidelines outlined in Section 5.1.3, varying the synthesizer parameters and input audio to create sounds with different characteristics:

- **Sound 1** is made up of three sawtooth waves, each at a different octave. The Cosine Factor is set to a low value, giving each grain a short attack. During the acceleration, each Voice is pitched up by 12 semitones, resulting in a sound with rich harmonic content, especially in the high frequencies. Grain duration spans from 1s to 700ms.

- **Sound 2** is created using distorted sawtooth waves combined with white noise. This combination produces an output characterized by sharp, high-frequency content and a prominent low end. The distortion significantly enhances the harmonic content, especially in the higher frequencies, adding brightness and intensity to the sound. Cosine Factor is set to a low value, ensuring a shorter attack to each grain. Grain duration varies from 600 milliseconds to 200 milliseconds.

- **Sound 3** is made up of three sine waves at different octaves and with a white noise. Each Voice has the Cosine Factor set to 1, ensuring the longest possible attack to each grain. In addition, a LPF has been applied to the white noise to filter out frequencies above 1000 Hertz. Grain duration varies from 1 second to 800 milliseconds.

- **Sound 4** is similar to Sound 3, but the pure tones are pitched down by 7 semitones, and the grains have a shorter attack, highlighting the transients and creating a more percussive texture. Additionally, a filtered sawtooth wave has been added an octave below to provide a stronger low end to the sound. Grain duration is set to vary from 700 milliseconds to 300 milliseconds.

- **Sound 5** is generated using three sawtooth waves. The first is filtered and pitched down an octave to fill the lower part of the spectrum, while the other two are pitched to form a minor chord. As the car accelerates, these two signals are detuned to create an inharmonic sound. The grains have a long attack time, while the grain duration varies from 1 second to 300 milliseconds.

- **Sound 6** is made using as input a filtered sawtooth and a sine wave that sweeps across the entire spectrum from left to right. The cosine factor is set to 0, so the window applied to each grain is rectangular. As the car accelerates, the sound is pitched up an octave, and the grain duration varies from 1 second to 300 milliseconds. The ascending pure tone causes rapid changes in the sounds spectrum, giving it characteristics that resemble a siren.

- **Sound 7** is created using dark-toned sounds layered with sine waves at lower volumes. The grains have short attacks, but a LPF is used to phase out the higher frequencies introduced by the transients. Grain duration varies from 600 milliseconds to 400 milliseconds. The result is a hum sound that resembles the one produced by electric engines.

- **Sound 8** is produced layering filtered swatooth waves at lower octaves with a sample of a recorded church bell. The grains have longer attack in order to leave out the initial transient of the bell. Grains duration varies from 1 second to 300 milliseconds. The resulting output sound has a dark tone and a prominent low end.

- **Sound 9** is generated using unfiltered, distorted samples layered with white noise. One of the input samples is filtered using a HPF. This makes the sound audible only as the car reaches a certain speed, with the sound pitching up to reach and surpass the cutoff frequencies. The grains have a short attack, and their duration varies between 600 milliseconds and 300 milliseconds. The resulting sound is rich in harmonic content and features pronounced transients.

The survey was organized as follows: participants were initially asked three demographic questions, which were considered relevant for the analysis. The first question focused on the respondent's age, as individuals from different age groups may have distinct life experiences that can influence how they perceive and respond to auditory stimuli. The second question investigated whether the participant holds a driver's license. This is important because it directly relates to their driving experience, particularly of ICEV.

The third question asked how often the respondent drives an EV. This was included to understand their familiarity with EV, as those who drive them more frequently may have different perceptions of the synthesized automotive sounds. Including participants' driving habits and experiences into the analysis provides valuable context for interpreting the feedback, enabling the identification of potential patterns in how various groups, based on factors like age or EV-driving experience, respond to the sounds produced by AGS.

Following this, participants were asked to evaluate the sounds one at a time. For each sound, a slider ranging from 0 to 100 and a step size of 1 was provided for each of the three adjectives: aggressive, refined, and futuristic. Participants were required to rate how strongly each adjective applied to the sound before proceeding to the next one. Importantly, respondents could only listen to the next sound after submitting their evaluation for the previous one. This was to ensure that participants gave their full attention to each sound and provided individual feedback without being influenced by upcoming samples.

6.1.1 Data Collection

Data were collected from 95 participants, with a mean age of 35 years, ranging from a minimum of 17 to a maximum of 91. Of these participants, 89 possess a driving license (93.68%), while 6 do not (6.32%).

Out of them, 72 never drove an electric vehicle (EV) (75.79%), 14 have driven it a few times (14.74%), 1 drives it rarely (three times a month or less) (1.05%), 3 drive it frequently (once a day) (3.16%), and 5 drive it daily (5.26%). The following table displays the number and percentage of individuals who responded to whether they had driven an EV and, if so, how often.

In Table 6.1 are presented the mean and standard deviation of participants' ratings for each of the nine sounds produced by the granular synthesizer. Results are in the range [0-1] as the data were normalized to fit within this range. The highest mean scores for each characteristic are highlighted in yellow, while the lowest scores are marked in red.

6.2 Results

The analysis began with a preliminary examination of the data, including the calculation of the mean and standard deviation for each sound across the three analyzed characteristics. These statistical metrics provide insight into the central tendency and dispersion of participants' ratings. The mean value indicates the average rating given by participants, reflecting the general perception of each characteristic, while the standard deviation quantifies the variation in participants' responses, offering insights into the consistency of these perceptions. A smaller standard deviation suggests that participants had similar ratings for that characteristic, while a larger standard deviation indicates more diverse opinions.

Following this, a one-way Analysis of Variance (ANOVA) was performed to assess whether there were statistically significant differences in mean ratings across the nine sounds for each characteristic. ANOVA is particularly useful in this context, as it allows for the comparison of multiple groups simultaneously, thereby identifying any overall differences among the sounds without inflating the risk of Type I error, which can occur when conducting multiple t-tests.

Table 6.1: Mean and Standard Deviation for Each Sound by Characteristic

Sound	Characteristic	Mean ± SD
Sound 1	Aggressive	0.61 ± 0.27
	Refined	0.18 ± 0.20
	Futuristic	0.55 ± 0.32
Sound 2	Aggressive	0.57 ± 0.26
	Refined	0.22 ± 0.23
	Futuristic	0.45 ± 0.28
Sound 3	Aggressive	0.28 ± 0.25
	Refined	0.50 ± 0.29
	Futuristic	0.64 ± 0.27
Sound 4	Aggressive	0.45 ± 0.26
	Refined	0.31 ± 0.21
	Futuristic	0.49 ± 0.26
Sound 5	Aggressive	0.44 ± 0.29
	Refined	0.26 ± 0.23
	Futuristic	0.55 ± 0.27
Sound 6	Aggressive	0.59 ± 0.31
	Refined	0.13 ± 0.18
	Futuristic	0.71 ± 0.29
Sound 7	Aggressive	0.22 ± 0.19
	Refined	0.57 ± 0.26
	Futuristic	0.65 ± 0.26
Sound 8	Aggressive	0.35 ± 0.24
	Refined	0.30 ± 0.21
	Futuristic	0.50 ± 0.23
Sound 9	Aggressive	0.57 ± 0.30
	Refined	0.22 ± 0.21
	Futuristic	0.47 ± 0.28

ANOVA requires the data to be normally distributed, a condition that is often satisfied by the Central Limit Theorem. This theorem states that, given a sufficiently large sample size, the distribution of the sample means will approach a normal distribution, regardless of the shape of the population distribution from which the samples are drawn. In this case, the number of observations is 95, which exceeds the typically recommended threshold of 30 or more for ensuring normality through the Central Limit Theorem. Another requirement for ANOVA is homoscedasticity, or homogeneity of variance, across the different analyzed groups. In this context, the variances are relatively high and not equal. However, it is important to note that ANOVA is generally robust to violations of the homoscedasticity assumption, particularly when the sample sizes are equal across all groups [3], as is the case here. This robustness means that ANOVA can still produce reliable results even when the assumption of equal variances is not fully met. A popular rule of thumb used in one-way ANOVA to verify the requirement of equality of variances is that the largest sample standard deviation should not be larger than two times the smallest sample standard deviation [39], which is a condition that is met in this case. The final requirement is the independence of data, which is crucial for the validity of ANOVA results. In this context, independence is achieved because the observations are collected from distinct participants or samples.

ANOVA results are presented in Figures 6.1, 6.2, and 6.3 for the three adjectives: **aggressive**, **refined**, and **futuristic**.

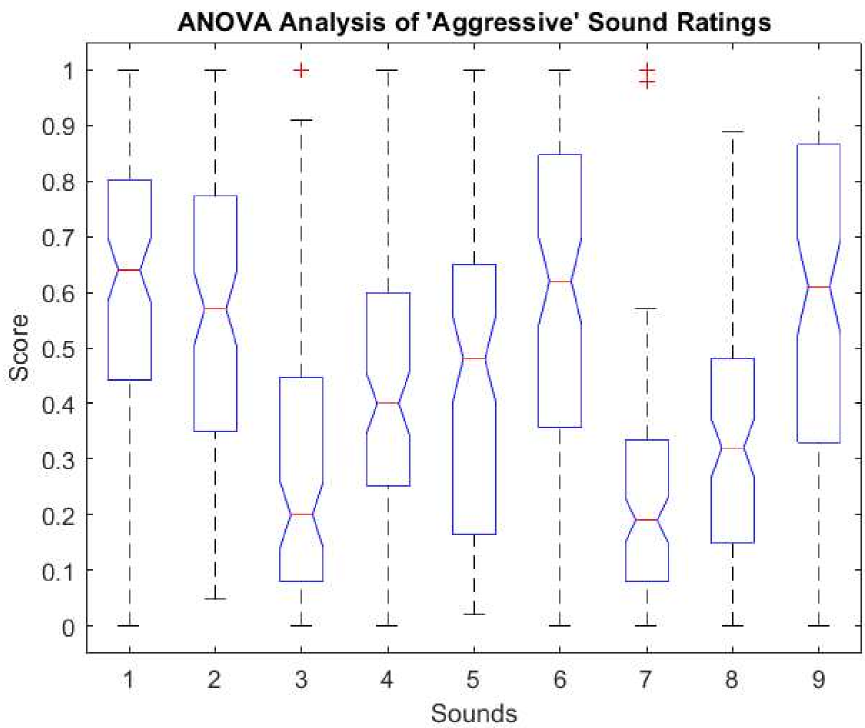

Figure 6.1: ANOVA results for 'Aggressive' feature

In each graph, the central red mark indicates the median, while the bottom and top edges of the box represent the 25th and 75th percentiles, respectively. The whiskers extend to the most extreme data points that are not considered outliers, with outliers plotted individually using the red '+' marker symbol.

The F-values are calculated as the ratio of the variance between the groups to the variance within the groups. This ratio indicates how much of the total

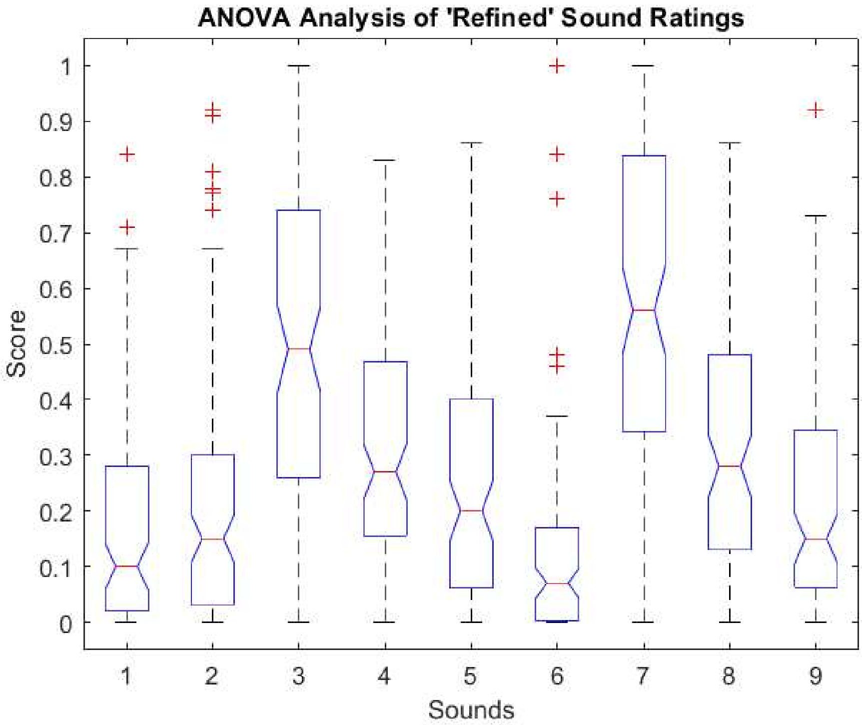

Figure 6.2: ANOVA results for 'Refined' feature

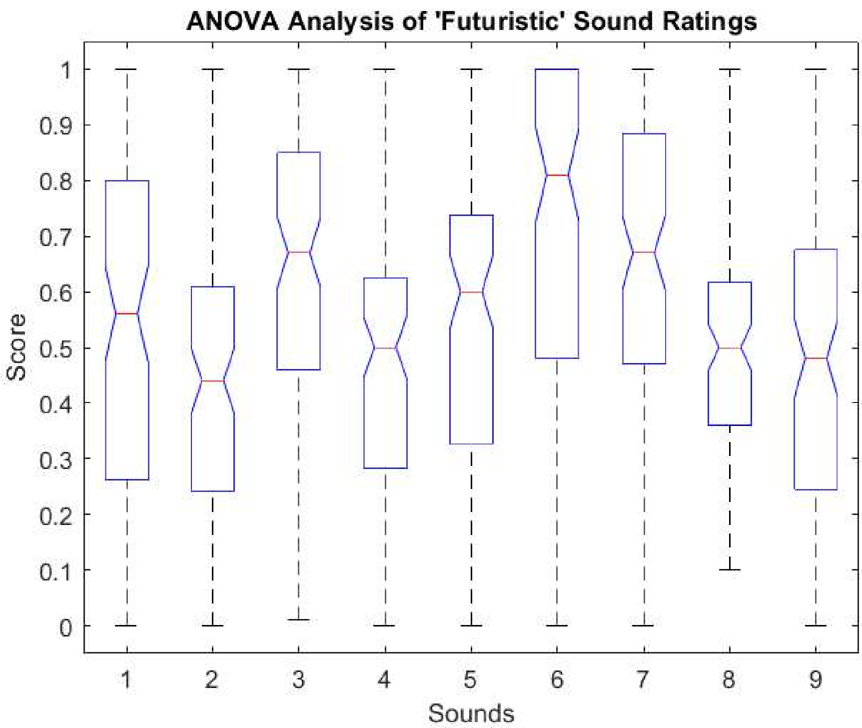

Figure 6.3: ANOVA results for 'Futuristic' feature

variation in the data can be attributed to the differences between the group means.

To obtain the p-value, the calculated F-value is compared to an F-distribution with the appropriate degrees of freedom. Results of this analysis is showed in Table 6.2.

Table 6.2: ANOVA Results: F-values and p-values for characteristics

Characteristic	F-value	Prob>F
Aggressive	27.52	3.76e-38
Refined	37.68	2.45e-51
Futuristic	10.12	1.51e-13

In this context, all three p-values are very low, indicating that there are statistically significant differences in sound ratings for the three characteristics. Specifically, the low p-values suggest that the null hypothesis, which is that there are no differences between the group means, can be rejected. This implies that the participants' ratings for at least one of the sounds significantly differ from the others in each characteristic.

Further analysis is required to determine which of the sounds differ from one another. To achieve this, a post-hoc test was conducted, specifically Tukey's Honestly Significant Difference (HSD) test. This test compares the means of the scores across the nine sounds for each characteristic. Results are showed in Figures 6.4, 6.5 and 6.6.

In these graphs, the mean of the different sound scores are represented by a circle, while the confidence interval is represented by a line extending from it. Two means are considered significantly different if their intervals are disjoint, while they are not significantly different if their intervals overlap.

6.3 Discussion

Looking at the data in Table 6.1, one immediate observation is the relatively high standard deviation across all characteristics, indicating considerable variation in participants' ratings for each sound. This suggests that there was a lack of consensus in how participants perceived aggressiveness, refinement, and futurism. Several factors likely contributed to this variability.

One key factor is the survey design. Participants used a slider with a step size of 1, ranging from 0 to 100, without seeing the exact numerical value. While

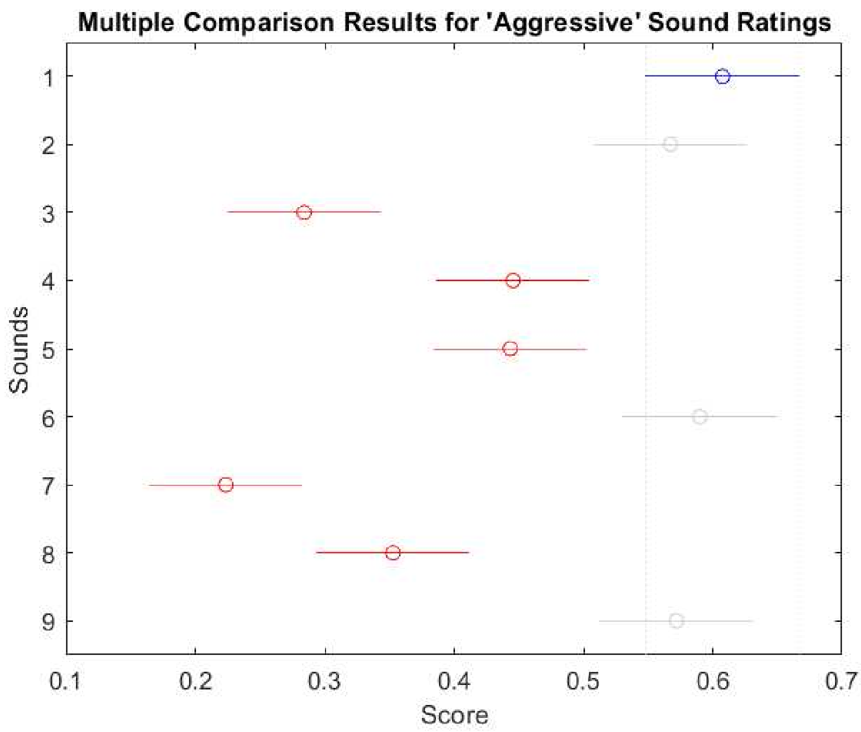

Figure 6.4: Tukey's HSD test results for 'Aggressive' feature

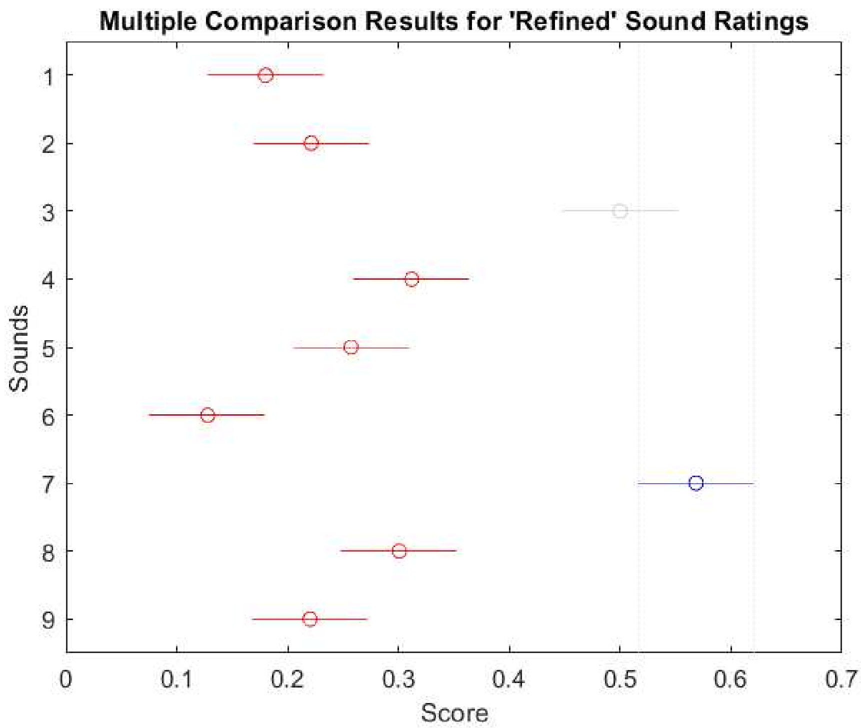

Figure 6.5: Tukey's HSD test results for 'Refined' feature

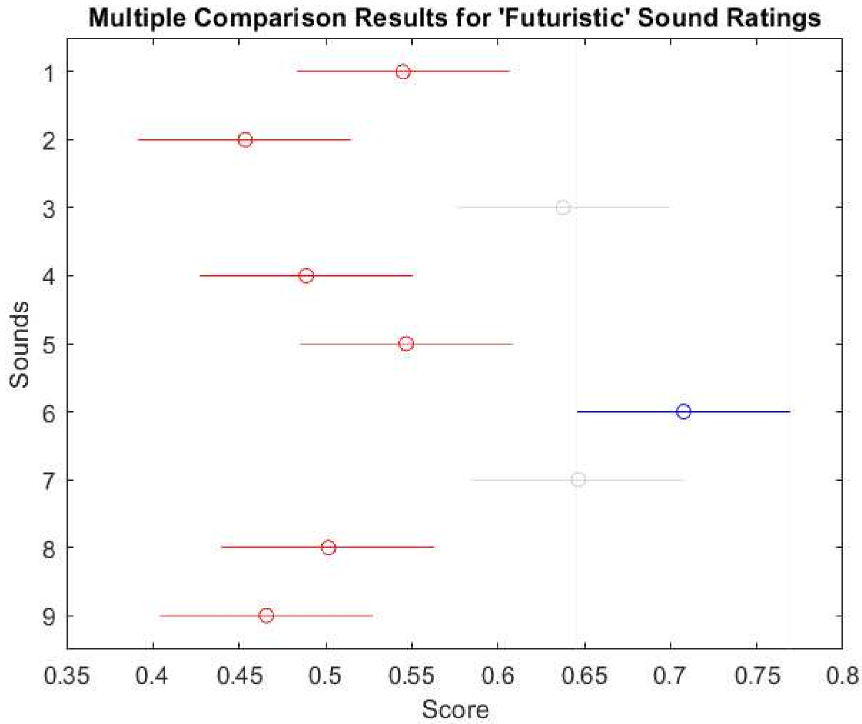

Figure 6.6: Tukey's HSD test results for 'Futuristic' feature

this design encourages quick, instinctive responses and reduces bias, it inherently increases variance compared to a Likert scale, which restricts responses to fewer fixed points. Although the Likert scale reduces variance, it also limits precision, whereas the continuous slider offers more flexibility but results in greater variability.

Moreover, the hidden slider prevented participants from overthinking or aiming for specific numbers, promoting more intuitive responses. However, without numerical feedback, participants may have interpreted the scale differently, leading to more variation. Lastly, subjective perceptions of abstract qualities like "aggressiveness," "refinement," and "futuristic" likely varied among participants, influenced by their individual experiences and cultural backgrounds, further contributing to the datas variability.

Analyzing the results in Table 6.1, a clear pattern emerges: sounds rated as more aggressive are generally perceived as less refined. This inverse relationship supports the psychoacoustic concepts discussed in Section 5.1.1, where qualities such as sharpness and intensity enhance perceived aggressiveness but diminish perceptions of refinement. The data align with the theory that specific acoustic properties evoke opposite emotional responses, indicating that aggressive sounds inherently lack the characteristics associated with refinement. This pattern is particularly evident in Sounds 1, 2, 6, 7, and 9.

In addition, the sounds closely align with the psychoacoustic principles that guided their design. Comparing the results in Table 6.1 with the design concepts discussed earlier in this chapter shows that following these principles can create sounds that evoke specific emotions.

For example, sharper and more intense sounds typically trigger feelings of aggression, while softer and smoother sounds are associated with refinement or calmness (Section 5.1.1). This connection between sound design and emotional response highlights the significance of psychoacoustic principles in crafting effective auditory experiences. The results confirm that sound design is both an artistic and a scientific process, where understanding how people perceive sound can lead to impactful emotional effects.

6.3.1 ANOVA Results Interpretation

Analyzing the ANOVA results presented in Figures 6.1, 6.2, and 6.3 allows for a deeper examination of the high variability in the data, previously indicated by the elevated standard deviations. Overall, there is greater consensus among participants when identifying sounds as less refined, which suggests a clearer perception of what constitutes a lack of refinement. However, as shown in Figure 6.2, some outliers exist, indicating that while many participants agree on certain sounds being less refined, a few individuals have different opinions. Additionally, as shown in Figure 6.3, the results for the 'futuristic' characteristic do not reveal a clear trend, with many medians clustered around the middle of the scale. This lack of consensus highlights the ambiguity of the term, as many participants requested further clarification on what specific characteristics define a sound as 'futuristic.'

6.3.2 Multiple Comparison Test Results Interpretation

As stated in Section 6.2, the p-values for all three characteristics are very low, indicating that at least one of the sounds is perceived differently from the others. To determine which specific sounds differ in perception, the results of the post-hoc tests presented in Figures 6.4, 6.5, and 6.6 will be analyzed.

Starting with 'aggressiveness' (Figure 6.4), Sounds 1, 2, 6, and 9 are statistically perceived as significantly different from all other sounds, being regarded as more powerful. These sounds share similar characteristics, including prominent high-frequency content and volume variations, which help justify the results. In contrast, Sound 7 is perceived as the least aggressive, with its confidence interval only overlapping with that of Sound 3. This indicates a statistical difference in aggressiveness perception between Sound 7 and all other sounds, except for Sound 3.

Regarding the 'refined' characteristic (Figure 6.5), there is a clearer separation: Sounds 3 and 7, while not statistically different from each other, are perceived as more refined compared to the other sounds. This result is supported by the characteristics of these two sounds, which are smooth and have a limited frequency spectrum. Sound 6 is perceived as the least relaxing, which is understandable given its characteristics that resemble those of a siren, with tones rapidly ascending in pitch. Sirens are designed to be attention-grabbing rather than refined, as their primary purpose is to be easily detectable in urgent situations.

Lastly, the 'futuristic' characteristic (Figure 6.6) highlights less clear trends. Sound 6 is perceived as the most futuristic, showing a statistical difference from all other sounds except for Sounds 3 and 7. As mentioned earlier, Sound 6 has a siren-like quality, which is not typically associated with cars but is more commonly found in emergency vehicles. This may evoke a sense of moving away from traditional car sounds toward a more innovative approach. In addition, sounds perceived as more refined tend to score higher on the futuristic characteristic. This could be attributed to the fact that many sounds with high-frequency content (such as white noise, unfiltered, or distorted tones) often evoke the auditory signature of a traditional ICEV, particularly that of sports cars. As a result, these familiar acoustic features may detract from the overall perception of futurism. In contrast, sounds that are smoother, cleaner, and less similar to conventional vehicles can enhance the impression of innovation and modernity, aligning more closely with contemporary ideals of futuristic design.

6.4 Limitations

It is important to state that this survey does not hold any scientific relevance and cannot replace a scientific experiment by any means. It is to be intended as a preliminary exploration aimed at gathering subjective impressions from participants, providing initial insights into the emotional impact of the sounds generated by the AGS. The results are intended to inform future research and development, but they should not be interpreted as definitive or conclusive findings.

Online surveys can be useful because a large number of people can be reached, allowing for diverse responses across different demographics. This helps gather a wide range of perspectives and experiences, making it easier to identify general trends or patterns in the data. Additionally, online surveys are more convenient for participants, enabling them to respond at their own pace and from any location, which can lead to higher participation rates compared to traditional methods. However, it is important to acknowledge that online surveys cannot fully replicate the controlled conditions of scientific experiments. For example, one participant may be wearing headphones while taking the survey, while another may listen to the sounds using phone speakers. These variations in listening conditions can influence how participants perceive the sounds, introducing variables that are outside the researcher's control.

Another challenge lies in the selection of sounds. In this context, there is no standardized approach to sound design, particularly when dealing with a large number of possible parameter combinations. This makes it difficult to establish a consistent framework for sound selection. In this work, the approach taken was to choose a diverse set of sounds with varying characteristics. These sounds were selected based on principles from the existing literature (presented in Section 5.1.2), which suggest that they should be perceived by the listener with different levels of aggressiveness and refinement. While this method provides a useful starting point, it is inherently subjective and lacks the precision that standardized methods would offer in a more controlled experimental setting.

7 Conclusions and Future Works

This thesis focused on the development of AGS, an application designed in Simulink to address the challenges raised by the quiet operation of EVs. The synthesizer offers an innovative solution for generating customizable engine sounds in real-time, leveraging granular synthesis as an alternative to the conventional wavetable synthesis. The primary objective was to enhance pedestrian safety while providing manufacturers with the flexibility to design unique sound profiles that reflect their brand identity.

A key feature of the AGS is its ability to adjust various granular synthesis parameters in real time, responding dynamically to the vehicle's operational state and speed. The wide range of tunable parameters presented in Chapter 4 allows sound designers to craft organic and specific sounds that meet diverse vehicle requirements, enhancing both functionality and the listening experience. Through the GUI, engineers can easily interact with the Simulink model, fine-tune its parameters, and listen to the changes in real time. This intuitive interface simplifies the sound design process, allowing for efficient adjustments and immediate auditory feedback.

The thesis also examined how variations in these parameters influence the resulting sounds and their emotional impact. An in-depth study presented in Chapter 5 outlined the principles of psychoacoustics and the corresponding sound design decisions that must be made to produce specific auditory experiences. This analysis highlighted the relationship between parameter adjustments and the emotional responses evoked by different sound profiles, providing valuable insights into effective sound design for EV.

To further validate the effectiveness of these sound designs, a survey was conducted, as detailed in Chapter 6. Participants rated a selection of nine sounds generated by the AGS based on three affective adjectives: aggressive, refined, and futuristic. This evaluation aimed to understand the emotional responses triggered by the various sounds, assessing the versatility of the synthesizer in creating a range of auditory experiences.

After analyzing the results using ANOVA and post-hoc tests, statistically significant differences were revealed between some of the sounds, demonstrating the capability of the AGS to produce sounds that convey a spectrum of perceived emotions. This feedback not only underscores the application's potential to enhance pedestrian safety but also highlights its ability to align with manufacturers branding efforts in the automotive industry.

7.1 Future Works

While this thesis has made significant steps in developing the Automotive Granular Synthesizer (AGS), there are several opportunities for future research and practical application.

Firstly, as noted in Section 6.4, the limitations of the survey necessitate further scientific experiments to validate the findings and provide a more robust assessment of the emotional impact of the generated sounds. This will enhance the understanding of how these sounds affect different listener demographics and contexts.

Secondly, the practical implementation of the AGS into a real-world system presents an exciting opportunity. By extracting C code directly from the Simulink model, the synthesizer can be integrated into microcontroller systems for use in actual electric and hybrid vehicles. This integration would allow for real-time sound generation based on vehicle parameters, significantly improving pedestrian safety.

Additionally, there is potential to expand the customization options within the AGS. Future enhancements could include features such as volume variations that respond dynamically to the vehicle's speed, allowing for a more immersive auditory experience. Implementing options to modify sound characteristics based on other operational states, such as acceleration or deceleration, could further refine the sound design.

Lastly, the development of user-friendly interfaces for tuning additional parameters, like the ability to leave out specific sound threads, could enhance the versatility of the synthesizer and better accommodate the diverse needs of sound designers and manufacturers in the automotive industry.

References

[1] H. Aoki, M. Ishihama, and A. Kinoshita. "Effects of Power Plant Vibration on Sound Quality in the Passenger Compartment During Acceleration". In: *SAE Technical Paper*. 870955. SAE International, 1987. DOI: 10.4271/870955.

[2] R. Bisping. "Emotional Effect of Car Interior Sounds: Pleasantness and Power and Their Relation to Acoustic Key Features". In: *SAE Technical Paper*. 951284. SAE International, 1995. DOI: 10.4271/951284.

[3] M. Blanca et al. "Effect of variance ratio on ANOVA robustness: Might 1.5 be the limit?" In: *Behavior Research* 50 (2018), pp. 937–962. DOI: 10.3758/s13428-017-0918-2.

[4] BMW. *Hans Zimmer: Individual Drive Sounds as Identity for Electric Vehicles*. 2021. URL: https://www.bmw.com/en/magazine/innovation/hans-zimmer-individual-drive-sounds-as-identity-for-electric-vehicles.html.

[5] F. Brandl and W. Biermayer. "A New Tool for the Onboard Objective Assessment of Vehicle Interior Noise Quality". In: *SAE Technical Paper*. 1999-01-1695. SAE International, 1999. DOI: 10.4271/1999-01-1695.

[6] CNN Business. *Electric Cars: Audi, Volkswagen, Tesla and the race for an electric future*. 2019. URL: https://edition.cnn.com/interactive/2019/08/business/electric-cars-audi-volkswagen-tesla/.

[7] G. D. Callow and R. Hedges. *The Subjective Response of Occupants to the Noise Inside Vehicles*. Research Report 1979/1. Nuneaton, Warwickshire, UK: MIRA Ltd, 1979.

[8] C. C. Chan. "The state of the art of electric and hybrid vehicles". In: *Proceedings of the IEEE* 90.2 (2002), pp. 247–275.

[9] C. C. Chan. "The State of the Art of Electric, Hybrid, and Fuel Cell Vehicles". In: *Proceedings of the IEEE* 95.4 (2007), pp. 704–718.

[10] Ford Motor Company. *Ford Europe Goes All-In On EVs*. 2021. URL: https://corporate.ford.com/articles/electrification/ford-europe-goes-all-in-on-evs.html.

[11] Hedges Company. *How Many Cars Are There in the World?* 2021. URL: https://hedgescompany.com/blog/2021/06/how-many-cars-are-there-in-the-world/.

[12] H. Fastl and E. Zwicker. *Psychoacoustics: Facts and Models*. 3rd. Berlin Heidelberg: Springer, 2007.

[13] L. Garay-Vega et al. *Quieter Cars and the Safety of Blind Pedestrians: Phase I*. Tech. rep. Document number: DOT HS 811 304. Washington, DC: NHTSA, Apr. 2010.

[14] Klaus Genuit and André Fiebig. *Sound Design of Electric Vehicles - Challenges and Risks*. HEAD acoustics GmbH. Germany, 2023.

[15] Renault Group. *The Story Behind the Sound Design with Jean-Michel Jarre: The Documentary*. 2021. URL: https://www.renaultgroup.com/en/news-on-air/top-stories-2/the-story-behind-the-sound-design-with-jean-michel-jarre-the-documentary/.

[16] R. Hanna. *Incidence of Pedestrian and Bicyclist Crashes by Hybrid Electric Passenger Vehicles*. Tech. rep. DOT HS 811 204. Washington, DC: NHTSA, Sept. 2009.

[17] T. Hashimoto. "Improvement of Sound Quality of Vehicle Noise". In: *JSAE* 49.6 (1995), pp. 65–70.

[18] K. G. Høyer. "The history of alternative fuels in transportation: The case of electric and hybrid cars". In: *Utilities Policy* 16.2 (2008), pp. 63–71.

[19] Ronacher A. Hussain M. Golles J. and Schiffbänker H. *Statistical Evaluation of an Annoyance Index for Engine Noise Recordings*. SAE Technical Paper 911080. SAE International, 1991.

[20] S. M. Hutchins et al. "Noise, Vibration, and Harshness from the Customer's Point of View". In: *Proceedings of the FISITA XXIV Congress, London*. IMechE paper C389/049. Institution of Mechanical Engineers. London: Mechanical Engineering Publications Limited, 1992.

[21] Hyundai. *What Does an Electric Car Sound Like?* 2021. URL: https://www.hyundai.news/eu/articles/stories/what-does-an-electric-car-sound-like.html.

[22] International Energy Agency. *Global EV Outlook 2021*. Licence: CC BY 4.0. Paris, 2021. URL: https://www.iea.org/reports/global-ev-outlook-2021.

[23] International Energy Agency. *Global EV Outlook 2023*. Licence: CC BY 4.0. Paris, 2023. URL: https://www.iea.org/reports/global-ev-outlook-2023.

[24] International Energy Agency. *Global EV Outlook 2024*. Licence: CC BY 4.0. Paris, 2024. URL: https://www.iea.org/reports/global-ev-outlook-2024.

[25] International Energy Agency (IEA). *Electric Vehicle Sales by Region and Scenario, 2030 and 2035*. Licence: CC BY 4.0. Paris, 2024. URL: https://www.iea.org/data-and-statistics/charts/electric-vehicle-sales-by-region-and-scenario-2030-and-2035.

[26] International Energy Agency (IEA). *Global Electric Car Stock, 2013-2023*. Licence: CC BY 4.0. Paris, 2024. URL: https://www.iea.org/data-and-statistics/charts/global-electric-car-stock-2013-2023.

[27] R. Irle. *EV-Volumes - The Electric Vehicle World Sales Database*. Glob. EV Sales 2021 H1 [Online]. 2021. URL: https://www.ev-volumes.com/.

[28] Paul Jennings et al. "Tools and Techniques for Understanding the Fundamentals of Automotive Sound Quality". In: *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 1 (Oct. 2010), pp. 1–16. DOI: 10.1243/09544070JAUTO1407.

[29] Yoshihiko Kozawa et al. "A Method for Evaluating Exhaust Sound Quality". In: *Journal of the Acoustical Society of Japan (E)* 48.11 (1992), pp. 807–814. DOI: 10.20697/jasj.48.11_807.

[30] May Jorella Lazaro et al. "Design and Evaluation of Electric Vehicle Sound Using Granular Synthesis". In: *Journal of the Audio Engineering Society* 70 (May 2022), pp. 294–304. DOI: 10.17743/jaes.2021.0062.

[31] C. Liu, L. Zhao, and C. Lu. "Exploration of the characteristics and trends of electric vehicle crashes: a case study in Norway". In: *European Transport Research Review* 14.6 (2022). DOI: 10.1186/s12544-022-00534-2.

[32] Richard H. Lyon. “Product Sound Quality - From Perception to Design”. In: *Sound and Vibration* 37 (Mar. 2003), pp. 18–22.

[33] H. Murata et al. *Sound Quality Evaluation of Passenger Vehicle Interior Noise*. Tech. rep. 931347. SAE, 1993.

[34] Y. Ohsasa and K. Kadomatsu. “Sound Quality Evaluation of Exhaust Note During Acceleration”. In: *SAE Technical Paper*. 951314. SAE International, 1995. DOI: 10.4271/951314.

[35] Francesco Orsi. “On the sustainability of electric vehicles: What about their impacts on land use?” In: *Sustainable Cities and Society* 66 (2021), p. 102680. DOI: 10.1016/j.scs.2020.102680.

[36] Edoardo Di Pietrantonio. *Automotive Granular Synthesizer (AGS): Design and Evaluation of a Sound Processing Algorithm for Electric Vehicles*. University of Padova, CSC, 2024.

[37] Porsche. *Porsche Sound - Taycan*. 2021. URL: https://media.porsche.com/mediakit/taycan/it/porsche-taycan/der-porsche-sound.

[38] Clarendon Press, ed. *The Oxford English Dictionary*. 2nd. Oxford, 1989.

[39] Kandethody M. Ramachandran and Chris P. Tsokos. “Chapter 9 - Analysis of variance”. In: *Mathematical Statistics with Applications in R (Third Edition)*. Ed. by Kandethody M. Ramachandran and Chris P. Tsokos. Academic Press, 2021, pp. 369–414. ISBN: 9780128178157. DOI: 10.1016/B978-0-12-817815-7.00009-9. URL: https://www.sciencedirect.com/science/article/pii/B9780128178157000099.

[40] *Regulation (EU) 2017/1576 of the European Parliament and of the Council*. 2017. URL: https://eur-lex.europa.eu/eli/reg_del/2017/1576/oj.

[41] Hannah Ritchie. “Tracking global data on electric vehicles”. In: *Our World in Data* (2024). https://ourworldindata.org/electric-car-sales.

[42] Curtis Roads. “Automated Granular Synthesis of Sounds”. In: *Computer Music Journal* 2.2 (Sept. 1978), pp. 61–62. DOI: 10.2307/3680222.

[43] Curtis Roads. “Introduction to Granular Synthesis”. In: *Computer Music Journal* 12.2 (1988), pp. 11–13.

[44] Curtis Roads. “The Evolution of Granular Synthesis: An Overview of Current Research”. In: *The Cambridge Handbook of Computer Music*. Cambridge University Press, 2009, pp. 229–240.

[45] Curtis Roads and John Strawn. *Foundations of Computer Music*. Cambridge, Massachusetts: MIT Press, 1985.

[46] A. Ronacher et al. "Evaluating Vehicle Interior Noise Quality Under Transient Driving Conditions". In: *SAE Technical Paper*. 1999-01-1683. SAE International, 1999. DOI: 10.4271/1999-01-1683.

[47] M. F. Russell, S. A. Worley, and C. D. Young. "Towards an Objective Estimate of the Subjective Reaction to Diesel Noise". In: *SAE Technical Paper*. 870958. Warrendale, PA: SAE International, 1987.

[48] J.A. Sanguesa et al. "A Review on Electric Vehicles: Technologies and Challenges". In: *Smart Cities* 4.1 (2021), pp. 372–404. DOI: 10.3390/smartcities4010022. URL: https://doi.org/10.3390/smartcities4010022.

[49] M. Schneider, M. Wilhelm, and N. Alt. "Development of Vehicle Sound Quality - Targets and Methods". In: *SAE Technical Paper*. 951283. SAE International, 1995. DOI: 10.4271/951283.

[50] Yein Song, Wonjoon Kim, and Myung Hwan Yun. "Auditory experience in vehicles: A systematic review and future research directions". In: *Heliyon* 10.15 (2024), e34838. ISSN: 2405-8440. DOI: 10.1016/j.heliyon.2024.e34838. URL: https://www.sciencedirect.com/science/article/pii/S2405844024108699.

[51] G. Stoet. "PsyToolkit - A software package for programming psychological experiments using Linux". In: *Behavior Research Methods* 42.4 (2010), pp. 1096–1104. DOI: 10.3758/BRM.42.4.1096.

[52] G. Stoet. "PsyToolkit: A novel web-based method for running online questionnaires and reaction-time experiments". In: *Teaching of Psychology* (in press).

[53] K. Takanami et al. "Improving Interior Noise Produced During Acceleration". In: *SAE Technical Paper*. 911078. SAE International, 1991. DOI: 10.4271/911078.

[54] Noriyoshi Terazawa, Yoshihiko Kozawa, and Tatsuo Shuku. "Objective Evaluation of Exciting Engine Sound in Passenger Compartment During Acceleration". In: *SAE Transactions* 109 (2000), pp. 386–394. URL: http://www.jstor.org/stable/44686883.

[55] G. Townsend. “A New Approach to the Impulsiveness in the Noise of Motor Vehicles”. In: *Proceedings of the Automotive Technology Congress and Exhibition (Autotech 85)*. Birmingham, UK: Mechanical Engineering Publications,

[56] K. Tsuge et al. “A study of noise in vehicle passenger compartment during acceleration”. In: *SAE Technical Paper*. 850965. SAE International, 1985.

[57] T. Wakita. “A Method for Evaluating Exhaust Sound QualitySteady Speed Sound and Sound Varying with Acceleration”. In: *Proceedings of the Japan Society of Automotive Engineers (JSAE)* 912.1 (1991), paper 912123.

[58] Wikipedia contributors. *Electric vehicle warning sounds — Wikipedia, The Free Encyclopedia*. `https://en.wikipedia.org/w/index.php?title=Electric_vehicle_warning_sounds&oldid=1243919996`. 2024.

[59] William A. Yost. “Overview: Psychoacoustics”. In: *Human Psychophysics*. Ed. by William A. Yost, Arthur N. Popper, and Richard R. Fay. Vol. 3. Springer Handbook of Auditory Research. New York, NY: Springer, 1993, pp. 1–12. DOI: `10.1007/978-1-4612-2728-1_1`.

[60] E. Yurtsever et al. “A Survey of Autonomous Driving: Common Practices and Emerging Technologies”. In: *IEEE Access* 8 (2020), pp. 58443–58469. DOI: `10.1109/ACCESS.2020.2983149`.