HELMUT SCHMIDT UNIVERSITÄT

Universität der Bundeswehr Hamburg

Tone Matching of a Monophonic Frequency Modulation Synthesizer using the Genetic Algorithm

Bachelor Thesis

by

Manuel Pastor Alemany

TUHH

Hamburg University of Technology

Start date:	01. May 2020
End date:	20. August 2020
Supervisor:	M. Sc. Etienne Gerat
Supervising Professor:	Prof. Dr.-Ing. habil. Udo Zölzer

Abstract

A synthesizer is a musical instrument whose sound is subsequently used in musical productions. Synthesizers can create, electronically, a variety of sounds with different timbres. It is capable of forming radically different sounds in such a flexible and versatile way. Programming a synthesizer is a complex task due to the amount of parameters it may have. In this project, a frequency modulation synthesizer is used as main working tool. The problem of tone matching is considered in this project where the goal is to achieve a sound similar to the specific target. The problem is addressed using optimization algorithms, specifically, the Genetic Algorithm, a tool that, through certain specifications, finds the values that best fit the parameters defined above.

Statement

Hereby I do state that this work has been prepared by myself and with the help which is referred within this thesis.

Hamburg, August 20th 2020

Foreword

The work presented here was developed in the Department of Signal Processing and Communication in the Helmut-Schmidt University Hamburg, as an exchange student from Polytechnic University of Valencia.

I would like to express my gratitude to Prof. Udo Zolzer for allowing me to do my bachelor thesis in the field I like the most of my degree. I would also like to thank my supervisor Etienne Gerat for being so patient with me and helping me understand the world of sound more fully. And of course, to thank my family for supporting me always. To my friends in Spain and the friends I took with me from Hamburg.

Hamburg, August 20th 2020

Contents

List of Figures

List of Tables

List of Symbols

$c(t)$	Carrier Wave
$m(t)$	Message Signal
$s(t)$	Modulated Signal
E_c	Amplitude of the carrier signal
ω_c	Angular Frequency of the carrier signal
ϕ_o	Initial phase of the Gold and Rader oscillator
f_c	Frequency of the carrier wave
f_s	Sampling Frequency
K_f	Amount of Hz for each volt of the modulating signal.
$y(n)$	Reference signal
$x(n)$	Output of the synthesizer
$\|Y(b,k)\|$	Magnitude of the reference signal
$\|X(b,k,p)\|$	Magnitude of the synthesizer output
$\|R(b,k,p)\|$	Magnitude of the substracion between magnitude of the reference signal and magnitude of the synthesizer output

Chapter 1

Introduction

1.1 Context

TUHH (Technical University of Hamburg - Harburg) is the university where I did my exchange semesters. I attended Digital Audio Signal Processing course given by professor Zölzer. I have learned about quantization, AD/DA conversion, audio filters, room simulation, dynamic range control, sampling rate conversion and audio coding.

As it was is the last semester of my bachelor's degree in Telecommunications Systems, Sound and Image Engineering I did my bachelor thesis in the HSU-HH (Helmut Schmidt Universität), the Federal Armed Forces of Hamburg, at the Department of Signal Processing and Communication.

My thesis is focused on digital audio signal processing, specifically on frequency modulation synthesizers and is supervised by professor Zölzer and Etienne Gerat.

1.2 The project

Since the Paleolithic, music has always been present in human culture. In one way or another, it was a cultural expression whether it was a religious rite or to break the loneliness of the workers.

At the end of the 19th century, some people already had audio recording devices. Tomas Edison invented the phonograph used to record monologues and famous speeches. In addition, some songs were included in this type of media. Around the 1920s, radio was invented with to the discovery of electromagnetic waves and the development of microphone technology. Microphone was responsible for collecting sound and converting it into an electrical wave.

In the 1980s, digital sound was introduced that sought to represent information digitally. It was a better method because an advantages of digital is that it is not sensible to the environment (temperature, humidity, etc) and that the information remains. From this new method, sound began to be represented as a sequence of numbers describing the behavior of the sound wave. The first digital synthesizers, as well as samplers, compressors and delays, arrived in the 1980s to introduce some kind of effect into the sound.

The first FM synthesizer is the GS-1 from 1980 made by Yamaha. Later, an analogue synthesizer called the SY-1 was released. It included an ADSR envelope generator (explained later) and two types of controls for adjusting the amplitude envelope. The SY-1 was capable of simulating the sound of a flute, guitar, and piano. Years later, Yamaha released a frequency-modulating synthesizer called the DX7 which had six operators with which it was possible to make more complex and elaborate sounds.

1.3 State of the art

There are many studies on the programming of sound synthesizers. One of the most important objectives of these machines is sound matching. The first algorithm used for frequency-modulation sound matching, finding the right value for each parameter, was the genetic algorithm described by Horner et al. [1]

The synthesizers are also capable of giving the user a tool with which to modify the timbre so that the user will be able to find the right values for the synthesizer parameters. Auto-learning was one of the techniques used by Arfib and others to generate a map that generates parameter values based on perceptual spectra. [2] To obtain a better search of the values for each parameter, there were researchers who generated both 2D and 3D interfaces for a synthesizer timbre environment, achieving a much more efficient research [3]. Dahlstedt helped to program the Nord Modular synthesizer using the interactive genetic algorithm, where the user was able to modify the frequency and select the best variations for better reproductions. [4] Recently a synthesizer program using the interactive genetic algorithm created by Yee-King was released as open source software. [5] This free software system is called SynthBot, capable of programming FM and AM synthesizers available in the Virtual Studio Technology (VST) format. [6] One of the latest studies on the topic is Tatar et al.'s PresetGen. Real instruments and artificial sounds, made with the OP-1 synthesizer created by Tennage Engineering, are paired by an advanced multi-object genetic algorithm. [7]

1.4 Motivation

Synthesizers are musical instruments that can both create previously unheard sounds and somewhat simulate sounds of real instruments. Nowadays, a music producer should have a synthesizer and therefore should know how to use this device since it opens up a whole new world of sound creation and handling.

The aim of this project is to build a frequency-modulated synthesizer which is capable of producing a sound similar to the reference one. Synthesizers create a note by using a MIDI keyboard where the sound is more dynamic. As the characteristics of a sound can be modified, either the texture or the shape, a user-parameterizable synthesizer has been designed. Since it is a virtual synthesizer, it is important to use the genetic algorithm (GA) as an optimization technique so that the algorithm is in charge of finding the optimal values of the user parameters that form a sound with a tone match to the reference one. Once the GA process is finished, a sound will be created using the synthesizer taking as input the results obtained from the previous step. In the last step, a psycho-acoustic test will be performed to analyze and evaluate the project.

1.5 Report Structure

- **Chapter 2:** presents the fundamentals needed to understand how Frequency Modulation synthesizer and the genetic algorithm work.
- **Chapter 3:** deals with the model of synthesizer, explaining each of the components of an operator, the combination between them and the common issues.
- **Chapter 4:** explains how to achieve a tone matching between a reference sound and another sound created by the FM synthesizer using GA.
- **Chapter 5:** depicts the results obtained after using several reference sounds and the conclusions.

Chapter 2

Fundamentals

2.1 Synthesis Techniques

From 1964, the world of synthesizers began to be commercialized. Many different techniques have been employed, and it has been concluded that the digital world offers a wide range of models beyond those possible in analog.

The three best known techniques in the world of synthesizers are [8]:

- **Additive Synthesis**: any sound can be described by the sum or addition in its respective amplitudes of only sine waves, or partials until the desired timbre is achieved if the concept of the Fourier series is applied.
- In **Subtractive Synthesis**, a filter is used to attenuate the partials of a signal in order to change the timbre of a sound. This method can be used in conjunction with other forms of synthesis whose function is to generate complex waves, eliminating unpleasant frequencies which in turn, softens the sound.
- In **Frequency Modulation Synthesis**, two types of signals are used, one to modulate the frequency of the other. The project will focus on this type of synthesis.

2.2 Frequency Modulation

Frequency Modulation was patented by Edwin Howard Armstrong, who was an American engineer and inventor in the Department of Electrical Engineering at the Columbia University, New York. [9]. It was first presented as a method in 1936 to reduce the disturbances in radio signals and to show how these developments may be utilized to produce a very great reduction in the effects of the various disturbances to which radio signal is subject. [9] Later, Frequency Modulation was applied in other fields such as Doppler effect in 1968, magnetic tape storage and sound.

FM is a type of non-linear or angular modulation in which the instantaneous radio frequency of a *carrier signal* varies according to the instantaneous amplitude of the *modulating signal* or

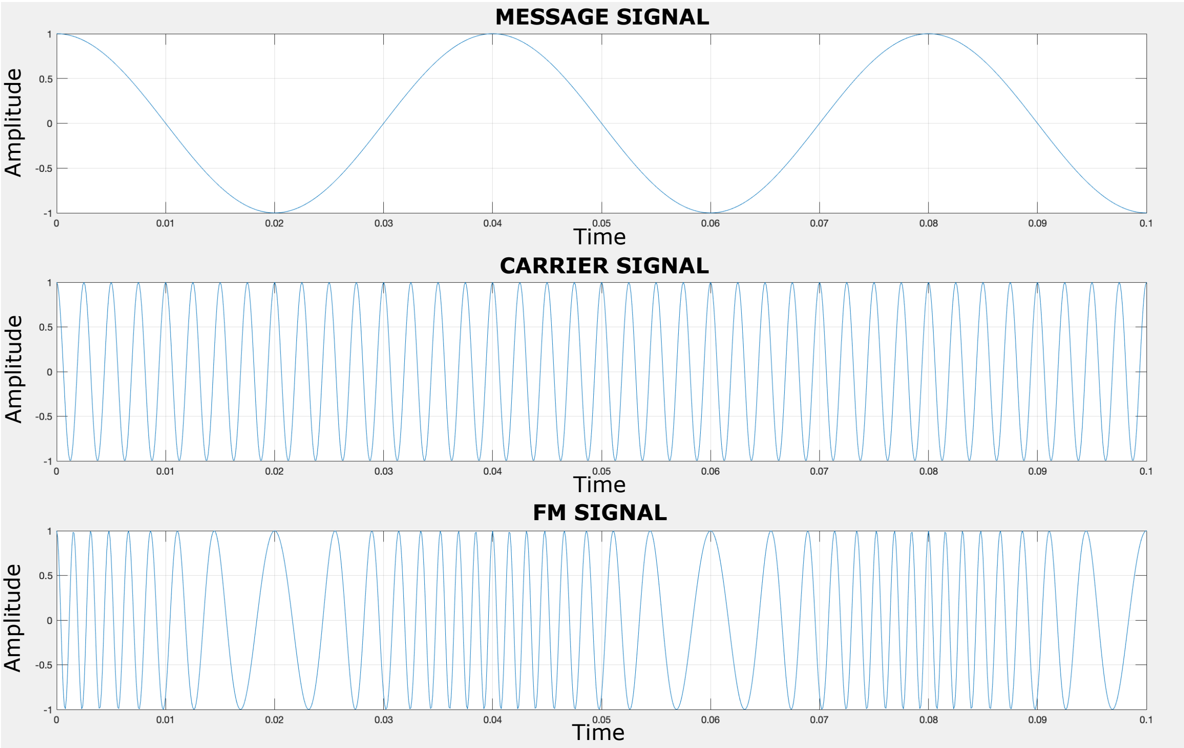


Figure 2.1: *Message, carrier and FM signals*

information wave. [10]. In this type of modulation, the phase and amplitude remain constant. It is the frequency of the carrier signal that changes as a function of the modulating signal.

Figure 2.1 shows the modulating, carrier and FM signals. The modulating signal is responsible for varying the frequency of the carrier signal. If the amplitude of the modulating signal increases, the frequency of the carrier signal increases. All these changes in amplitude, on the part of the modulating signal, and in frequency, on the part of the carrier signal, form the waveform in frequency modulation.

The carrier wave is defined as

$$c(t) = E_c \cdot cos(\omega_c \cdot t + \phi) \tag{2.1}$$

where E_c is the amplitude, ω_c the angular frequency, t time and ϕ the phase.

The message signal is defined by:

$$m(t) \tag{2.2}$$

Achieving the modulated FM signal is done by two steps: [11]

1. The modulating signal should be integrated to have an equation for the phase versus time.

2. Phase modulation encodes the message signal as variations in the instantaneous phase of the carrier wave.

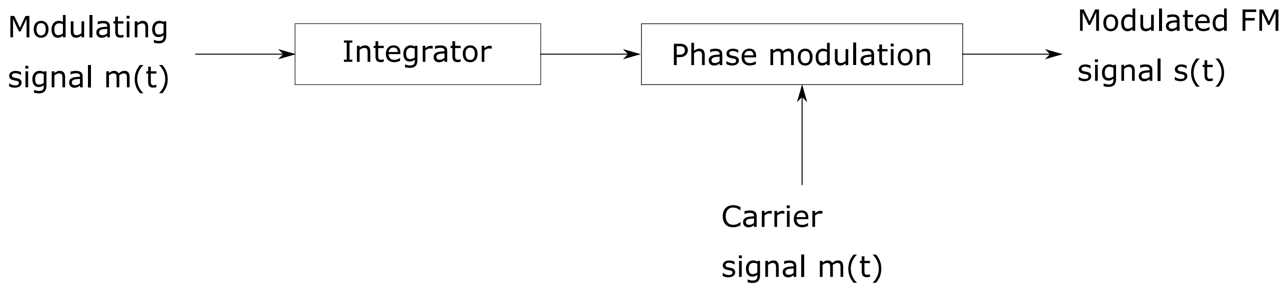


Figure 2.2: *Frequency Modulation diagram*

After the integrator, the result is an equation for the phase as a function of time:

$$\Theta(t) = 2 \cdot pi \cdot f_c \cdot t + 2 \cdot pi \cdot K_f \cdot \int_0^t m(t)\, dt \tag{2.3}$$

where f_c is the carrier frequency and K_f is the amount of Hz for each volt of the modulating signal. The modulated FM signal is obtained after the phase modulation as an equation as:

$$s(t) = E_c \cdot cos[2 \cdot pi \cdot f_c \cdot t + 2 \cdot pi \cdot K_f \cdot \int_0^t m(t)\, dt] \tag{2.4}$$

FM can be applied in differents fields:

1. **Radio**: reduce disturbances in a radio signal system. A characteristic is included in a transmitted waveform to avoid reproduction of the disturbances. This method includes in the system, a receiver that rejects all kinds of disturbances in the signal and responds only to the wave with special characteristic. [12]

2. **Sound**: in an audio signal, FM can be detected by witnessing a change in tone in the signal due to the change in the fundamental frequency of the sound. The harmonics that make up an audio signal are shifted, spectrally, in frequency. [13] In addition, FM is also used to synthesize a sound using the FM synthesis.

3. **Doppler effect**: the frequency of a wave, wavelength and amplitude varie due to the displacement of the source with respect to its observer. [14]

4. **Magnetic tape storage**: system used to copy video signals using frequency modulation where the reproduction of the carrier signal is modulated by a reproduction device and which, in addition, has a frequency limiting characteristic outside the frequency modulation range of the carrier signal. If a loss of amplitude is detected by the envelope of the carrier signal, a dummy signal is inserted at the missing location. A device in charge of magnetic recording takes the FM signal and transfers it to a magnetic tape. [15]

2.2.1 Frequency Modulation Synthesizers

The FM synthesis was created by Johnh Chowning in 1973. Its ability to create realistic and different sounds was powerful and low in computational cost. [16] It is a technique whereby the frequency of an carrier oscillator is altered, or distorted, in accordance with the amplitude of a modulating signal. [17]The FM synthesizer can generate sounds of various varieties and with many harmonics.

The SY-1 was the first AM synthesizer shown in Figure 2.3. Unfortunately, it was not very cost-effective because of the huge amount of integrated circuits required by the semiconductor technologies of the time, and also because of the difficulty experienced in finding the right balance between size and functionality. [18]

The first company to manufacture the first FM synthesizer was Yamaha which introduced the DX7 frequency modulation synthesizer to the world, shown in Figure 2.4. For a synthesizer to be able to generate tones, it must be made up of several operators whose function is to produce and change sounds. DX7 had a great impact on the music industry in the 1980. [18]

2.2.2 Operators used in FM syhnthesis

An operator is a set of components form a sine wave whose frequency is modulated and, in addition, its envelope is determined by another input [19] The basic components of an oscillator are: oscillator, ADSR Envelope (Attack, Decay, Sustain and Release) and a Voltage Controlled Amplifier (VCA).

- **Oscillator** is in charge of generating a wave that can have differents of shapes: sinusoidal, square or triangle. In most synthesizers, there is an adjustable frequency and amplitude.
- **ADSR Envelope**, shown in Figure 2.5, controls the volume over time. As the synthesizers are often controlled by a keyboard, depending on how long a key is pressed, the envelope will act in one way or another. The Attack, Decay and Sustain states will occur while the key is held down and the Release state will occur when the key is stopped being pressed.
- **VCA**'s output is managed by a control signal. It is responsible for modifying the amplitude of the signal depending on the amount of voltage applied to the AM control input. [20]

Operators can be differentiated into two types: carriers and modulators. The modulators are the ones in charge of changing the frequency of the carriers operators. Since their output is connected to another operator, the signal will not be considered as part of the audio path. The output of carriers operators is not connected to any other operator. The sound produced by this type of operator will be will be used as audio signal directed to the output of the synthesizer and therefore will be heard.

Figure 2.3: *SY-1 synthesizer*

2.2.3 ADSR envelope for synthesizers

ADSR envelopes aim to approximate the behavior of the most common instruments shown in Figure 2.5. Synthesizers also make use of it. Instrumens play a sound that changes according to the time between pressing the key and releasing it. It is one of the way of action from the musician to control the sound and make it more expressive. [21]

The states correspond to:

- **Attack**: it is a time variable whose value depends on the seconds in which the sound reaches the maximum volume once it has been activated (the key is pressed)
- **Decay**: it is a time variable whose value depends on the seconds in which the sound decays to the sustain level once the initial peak has been reached.
- **Sustain**: it is a variable that depends on the envelope value that denotes a constant level that the sound takes once it has dropped until the note is no longer pressed. This parameter denotes a level rather than a time period.
- **Release**: time variable whose value depends on the seconds it takes for the sound to stop being audible (the key is released)

2.3 Genetic Algorithm Terminology

GA is a method that mimics Charles Darwin's natural biological evolution [22]. It is considered a very effective system in both search and optimization. It is essentially an efficient, parallel and global search method, which can automatically acquire and accumulate knowledge about the search space, and adaptively control the search process to obtain the best solution. [23]

2.3.1 Natural Selection's fundamentals

- $Fitness function$ is that function which is to be optimised. The ability of an individual to compete with other individuals is determined by this [24]. It gives a fitness score, that

Figure 2.4: *DX7 synthesizer*

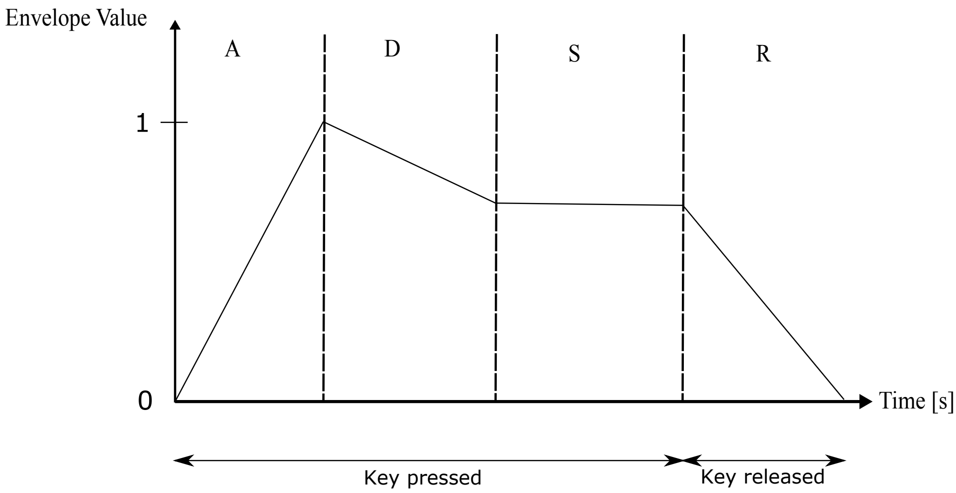


Figure 2.5: *ADSR Envelope*

will be used to rank individuals according to the optimization criterion. The overall aim is to find an individual that minimized this fitness function

- The fitness function is applied to any *individual*. The individual's score is taken from the fitness function value. [24]. Individuals are the possible solution and that the best is always selected among them according to the fitness function
- A group of individuals form the *population*. There can be replicas of the same individual. At each iteration, the algorithm produces a new population using the individuals of the present population. Each new group is called a new generation. [24]
- Each individual has a *fitness value* that is taken from the fitness function for that individual. The smallest fitness value for any individual in the population is the best fitness value. [24]
- The GA produces the new generation, called *children*, using individuals from the current population, called *parents*. Normally, the algorithm chooses the parents that have the best fitness value. [24]

2.3.2 Outline of the Genetic Algorithm

In this section, the steps of the Genetic Algorithm are explained, from the selection of the best individuals to the replacement of the new generation. The steps to follow are shown in Figure 2.7 [25]

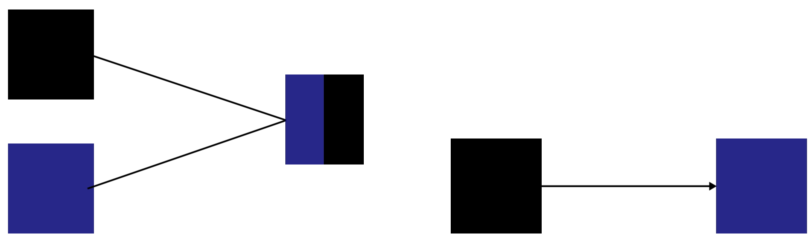
1. An initial random population is produced by the algorithm as soon as it starts. The initialization of the population is a random process. However, the number of individuals can be adjusted depending on the criteria to be used.
2. Each individual in the current population is qualified by the algorithm through the calculation of the fitness value. The fitness value determines how close the individual is to the required end. The fitness for an individual is generally determined algorithmically. The search for the best individual can be controlled in several ways until:

- The maximum number of generations have been passed.
- That an individual has met the performance criterion.
- The population converges on a single individual.

3. Some individuals are chosen for further reproduction. The most suitable individuals in a population are likely to survive and reproduce in the selection step of the algorithm. Selection process is based on the fitness value of the individual, so that the individuals most fitter are those chosen and those who can get to reproduce.

4. The children are produced by the previously chosen parents. They can be produced by the methods:

   - Reproduction: it selects individuals from the old generation, according a better chance to individuals presenting a better performance [26].
   - Mutation: a random change in the individual's "genetics", shown in Figure 2.6(b)
   - Crossover: combination of two individuals, shown in Figure 2.6(a).



(a) Crossover (b) Mutation

Figure 2.6: *Breeding process*

5. The current population is totally replaced by the new generation. When the next generation is produced, some former individuals may survive, specifically those with the best fitness value. By surviving intact, a hole is formed between them and the new children. A population with many individuals without a generational hole could lose a promising individual. A population with few individuals will not lose its best individuals.

6. The algorithm will be stop when the criteria is completed. The completion criterion can be set so that it ends when an optimal fitness value has been found, a specific number of new generations is exceeded, an application-dependent condition is applied, or any other criterion ensures the usefulness of the solution. [8]

Finding an algorithm completion criterion can be somewhat difficult. The fitness value of each individual can remain the same for several generations before a superior one is found. A common thing in this type of algorithm is to establish a criterion in which the GA ends in a certain number of generations. If there are no acceptable solutions, the GA can be restarted or a new search can be started.

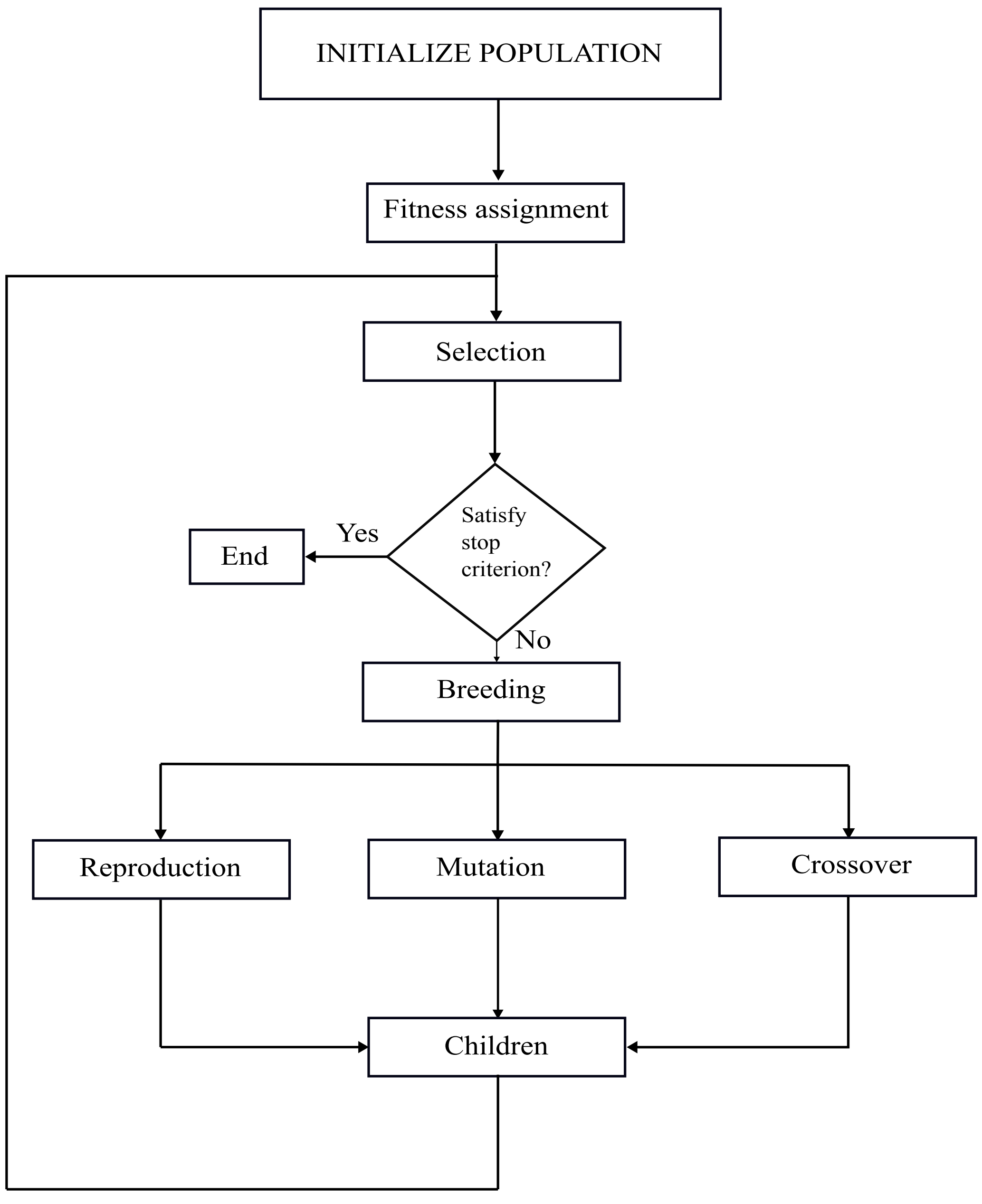


Figure 2.7: *GA process diagram*

Chapter 3

Synthesizer Model

In this project, a simple synthesizer is built using three operators. They can be combined in four difference architectures.

3.1 Operator implementation

3.1.1 Gold Rader filter

The Gold Rader filter (also called coupled form) is a particular form of Infinite Impulse response filter (IIR filter). It will be used here as an oscillator to produce a sinusoidal waveform.

Its difference equations are:

$$y_1(n) = b \cdot y_1(n-1) + a \cdot y_2(n-1) \tag{3.1}$$

$$y_2(n) = b \cdot y_2(n-1) - a \cdot y_1(n-1) \tag{3.2}$$

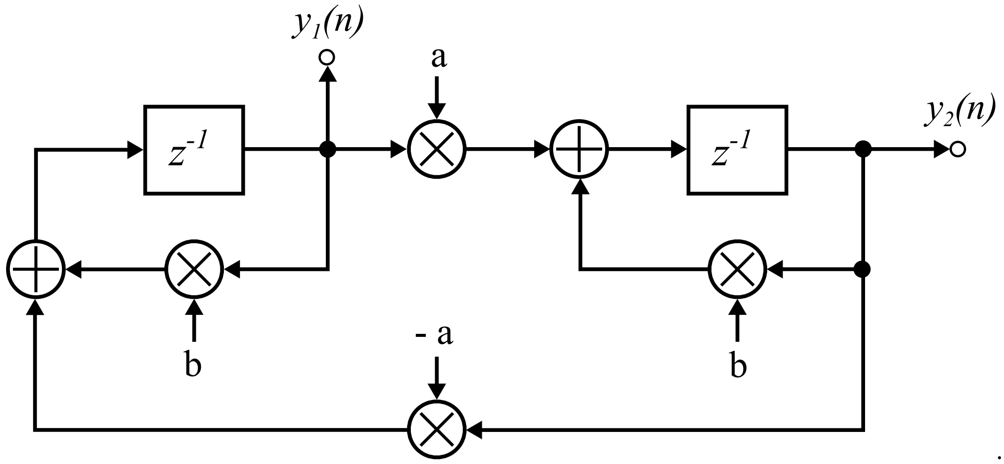


Figure 3.1: *Gold and Rader diagram*

Its variables take the values of:

$$a = sin(\omega_0) \tag{3.3}$$

$$b = cos(\omega_0) \tag{3.4}$$

$$\omega_0 = 2 \cdot \pi \cdot f_0/f_s \tag{3.5}$$

where ω_0 is the angular frequency, f_s is the sampling frequency, f_0 the fundamental frequency of the oscillator, a and b are variables whose values are defined to make the oscillator quasi-stable, as a self-oscillator is desired. The phase of the parameters a and b depend on the $fundamental\ frequency$.

Initializations of the states $y_1(0)$ and $y_2(0)$ define the starting amplitude and phase of the oscillator where A is the amplitude of the oscillator and θ_0 the initial phase.

$$y_1(0) = A \cdot sin(\theta_0) \tag{3.6}$$

$$y_2(0) = A \cdot cos(\theta_0) \tag{3.7}$$

As visible in Figure 3.1, the oscillator generates two signals y1(n) and y2(n). The signals are sinusoids at identical frequencies but phase shifted from 90 dregrees.

3.1.2 ADSR Envelope

To realize the ADSR envelope presented in chapter 2, the approach of a state machine has been chosen, as shown in Figure 3.2. ADSR envelope has four different states: Attack, Decay, Sustain and Release. The concept of state machine can be defined as an algorithm whose progression depends on the inputs and outputs. An additional state can be considered, when no key is pressed and the last state (Release) has come to an end. This state called idle can be seen as the initial state. The behavior of the state machine depends on two elements, one external (the note on or off thus a key is pressed or released) and on one internal (the current value of the envelope).

The State Machine process begins in the Idle State. Once a key is pressed, the process jumps to the Attack State. If the envelope value reaches its maximum, the next state would be Decay State, but if the note is no longer pressed before achieving the maximum value, the process would jump to the Release State.

In the Decay State, two things could happen: if the envelope value is less than the sustain level, there would be a change of state to the Sustain State, but if the note is no longer pressed, it would pass to the Release State.

The Sustain State is the moment when the envelope maintains the same value until when the key is no longer pressed.

The Release State occurs after the note is stopped holding down. The envelope decreases away little by little. Once the envelope reaches zero the process would jump to the Idle State.

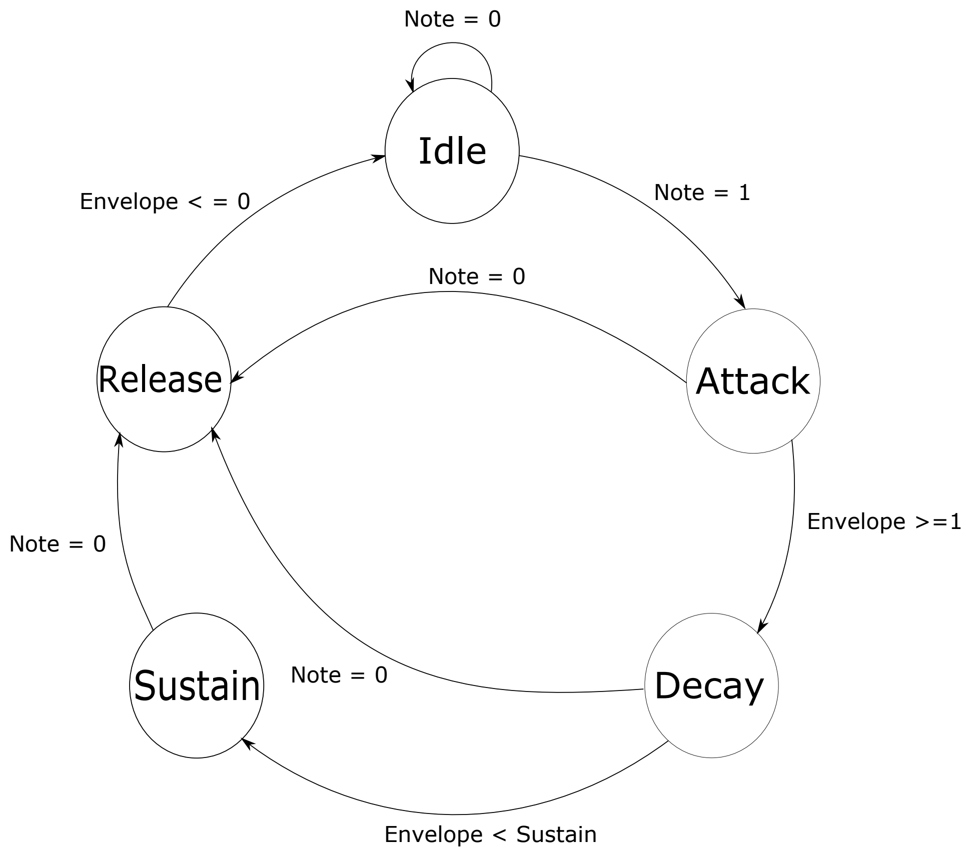


Figure 3.2: *State Machine diagram*

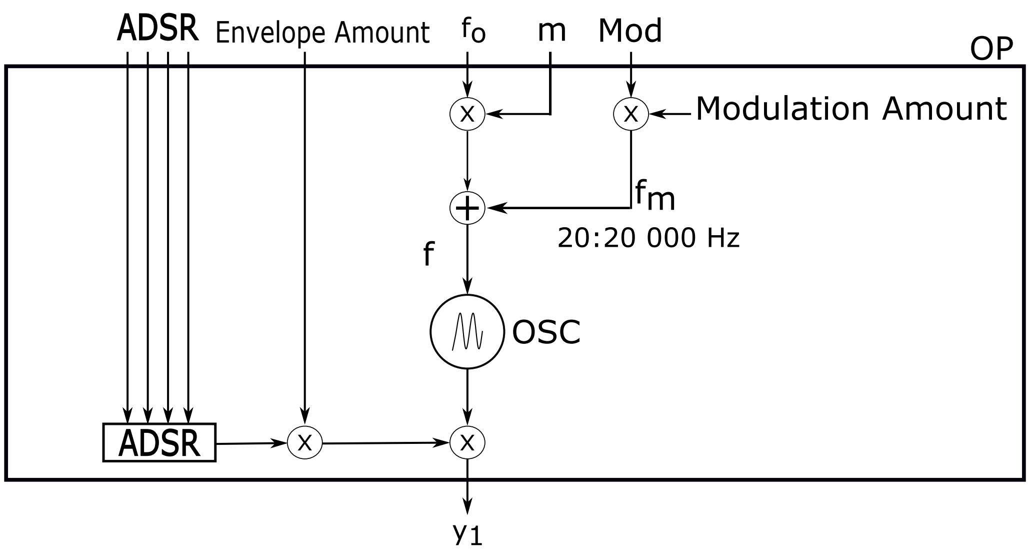


Figure 3.3: *Operator Diagram*

3.1.3 Additional Components and Operator's Output

Mainly, operators are built up by an oscillator, ADSR Envelope, and gain. Furthermore, it is needed additional components to avoid aliasing and to keep all the frequencies within the human earing range.

These additional components, referring to Figure 3.3 are:

- **Factor m** to allow some oscillator to be set at harmonics of f0 or octaves. The value of this parameter has to be a positive number.
- **Mod** variable to differentiate between a carrier operator or a modulation operator. It is initialized with a value of zero which means the operator will be a modulator. For being a carrier modulator, it should be assigned the output of the previous operator.
- An additional gain called **Modulation Amount** for operators which are carriers. As it is a Frequency Modulation synthesizer, the output of a modulator modulates the frequency of a carrier operator. The values of the modulator's output are between -1 and 1. A gain is needed to map this range to a range that makes sens in term of frequency (20-20kHz).
- A **function which band limits the frequency** coming from the modulator within the human earing range.

Once the frequency is band-limited, it is summed with the fundamental frequency multiplied by the modulating factor. The result will be the frequency of the oscillator. The operator's output will be the multiplication between the oscillator's output and the ADSR envelope multiplied by a gain as shown in Figure 3.3.

3.1.4 Improvement

As the computer should have sufficient memory resources to run several instances of the synthesizer during the optimization and also for the parallel operation between many synthesizers, working sample by sample is needed

Internal states are variables who store their values unless they are altered by an assignment. They need to be saved from an iteration to the next. ADSR envelope, the parameters are current state and output envelope which is used for shaping the output and also to determine which state is the next. Gold and Rader filter, the variables are previous states of the filter. The internal states are assigned as outputs to be used in the next iteration as inputs.

3.2 Operators combination

There is two types of operators: modulators or carriers. In this project, the number of operators has been limited to three, to reduce the complexity of the optimization task. The differents combinations that can be done with three operators are:

Users Parameters				
OP1	OP2	OP3	Lower Bound	Upper Bound
Attack Time 1	Attack Time 2	Attack Time 3	0	2
Decay Time 1	Decay Time 2	Decay Time 3	0	4
Sustain Level 1	Sustain Level 2	Sustain Level 3	0	10
Release Time 1	Release Time 2	Release Time 3	0	1
Envelope Amount 1	Envelope Amount 2	Envelope Amount 3	0	1
Modulate Factor 1	Modulate Factor 2	Modulate Factor 3	0.5	64
Algorithm in FM synthesis			1	4

Table 3.1: *Users Parameters*

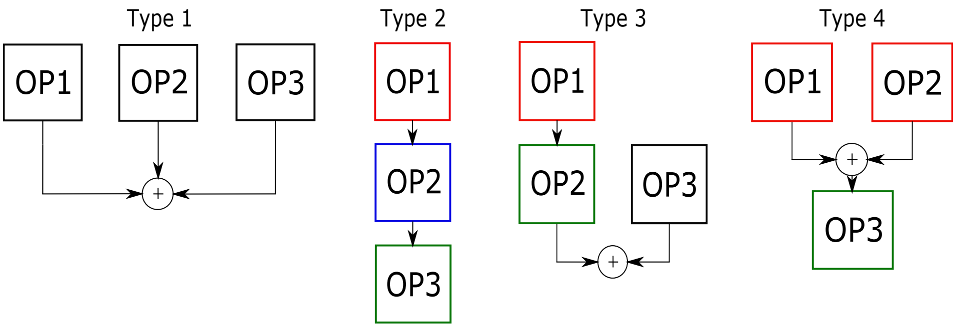


Figure 3.4: *Four differents operators combinations*

As shown in Figure 3.4, four types of combinations of operators are defined. In black, those whose output is added with another output of any type of operator. In Figure 3.5, it is showned how one operator modulates other one. In red, the operators that modulate the frequency of another operator. The sum of two operators modulating another operator, converts them also into modulators. In green, those operators whose frequency is modulated by another operator. Finally, in blue, those operators which act as carrier operator and also modulate the frequency of another one.

3.2.0.1 Users Parameters

Users parameters are parameters that can be changed manually by the user. There are some of the parameters from the ADSR Envelope as the *Attack Time*, *Decay Time*, *Sustain Level* and *Release Time*. Other parameters as the *Envelope Amount*, *Modulate Factor* and the *Algorithm in FM synthesis*. These user parameters are those that are optimized during the GA.

3.3 Common issues

Aliasing (also called spectral overlap) occurs when half the sample rate is exceeded by the highest harmonics in the signal. When the harmonics are in the low frequency zone of the spectrum, signal distortion occurs. In order to avoid this problem, the use of a low-pass filter

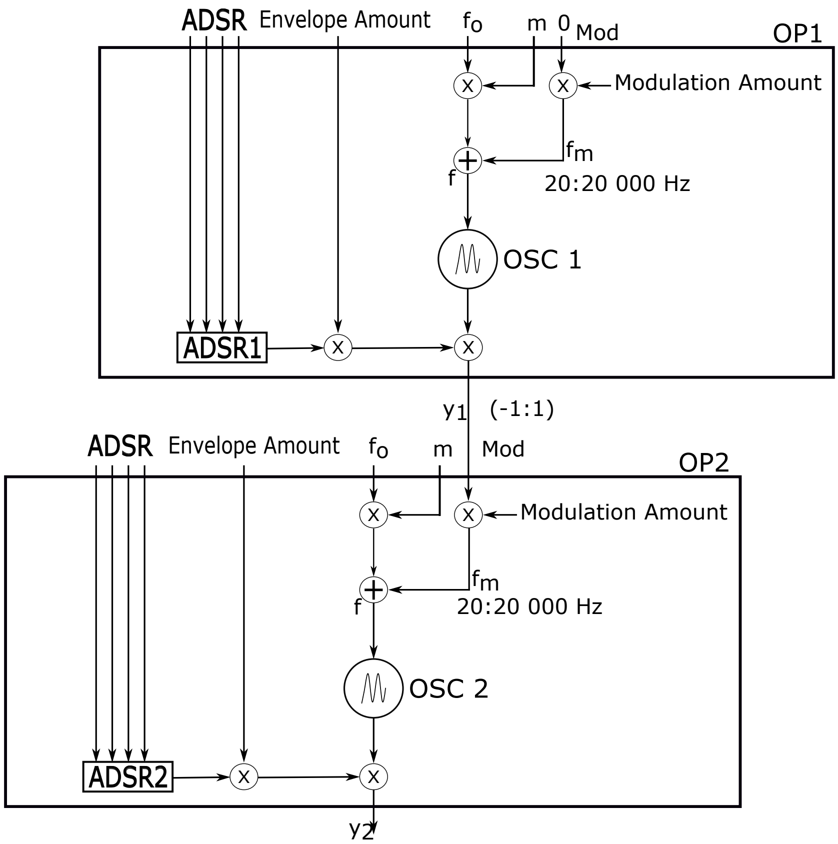


Figure 3.5: *Operators combination diagram*

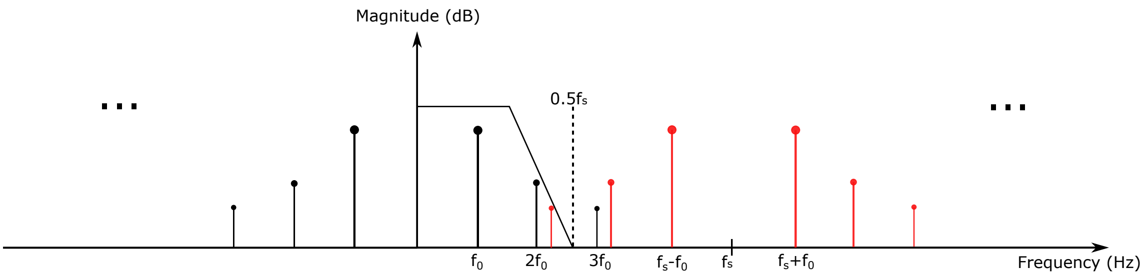


Figure 3.6: *Example of a spectrum of an aliased signal*

is necessary with a cut-off frequency below than the Nyquist frequency.

$$f_c = f_s/2 \tag{3.8}$$

Example: A sinus signal is wanted to be synthesized which contains the first three harmonics with a fundamental frequency $f_0 = 8000\ Hz$ in a system with sampling frequency $f_s = 44100\ Hz$. [27]

The harmonics are the integer multiply of the fundamental frequency. The harmonics of f_0 will be $2f_0$, $3f_0$, etc. The third harmonic will exceed the Nyquist frequency and it will be reflected in $f_s - 3f_0$ producing distorsion and aliasing.

Band-limiting of the input is carried out by a Low Pass Filter

$$f_c = f_s/2 = 22000Hz \tag{3.9}$$

The filter is not cutting abruptly, so it has to be before so at the nyquist frequency everything is cutted.

Chapter 4

Tone Matching

Tone Matching is one of the important process when using a synthesizer. This operation is usually done manually, and it is long an tedious especially for unexperienced users. The main goal of this project is to perform an automatic tone matching, in other words, find the user parameters that will emit a sound as close as possible to the target sound. [1] The inputs taken by a tone matching program are the sound synthesis algorithm ed by the and the desired sound. [28]

4.1 GA applied to FM synthesizer

The Tone Matching of a monophonic Frequency Modulation synthesizer is done by taking a simple reference sound. The sound could be a note from an instrument or just a simple sound built up from the synthesizer. The project's goal is to reach a similar sound by using a synthesizer's model. The Genetic Algorithm will tune the parameters of synthesizer in order to generate a sound perceptually close to a given reference sound.

4.1.1 Population and chromosomes

The first step of the Genetic Algorithm is to initialize population which is a number of potential solutions. In this case, a population of 200 individuals is initialized. Following this recommendation: *Positive integer of population's size of 50 when the number of variables is less than or equal to 5, otherwise 200 population's size.* [29] The most common type of coding used in GAs is binary, however codings using integers or real value numbers are becoming more widespread. In this project, coding with real value representations has been carried out. Real-value coding in GA offers several advantages over binary coding. Less memory is required since the efficient floating point of the computer's internal representations can be used directly; since there is no discretion of numbers, there is no loss of accuracy; and more variations of genetic operators can be employed. [30]

The GA will optimize the $Users\ Parameters$. These parameters can be seen as a chromosoma. To limit de complexity of the search [31], the Users Parameters are constrained in a range that makes sens for what they represent. All parameters representing time will be ranged between zero and few seconds. The upper limit is here chosen arbitrarlly in order to limit the overall length of the sounds produces to a maximum of two seconds.

4.1.2 Fitness score and selection strategy

During the optimization process, it is necessary to check which individuals perform best during the problem. To make this possible, there is a so-called objective function. Each individual has to be evaluated using the selection algorithm. This is responsible for choosing those individuals whose relative aptitude is better for subsequent reproduction. The selection algorithm is responsible for defining how many times an individual is chosen for subsequent reproduction and how many offspring it will produce.The new generation will have approximately the number of descendants proportional to the performance of the individuals. The selection of individuals is defined in two steps [30]:

1. The number of trials an individual can expect.
2. The discrete number of offspring will take the value of the expected number of tests.

Three methods of reproduction have been carried out during this project. First, the crossover breeding system was used in which the offspring acquire characteristics from both parents. To use this breeding system, the fraction of the next generation population that the crossover forms has to be determined. This function has to take the value of a positive scalar equal to 0.8. [29] The reproduction system called mutation is a random process where a gene is chosen to modify and replace some of its characteristics. To be able to use this method, the previous function has to take the value of 1. There is a system called reproduction that only chooses those individuals that can offer a better performance in the future generation. By default, the GA algorithm provided by Matlab uses this method. In some compilations, another option has been added to try to improve optimization: the N_{elite} best chromosomes are kept and the rest of the population is generated with reproduction, crossover or mutation on selected chromosomes. The resulting chromosomes are used in the new generation. [31] Once the reproduction process is over, the new individuals have to replace the old population.

It is difficult to determine a convergence criterion because the fitness value can remain the same after many generations before finding an optimal result. One criterion is to finish the optimization after a certain number of generations. In this project, GA stops if the average relative change in the best fitness function value is less than or equal to the $Function\,Tolerance$ [29].The genetic algorithm will return the user parameters that best suit the synthesizer to form a sound very similar to the reference sound. These optimized values will be used to build the sound using the synthesizer built in this project as shown in Figure 4.2.

The genetic algorithm is developed as shown in the Figure 4.1. The x-axis is the number of generations that the machine has used to reach the optimized values. The y-axis is the penalty value of the individuals. The blue points are the average of the penalty value and the black points are the best penalty value scores. Each point represents an individual to whom the fitness function will be applied and therefore has a fitness value. The aim of this toolbox software is to find the minimum values for this function. The graph shows a decrease in the best penalty value generation after generation.

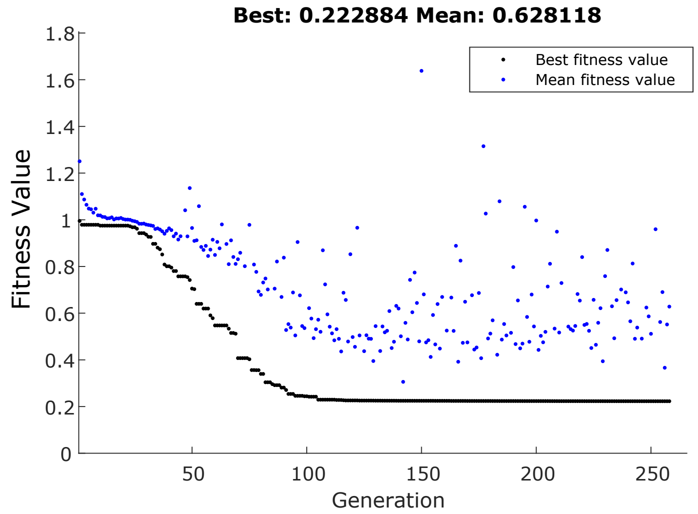


Figure 4.1: *Evaluating audio similarity*

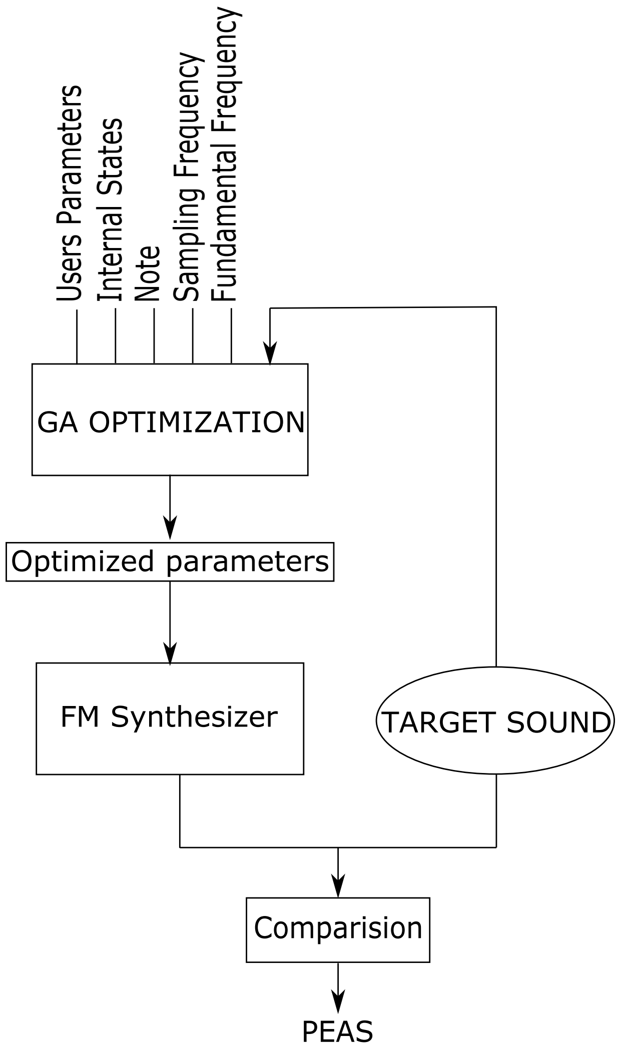


Figure 4.2: *Similarity value of the sounds*

4.2 Perceptually Motivated Metrics

Objective metrics provide a mathematical description of an error signal. A human can perceive that two sounds are heard equally, which leads one to believe that these two signals have the same mathematical characteristics. However, there are signals that are felt as being the same sound even if they have different parameters. [32]

4.2.1 Perceptual Evaluation of Audio Similarity (PEAS)

PEAS method is in charge to figure out the similarity between the reference signal and synthesizer's sound. Its solution is a rate value between two signals from a perceptual point of view. If its value is equal to zeros means the signal coming from the synthesizer is equal from the refence sound in a perceptive way. If its value is greater than one, it indicates the signals are different. [32]

As Figure 4.3 explains, the reference signal $y(n)$ (the sound to be imitated) and the output of the synthesizer, $x(n)$, must be taken as inputs, having used the previously calculated optimized parameters. The Short-Time Fourier Transform (STFT) is applied to both signals to convert them to the time-frequency domain. The STFT magnitude of each signal is calculated in the post-processing block, $|Y(b,k)|$ and $|X(b,k,p)|$. The residual is calculated by subtracting the magnitude of the two signals used, getting $|R(b,k,p)|$. To calculate the final result, first the result of the subtractions has to be squared and each of its elements added. The same process has to be done with the magnitude of the reference signal. The final result will be the division between the two previous operations shown in the Equation 4.1 [32].

$$PEAS = \frac{\sum_{\widetilde{k}} \sum_b (R(b,\widetilde{k},p)^2)}{\sum_{\widetilde{k}} \sum_b (|Y(b,\widetilde{k})|^2)} \tag{4.1}$$

With this score, the best optimization solutions will be known and the user parameters corresponding to the best PEAS will be used to build the signal that will simulate the reference sound.

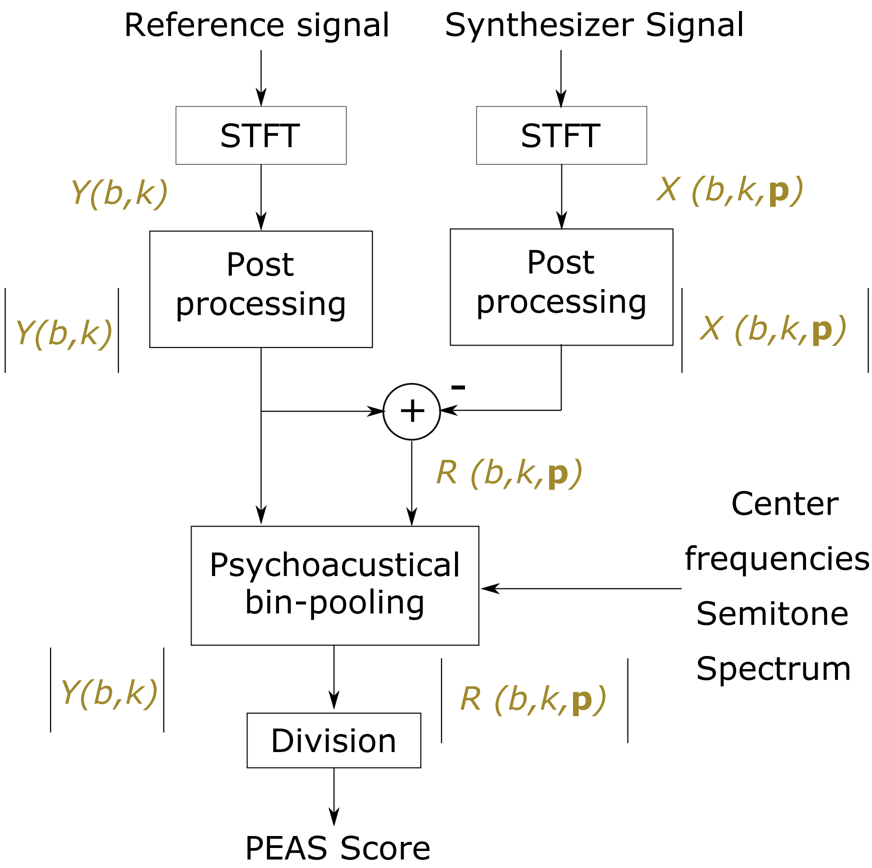


Figure 4.3: *Evaluating audio similarity*

Chapter 5

Results and Conclusions

During this project, several types of sounds were tested applying the Genetic Algorithm using differents features to achieve the best tone matching between the reference sound and the synthesizer sound. The synthesizer only works with mono sounds. Those sounds that were in the stereo format have been converted to mono.

5.1 DX7 sounds

The idea is to use DX7 synthesizer from Yamaha sounds with two different frequencies for each of them. The note A2 with a frequency of 110 Hz and the note A3 with a frequency of 220 Hz. To achieve a fair comparison, the sounds of the DX7 were generate with only three of its operators. Seventeen different sounds have been employed, numbered from 1 - 9 and specified by their frequency. To check which type of method is more effective, it has been used: reproduction, crossover and mutation.

Both when reproduction and mutation are applied, the ratio between the original sound and the sound formed by the synthesizer are very high, almost reaching one. When using the optimized values of the user parameters to create the synthesized sound, the result is not a sound similar to the reference sound. For all the sounds that have been tried to imitate, high values of the PEAS have been obtained shown in Figure 5.1 and in Figure 5.2.

Applying the crossover method the values obtained are, for the most part, better than the previous results as shown in Figure 5.3. Several user parameters have been obtained that can build a sound similar to the original as shown by the PEAS value. Some of the best results are shown in Table 5.1. Taking the DX7 - 3 A2 sound as a reference, the values to be taken by the user parameters are those shown in the Table 5.2. In Appendix A.1 , a comparison of the PEAS results between the three breeding methods used is shown.

	DX7 - 3 A2	DX7 - 3 A3	DX7 - 6 A2	DX7 - 7 A2
PEAS	0,22277	0,33061	0,36382	0,31971

Table 5.1: *PEAS score of the best tone matched DX7 sounds*

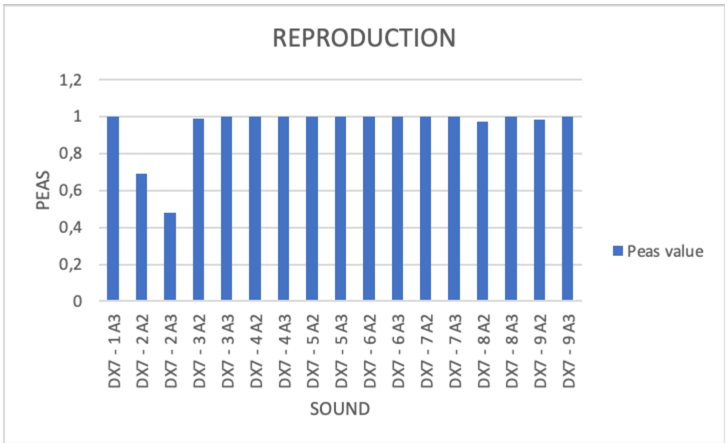


Figure 5.1: *Peas value of the DX7 sounds using reproduction*

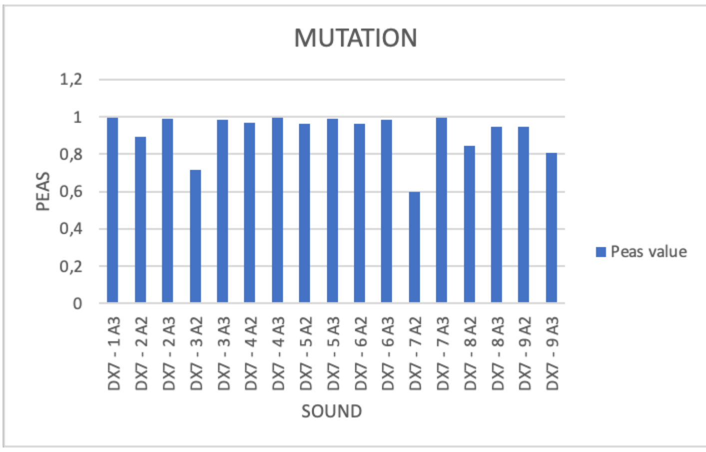


Figure 5.2: *Peas value of the DX7 sounds using mutation*

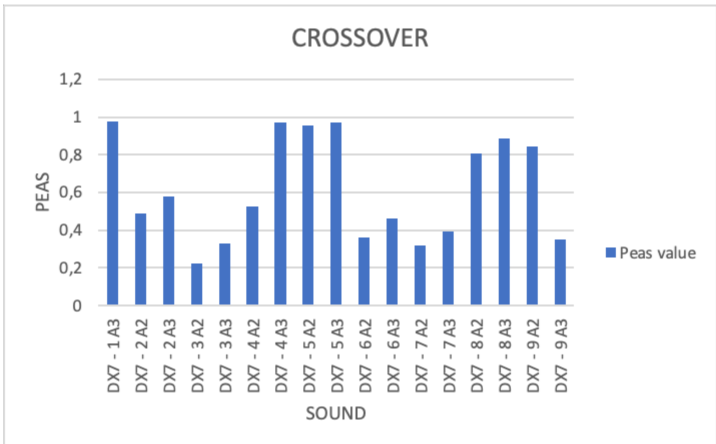


Figure 5.3: *Peas value of the DX7 sounds using crossover*

	Operator 1	**Operator 2**	**Operator 3**
Attack Time (s)	1.4183	0.0992	0.0268
Decay Time (s)	2.8996	0.0802	1.0431
Release Time (s)	8.1	6.2440	0.0206
Sustain Level	0.1647	0.0899	0.1085
Envelope Amount	0.9591	0.0014	0.0676
Modulate Factor	54.5155	33.2936	1
Current Combination		3	

Table 5.2: *DX7 - 3 results for the note A2*

Musical Instruments				
Note	Piano	Trumpet	Flute	Violin
G3 (196 Hz)	✓	✓	×	×
C4 (261 Hz)	✓	✓	✓	✓
G4 (392 Hz)	✓	✓	✓	✓
C5 (523 Hz)	✓	✓	✓	✓
G5 (784 Hz)	✓	✓	✓	✓
C6 (1046 Hz)	✓	✓	✓	✓
G6 (1568 Hz)	✓	×	✓	✓

Table 5.3: *Musical instruments and notes*

After analyzing the results, it is concluded that the most effective method of breeding is the crossover because it achieves optimal values that offer a sound very similar to the original.

5.2 Instruments

In this section, the user parameters have been optimized with instruments such as flute, piano, trumpet and violin. As in the previous section, it was shown that the most efficient reproductive system was the crossover, it has been used again to achieve the tone matching with real instrument sounds. The sounds were downloaded from a web page [33] where the instruments mentioned above with different notes are found. Each note was assigned a fundamental frequency used for optimization. The audios were mono and sampled with a frequency of 11 kHz. The sample frequency used to generate a similar sound is the same as the downloaded audios. In Table 5.3 the instruments with the note used are shown. The gaps that have a checkmark are those that have been worked with.

The results obtained show a somewhat high ratio between the reference sound and that generated by the synthesizer. In Table 5.4, the best values are shown. This is due to the simplicity of the synthesizer as it only uses three operators. To improve the PEAS value more operators could be included, to have more combinations between them and thus have more possibilities to create a more similar sound.

	Piano	Trumpet	Flute	Violin
PEAS	0.88718	0.97523	0.80557	0.9636

Table 5.4: *PEAS score of the best tone matched Instrument sounds*

	hi.wav	huh.wav	high-pitched whistle	low whistle
PEAS	0.90436	0.9276	0.9498	0.91405

Table 5.5: *PEAS score of the human sounds*

5.3 Human Sounds

To simulate the sound of a human, two whistles of different tones and two human expressions have been used.

In order to know the fundamental frequency, a method that can be done in two different softwares:

- $MATLAB$ can be used to determine the fundamental frequency of a human-originated sound. To do this, the Fourier Transform has to be used and then, locate the peaks. The fundamental frequency will usually be the peak with the lowest frequency.
- Another system to find out the fundamental frequency is the *Audacity* program. To do this, the reference sound has to be imported. Once imported, there is a tool in the program which displays the spectral analysis. As with the previous method, the first peak to appear will be the fundamental frequency.

In this section, two types of whistles with different pitches have been used. In addition, monosyllabic words recorded with a human voice have also been used. After optimization, very high PEAS results have been obtained, as shown in Table 5.5 which means that the algorithm is not able to search for values for user parameters that construct a signal similar to the reference sound. It is concluded that the three-operator synthesizer is not capable of recreating human voices.

5.4 Psycho-acoustic test

Psychoacoustics is the science that studies the perception of sound and how auditory sensations are related to acoustic emissions. In this project, several sounds from synthesizers in the DX7 range were analyzed. Two different frequencies were applied to the same sound. The first frequency was 110 Hz and the second frequency was 220 Hz. Doubling the frequency is perceived with the increase of one octave. The sounds created by the synthesizer are complex sounds.

The test that has been carried out is a balance test. Balance tests are the ones that measure the user experience, whether it is the same intensity, same frequency, same duration, between the reference sounds and the sounds to be evaluated. The users will evaluate the sounds by

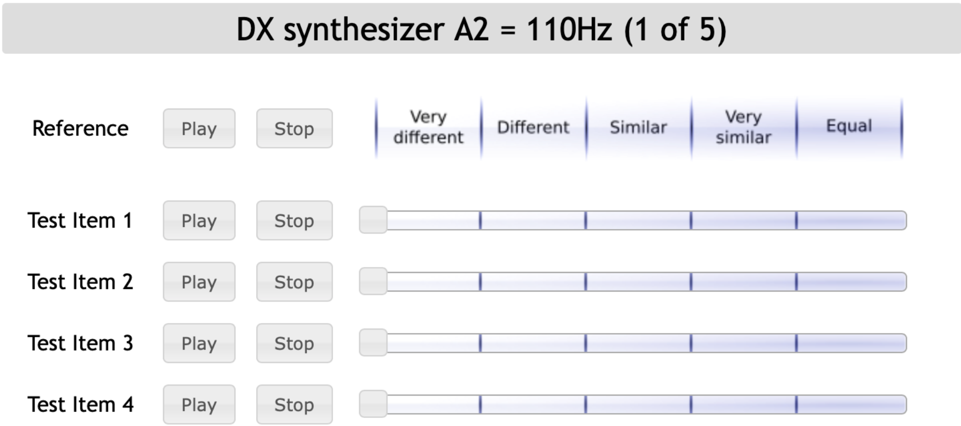


Figure 5.4: *Image of the pyschoacoustic test*

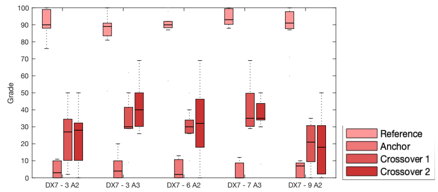


Figure 5.5: *Results of the listening test*

means of a gradual scale that goes from very different to equal [34], as shown in Figure 5.4. Four options have been included in the test in which the similarity between these and the reference sound should be evaluated. What users do not know is that within these options, one of them contains the same audio as the reference one. There are two options called Crossover 1 and Crossover 2 which are audios built by the synthesizer using the optimized user parameters. These models had a PEAS of between 0.2 and 0.4. The last option is the anchor. It is the sound that sounds the worst. It has been built from a random value vector within the numerical range of the user parameters.

The results of the psycho-acoustic test were analysed using the boxplot, shown in Figure 5.5. The evaluation consists of the similarity between the reference sound and the synthesizer sound where 100 means that there is no difference and 0 means that the sound has no similarity. The line inside each box represents the average of all the values. This line divides the box in two, the bottom (first quartile) is 25% of all evaluations that have an equal or lesser value and the top (third quartile) is 75% of all evaluations with an equal or greater value.

The test was performed by eleven people. Most people had no prior knowledge of frequency-modulating synthesizers and only one was a musician. Despite having obtained a PEAS value very suitable for the job, the two models built with the synthesizer using the optimized parameters, users have not noticed any equality between these and the reference sound. In some cases, these sounds have been evaluated at 70% grade.

5.5 Conclusions and future work

This project aimed to build a frequency-modulated synthesizer. The benefits of this type of synthesis are the amount and variability of sounds that can be created with this method and the direct access to the sound parameters. Therefore, a frequency modulation synthesizer has been built with a limited set of user parameters so that it could automatically generate sounds similar to the target sound by using the genetic algorithm working with Matlab.

Having applied the three breeding methods of the genetic algorithm, it has been demonstrated that the method that most optimizes the values of the user parameters is the so-called crossover. Even so, full similarity between two sounds has not been achieved. The PEAS value was in charge of calculating the similarity between two signals. For the reference sounds of the DX7 synthesizer, the smallest value that has been achieved, and therefore the one that a more similar sound produces, is around 0.2. For instrument and human sounds, a suitable PEAS value has not been achieved, leading to the failed simulation attempt. Several improvements can be followed, adding more operators would increase the number of combinations between them and therefore also increase the complexity of the synthesizer.

Future work could be the extension and optimization of the present synthesizer. A synthesizer could also be made in which the user would be able to decide how many operators he wants and in which way he wants to interconnect them. [27] Other optimization methods after the GA, like gradient based since the the combination could be first found with the GA. Find a more perceptual fitness function than PEAS.

Appendix A

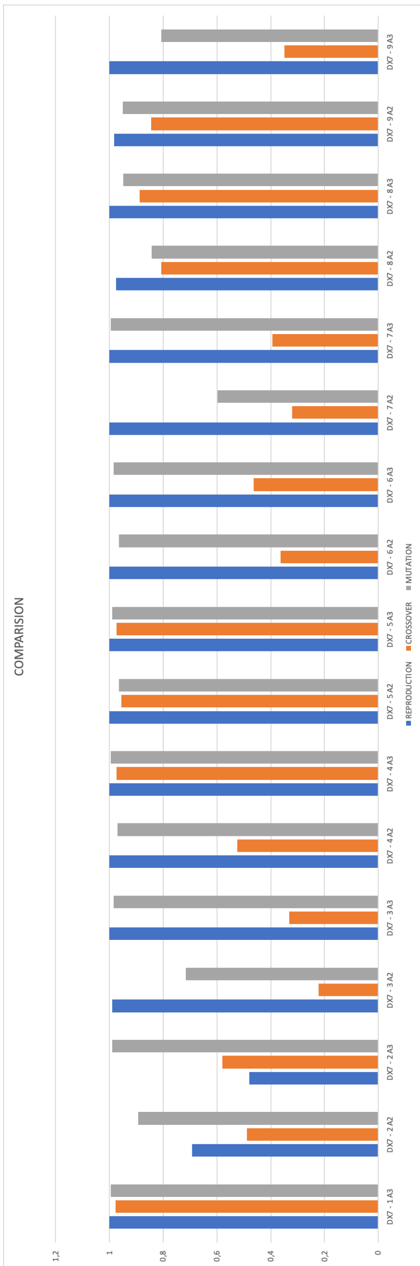


Figure A.1: *Comparision between reproduction, mutation and crossover*

List of Abbreviations

A

ADSR	Attack Decay Sustain and Release

F

FM	Frequency Modulation

G

GA	Genetic Algorithm

H

HSU-HH	Helmut-Schmidt-University/University of the Federal Armed Forces Hamburg

I

IIR	Infinite Impulse Response

P

PEAS	Perceptual Evaluation of Audio Similarity

S

STFT	Short Time Fourier Transform

T

TUHH	Technical University of Hamburg - Harburg

V

VCA	Voltage Controlled Amplifier
VST	Virtual Studio Technology

List of Software

Name	Version	URL	Comment
MATLAB	R2018a	mathworks.com	
Inkscape	1.0	inkscape.org	Vectorial drawing
TexStudio	2.12.22	www.texstudio.org	Presentation adn report
JavaScript			Pyschoacoustic test

Table A.2: *Used softwares*

Bibliography

[1] M. J. Yee-King, L. Fedden, and M. d'Inverno, "Automatic programming of vst sound synthesizers using deep networks and other techniques," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 2, pp. 150–159, 2018.

[2] D. Arfib, J.-M. Couturier, L. Kessous, and V. Verfaille, "Strategies of mapping between gesture data and synthesis model parameters using perceptual spaces," *Organised sound*, vol. 7, no. 2, p. 127, 2002.

[3] S. Fasciani, "Tsam: a tool for analyzing, modeling, and mapping the timbre of sound synthesizers," 2016.

[4] P. Dahlstedt, "Creating and exploring huge parameter spaces: Interactive evolution as a tool for sound generation," in *Proceedings of the International Computer Music Conference, Habana, Cuba*, 2001.

[5] M. J. Yee-King, "The use of interactive genetic algorithms in sound design: a comparison study," *Computers in Entertainmnent: Special Issue on Musical Metacreation*, 2016.

[6] M. Yee-King and M. Roth, "Synthbot: An unsupervised software synthesizer programmer," in *ICMC*, 2008.

[7] K. Tatar, M. Macret, and P. Pasquier, "Automatic synthesizer preset generation with presetgen," *Journal of New Music Research*, vol. 45, no. 2, pp. 124–144, 2016.

[8] A. Johnson and I. Phillips, "Sound resynthesis with a genetic algorithm," *Imperial College London*, 2011.

[9] E. H. Armstrong, "A method of reducing disturbances in radio signaling by a system of frequency modulation," *Proceedings of the Institute of Radio Engineers*, vol. 24, no. 5, pp. 689–740, 1936.

[10] A. J. P. González, "Frequency modulation and demodulation: Design and construction of fm training modules."

[11] E. Ambitiously, "National instruments measurement fundamentals series: Frequency modulation," https://www.ni.com/de-de/innovations/white-papers/06/frequency-modulation--fm-.html, updated in 2019, [Online; accessed 28-June-2020].

[12] E. H. Armstrong, "A method of reducing disturbances in radio signaling by a system of frequency modulation," *Proceedings of the Institute of Radio Engineers*, vol. 24, no. 5, pp. 689–740, 1936.

[13] D. K. Mellinger, "Feature-map methods for extracting sound frequency modulation," in *Conference Record of the Twenty-Fifth Asilomar Conference on Signals, Systems & Computers.* IEEE Computer Society, 1991, pp. 795–796.

[14] D. Censor, "The generalized doppler effect and applications," *Journal of the Franklin Institute*, vol. 295, no. 2, pp. 103–116, 1973.

[15] J. Sochor, "Process for directly copying frequency modulated video signals from magnetic tape to magnetic tape," Apr. 7 1992, uS Patent 5,103,349.

[16] A. Oland and R. B. Dannenberg, "FM synthesis: Introduction to Computer Music," Carnegie Mellon University, page 1.

[17] J. Chowning, "The synthesis of complex audio spectra by means of frequency modulation," *Journal of the Audio Engineering Society*, pp. J. Audio Eng. Soc. 21 (7), 526–534., 1973.

[18] Yamaha, "Chapter 2: Fm tone generators and the birth of home music production," https://es.yamaha.com/es/products/contents/music_production/synth_40th/history/chapter02/index.html, [Online; accessed 22-July-2020].

[19] D. Benson, "Mathematics and Music," Department of Mathematics, University of Georgia, April 27th 2002, chapter 8, page 14.

[20] SoundBridge, "Voltage Controlled Amplifier (VCA) ," https://soundbridge.io/voltage-controlled-amplifier-explained/, [Online; accessed 8-August-2020].

[21] P. D. Marshall and D. K. Sidorov, "CM3106 Multimedia," School of Computer Science and Informatics Cardiff University, UK.

[22] A. Chipperfield, P. Fleming, H. Pohlheim, and C. Fonseca, "Genetic algorithm toolbox for use with matlab," 1994.

[23] G. Liu and J. Chen, "The application of genetic algorithm based on matlab in function optimization," in *2011 International Conference on Electrical and Control Engineering*, 2011, pp. 5034–5037.

[24] H. C. MathWorks, "Genetic Algorithm Terminology," https://es.mathworks.com/help/gads/some-genetic-algorithm-terminology.html, [Online; accessed 3-August-2020].

[25] J. Biles *et al.*, "Genjam: A genetic algorithm for generating jazz solos," in *ICMC*, vol. 94, 1994, pp. 131–137.

[26] D. Vrajitoru, "Crossover improvement for the genetic algorithm in information retrieval. ," PII: S0306-4573(98)00015-6,1998 Elsevier Science Ltd.

[27] K. A. Stenberg, "FM synthesizer for real time use from Csound," Bachelor Thesis, University of Alicante, September 2013, chapter 2, pages 33-34.

[28] M. J. Yee-King, "Automatic sound synthesizer programming: techniques and applications," Ph.D. dissertation, University of Sussex, 2011.

[29] H. C. MathWorks, "Genetic Algorithm Documentation," https://www.mathworks.com/help/gads/ga.html?, [Online; accessed 23-July-2020].

[30] A. Chipperfield and P. Fleming, "The matlab genetic algorithm toolbox," 1995.

[31] M. Macret, P. Pasquier, and T. Smyth, "Automatic calibration of modified fm synthesis to harmonic sounds using genetic algorithms," *Proceedings of the 9th Sound and Music Computing conference (SMC 2012)*, pp. 387–394, 07 2012.

[32] F. Eichas, "System identification of nonlinear audio circuits," Ph.D. dissertation, 2020.

[33] D. Ellis, "Musical Instruments ," http://www.ee.columbia.edu/~dpwe/sounds/instruments/, [Online; accessed 13-August-2020].

[34] J.-P. Groby, "Introduction to psychoacoustics and psychoacoustic tests, by kristian jambrosic (university of zagreb)," 2018.

[35] Biamp, "Sound Masking 101 ," https://cambridgesound.com/learn/sound-masking-101/, [Online; accessed 15-August-2020].