SYNTHESIS CHALLENGE 2022

Myriam Xueying Grandal Alonso

Synthesis Challenge

Index

Figures

Abstract

In the first part of the project, it was required to synthesize a electric vehicle sound given a set of parameters like RPM, speed, acceleration, steer wheel . Hence, given the motion data, it is possible to create a suitable sinusoid, filling it with specific amount of samples in order to get the proper sound. Therefore, we will apply a relation between the frequency and the speed; including a constant, to create a direct proportional relation among them. With this purpose, whenever the car increases its speed, the frequency will increase. To create a neat sound, it is necessary to smooth the signal applying some filtration like LPF and HPF setting the correct parameters to generate the most suitable sound and a suitable normalization. In my case, it was also included a BIP sound to detect the steer wheel crossing to 0.

On the second part of the project, a musical instrument synthesis has been developed selecting “Barcarrolle” MIDI file and applying some specific techniques like Additive and Formant synthesis, to obtain the final sound. For that purpose, it has been analysed the “main.m” file in order to obtain the different instruments and percussion sounds included on the song. By implementing an external function called “Analyza.m”(spectral analysis), it was possible to obtain the envelope with the harmonics of the sample used (‘Cmajorpiano.wav’) and create the piano sound by applying additive synthesis.

For the percussion (ensemble timbres) it was decided to apply formant synthesis to an ensemble of violin.

Introduction

Regarding the synthesis challenge, there are two main tasks that are required to be fulfilled:

- Synthesize sound of the electric vehicle Superb by Škoda according to car motion control.
- Musical instrument synthesis of Barcarolle composed by 2 different instruments:

  -Grand Piano and String Ensemble 1 (different notes).

In order to develop the tasks, it has been given some MIDI files to work with in MATLAB environment:

-"main.m": includes synthesis of a single instrument as well as a polyphonic synthesis.

-"synth.m": where the realization of the synthesis is developed and the functions will be implemented.

-"speed.m": where the realization of the car synthesis is developed and the different sounds will be generated.

-**control signals.txt:** where the definition of the parameters for animator controller are described. Some of them are: speed, RPM, steer wheel, acceleration. The vector with the motion variables are given, especially, the speed and RPM are the ones that defines better the changes during the car trajectory.

-**Car motion video:** For better understanding of the control signals parameters, a simulation of the car motion is supplied. There, it is provided the car movements, changes of direction and steer wheel movements.

Moreover, regarding the instrument synthesis, some MATLAB functions have been provided: "getTempoChanges.m","readmidi.m","writemidi.m".

In order to get the information about the MIDI file, specifically Barcarolle, it was used the function "main.m" that it is able to specify the instruments that appear in the song:

-**Channel 1**: instrument nº 49 (129 notes). /**String ensemble 1.**

-**Channel 4:** instrument nº 49 (162 notes). /**String ensemble 1.**

-**Channel 7:** instrument nº1 (1238 notes). / **Acoustic Grand Piano.**

Synthesis Challenge Methodology.

TASK 1: Synthesize sound of the electric vehicle Superb by Skoda.

For this task, there were given several parameters that described the car motion like the speed, RPM and acceleration. Hence, the first step was to analyse the data, important to synthesize the car sound.

For this purpose, the RPM vector was read from the "Control.signals.txt" file. Specifically, is the column 10 from the table. As it was required to generate an electric car sound, the RPM signal was smoothed in order to get rid of the irregular peaks of the original raw data. The RPM signal has the following shape:

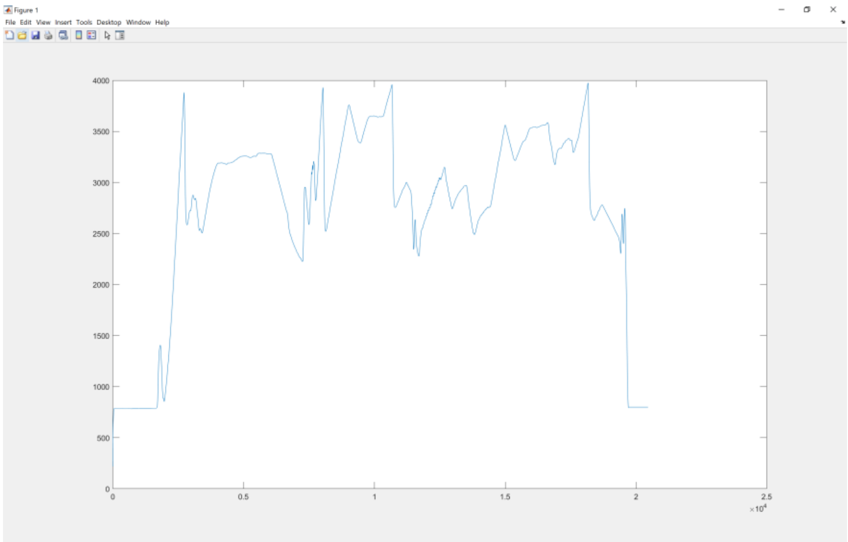

Figure 1: RPM smooth signal

The smoothing, its done by taking a specific window of samples and make the mean of the current and previous sample in order to do a proper averaging. Specifically, it was set to 50.

In order to do the car synthesis, it is necessary to generate a sinusoidal signal with different samples inside it. Hence, as it is known from the data given, each parameter (speed, RPM, acceleration) is taken every 4ms. Consequently, the duration and the sampling frequency (10.000Hz) are set.

In order to fill up the sinusoid, the *note_dur* variable is calculated by multiplying the sampling frequency (10.000Hz) by the duration(4ms) obtaining the length of the signal.

To implement the samples inside the sinusoid (phi), two loops are created. The first one, allows to convert the RPM raw data into frequency by the following expression:

$$f_i = f_0 + round(\frac{S_{avg(i)}}{c})$$

By this formula, every time the cars increases its speed, the frequency will increase. Hence, higher frequencies will be obtained with higher speed. The fc parameter is the initial frequency and it was set to 10Hz, if you increase it, the tone will be higher. Moreover, each RPM is divided by a constant (c=30Hz), in order to obtain the fundamental frequency; corresponding to a full cycle of the engine. *(Jan Jagla, 2012*)

Moreover, it is necessary to add the temporal difference between samples (dp) which is calculated by the following expression:

$$dp = \frac{f_i * 2 * \pi}{fs}$$

The final step is to include this samples inside the sinusoid. Therefore, it is necessary to apply the second for loop that its delimited by the *note_dur* and it is able to fill the final signal.

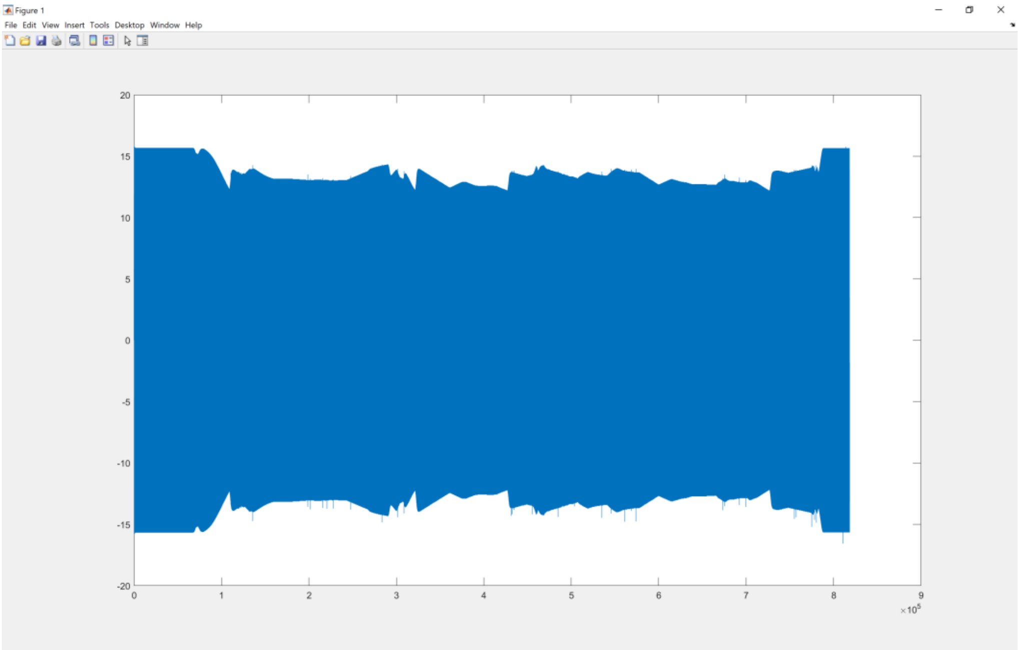

Figure 2: Final sinusoide

Moreover, in my particular case, the steer wheel vector was used to provide a BIP smoothed sound each time the wheel crosses to 0. This is possible because the data is given by angles.

Therefore, different conditions were set up. Firstly, a vector of zeros is set and the conditions are established. The BIP flag is activated(set to 1) when the steer wheel crosses through zero and this happens in three specific occasions:

- When the angle changes from positive to negative.(Turning right).
- When the angle changes from negative to positive. (Turning left).
- When the angle changes from a value different from 0 to 0. (Turning to the centre).

To generate the BIP sound, a random noise was generated. To avoid a electric sound, that would be so aggressive to the driver, a gaussian window is applied to smooth the random noise and provide a soft noise to advise the driver about the steer wheel changes.

In order to fit correctly the sinusoid car sound, the same size was given to the BIP.

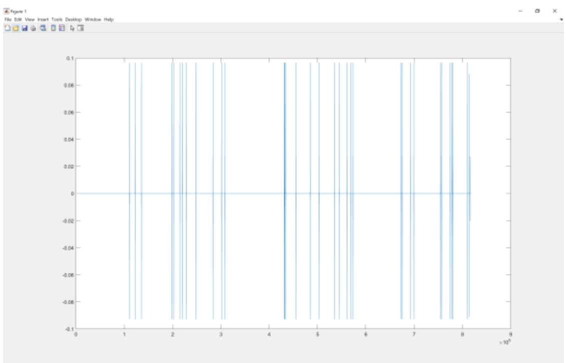

Figure 3:BIP steer wheel changes.

To continue with, this generated BIP was added to the car sound synthesised sound.

Finally, as the final sound was too loud, it was necessary to attenuate the high frequencies to sound softer by implementing a specific High pass and Low pass filtration (pass band filtration). For this purpose, it is necessary to set the proper coefficients of the High pass and low pass filter to smooth the sound. Specifically, on the low pass filter, the coefficients were set to a=[1 1] and b[1 -0.9]. On the other hand, the High pass filtration was done with the following values a=[1 0.9] , b=[1, -1]. It was possible to smooth it more or less but the suitable sound was got with this specific values.

TASK 2: Musical instruments synthesis of Barcarolle MIDI file.

In the Barcarolle midi file appear 2 different instruments: Acoustic Grand Piano and String Ensemble with different amount of notes.

To synthesize the musical instruments, two different techniques were applied:

- **Additive Synthesis:** It is an original spectrum modelling technique. Based on Fourier's theorem, which states that any periodic function can be modelled as the sum of individual sinusoids. Each particular sinusoid will have its particular characteristics regarding the amplitude and harmonics frequencies. *(Matthias Robine, 2006).* Because of its flexibility and suitable results, is the technique selected to create the piano sound. (Mantione, 2017)
- **Formant Synthesis:** It is one of the most popular methods based on the source-filter-model. In order to produce a high quality sound, normally, it is required to used from three to five formants. They are fixed frequency peaks over a variety of frequencies. Therefore, in order to generate a proper instrument sound, it has been applied this synthesis to obtain complex sounds synthesising a violin sound.

➢ Acoustic Grand Piano:

In order to synthesize the piano sound, additive synthesis was applied. Specifically, the coefficients were taken from a real piano sample ('Cmajorpiano.wav'). For this purpose, the spectral information of the sample was taken from the function "analyza.m" applying spectral analysis. This function, allow us to obtain the harmonics within its respective amplitudes as seen in the following figure:

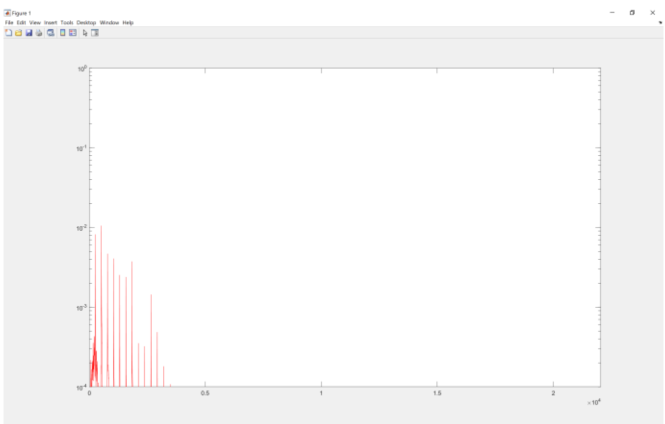

Figure 4:Envelope extraction of the sample to take the amplitude and harmonics

To continue with, it is necessary to create a vector, containing the coefficients amplitude. Hence, the synthesized piano sound will be a sinusoidal signal with the correspondent harmonics. After implementing the harmonics with the fundamental frequency, an exponential envelope is applied. For a simplification of the signal, a normalization was also applied.

In the next figure, shows the sample used for the synthesis in temporal domain('Cmajorpiano.wav'):

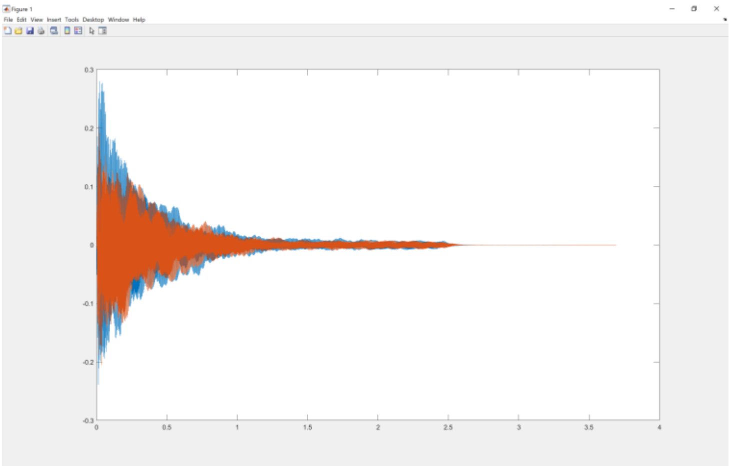

Figure 5: 'Cmajorpiano.waw' exponential envelope

As it has exponential characterization, the previous technique was applied(exponential envelope).

- **String ensemble:**

The string ensemble 1 & 2 were substituted by violin sound, implementing formant synthesis. For this purpose, it is necessary to set the different parameters like fundamental frequency and the duration. Moreover, it will be added a certain vibrato to each harmonic. As a result, the initial(raw) signal will be given a certain shape according to the resonator.

The next step is to set up the resonator coefficients(B and C). In order to do a suitable setting, a filter within the violin model characteristics is provided. In particular, the setting parameters are the frequencies and BW of the violin to obtain the formant of the violin.

Once the signal has been filtered, an ADSR envelope is applied to smooth the resulted signal. This envelope is controlled by 4 parameters (attack, decay, sustain and release).

Finally, the interpolation(*'interp1'*) function is applied to select the specific position of the envelope and a normalization is done afterwards to simplify and adjust the resulting signal.

Results

The car synthesis was done by analysing the set of parameters defining the car motion, and in my particular case, was done by RPM vector. By filling up the sinusoid with the correct samples, doing the conversion of RPM to frequency along the signal, the car sound was obtained. Specifically, it is included in the result folder with the name "car.m4a". It is possible to improve the sound by applying amplitude modulation, including the acceleration variable and making the frequency change and increase while it accelerates, but the amplitude decreases and just the peaks of acceleration are perceived.

As the requirements asked for the car audio in m4a format, it was necessary to apply a resample as the sampling frequency was set to 10kHz and the synthesis was made for that specific value. Hence, resampling was done using "*resample*" function of MATLAB. This was necessary because by default, the only sampling frequencies allowed for m4a are 44100 and 48000Hz. Moreover, the result is also available in .wav and it is important to notice that in m4a format it is better to hear the audio with headphones.

The instrumental synthesis of the Barcarolle MIDI, it is stored in the result folder with the following name: "Barcarolle.m4a". The initial Barcarolle file was set with a different piano note (C major) and the ensemble string was simple.

In order to improve the sound, a different piano note was used as a sample and the result is the piano notes have a higher tone than the original one. Moreover, the string ensemble was replaced by a violin, hence, the result makes the sound more unified and complex than the original one.

Conclusion

It was possible to realize all the different tasks by applying different techniques like filtration (LPF,HPF, gaussian windowing) additive synthesis and formant synthesis.

In the case of the car synthesis, it was possible to create a suitable electric car sound adding few extra sounds like the BIP to advise the driver the changes on the steer wheel. Moreover, to attenuate the final sound (high frequencies) and reduce the volume, a band pass filter was applied, selecting the correct parameters to smooth the filtering.

The final car sound is correct but it could be improved by for example, applying an amplitude modulation to change the amplitude within the acceleration and make more noticeable the changes this parameter. Moreover, to create a more musical sound, it would be possible to apply a *sawtooth* signal instead of the sinus, and adjust it to provide a more decorated sound.

According to the musical instrument synthesis, two specific techniques were selected (additive and formant synthesis) and provided suitable results according to the original midi file. The string ensemble was replaced by violin sound and it makes the instrumental sound more natural and more complex.

To conclude, I am quite happy with the results obtained because it made me apply the knowledge acquired during the seminars and explore the synthesis of audio field which personally, I find really interesting and enriching; as well as so complex.