Summer 2016

Engine sound simulation and generation in driving simulator

Shuang Wu

Department:

ENGINE SOUND SIMULATION AND GENERATION IN DRIVING SIMULATOR

by

SHUANG WU

A THESIS

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

2016

Approved by

Ming C. Leu, Advisor

Robert G. Landers

Suzanna Long

ABSTRACT

Simulating a driving environment that provides engine sound as auditory feedback to the driver is a challenging task due to the complexity of engine sound, a wide range of operating speeds, and repeated sound clicking during playback. This thesis describes a method of engine sound simulation and generation for a driving simulator. By analyzing sample sounds, spectral modeling synthesis was used to decompose the sound samples into deterministic and stochastic components. Then, the modeled deterministic and stochastic signals were employed to resynthesize the sounds. To represent engine sounds not available in the recorded database, sound interpolation was applied to two neighboring engine sounds. In addition, a cancelation method was developed to remove the repeated clicking that occurs when engine sounds are played in a loop. Finally, a powertrain model was used to calculate the engine speed. As seen by comparing the spectrograms of the sounds, the resynthesized sounds exhibited high similarity to the recorded sample sounds. Also, the spectrogram of the sound generated by interpolating two sample sounds was found to agree fairly well with the actual engine sound.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude towards my advisor, Dr. Ming C. Leu, for his support, advice and guidance throughout my graduate studies. I would like to thank Dr. Suzanna Long for her constant help and patient advice in this research project. I would also like to thank Dr. Landers Roberts for spending his time and effort as my committee members and helping me through the course of studies.

I would like to thank Dr. Chandrashekhara, K. and Samaranayake, V. A., who have helped me greatly during my course study, project work and research work. I would like to thank Dr. Appelman, Howard and Dr. Birman, Victor for their help through the course of studies. I would like to thank all my research mates in VR&AM lab for their help during my graduate study.

Finally, I am deeply indebted to my parents for their support, encouragement and all my friends who stood by me during the entire duration of my studies at Missouri University of Science and Technology.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Page

LIST OF TABLES

1. INTRODUCTION

Simulating a driving environment may be advantageous by providing auditory feedback from the simulator to the driver. The audio signal gives an indication of the state of the vehicle, which can be of great importance for vehicle handling and performance improvement purposes [1]. Previous simulator studies have shown that certain outputs, such as low-frequency sound [2] and combinations of sound and vibrations [3], can affect driver behavior.

Engine sounds are generally not as loud as road noise while driving; however, engine sound tend to be more interactive, and drivers tend to more easily notice engine sounds. In many instances, such as when operating a heavy truck, drivers tend to use audio cues to decide when to change gears.

Moreover, engine sound is strongly correlated with vehicle speed and influences the perception of speed. A driving simulator study conducted by Denjean [4] indicated that the engine speed information provided by engine sound is very important for the task of maintaining a steady vehicle speed. Drivers often underestimated their speed to a greater extent in a driving simulator and tended to drive faster when no engine noise feedback was present compared with conditions in which engine sound feedback was present. Other studies [5, 6] have also demonstrated the underestimation of vehicle speed and the difficulties experienced by drivers in maintaining a target speed when acoustic feedback is absent.

To generate engine sound in driving simulators, it is common to represent the sound from a vehicle engine by snippets of recorded sound played in a loop. Snippets recorded at a speed of revolution close to the current virtual speed of revolution of the engine are continuously chosen. For engine sounds that were not available in a database, Blommer [7] used pitch bending and sound level mixing via a "constant power" method to simulate these sounds. In simulating engine sounds, Heitbrink [8] used a sample-based synthesis approach, also known as a wavetable approach. During playback, crossfading was applied between two sample sounds whose frequencies are shifted to represent the unavailable engine sound.

Although playing snippets of recorded engine sounds in a loop is a valid method for representing engine sounds, such a method suffers from a certain limitation that may become problematic. The onset and offset of a sound need to coincide with sound wave boundaries. For complex engine sounds, natural boundaries may be difficult to identify. Consequently, the onset and offset of each sound will be audibly identifiable as repeated clicking. To hide such clicks, the sound snippets must be slowly faded in and out.

This thesis study performed the simulation of engine sounds by use of spectral modeling synthesis (SMS) and sound transformation. In terms of sound synthesis, the deterministic plus stochastic model was used to reproduce sample sounds. A cancelation method was proposed to remove the repeated clicks produced when engine sounds are played in a loop. For the simulation of engine sounds that are unavailable in the database, sound interpolation was used by modifying the amplitude and frequency of a sound

spectrum. Finally, a vehicle powertrain was introduced to calculate the engine speed for the sound generation.

The thesis is presented as follows. Section 2 introduces the sound acquisition process and analysis. Section 3 explains the sound modeling and synthesis approach. Section 4 presents the methods used for sound transformation and modification. Section 5 introduces a powertrain model for the driving simulator. Section 6 provides the conclusions of the study.

2. SOUND ACQUISITION AND ANALYSIS

To simulate engine sound in a driving simulator, it is necessary to record sample sounds for subsequent sound analysis, modeling and synthesis. The engine sound may be recorded in a vehicle mounted on a dynamometer. However, a dynamometer was not available for our recording session; instead, all engine sounds were recorded in a stationary vehicle in neutral. Five engine sounds were recorded at intervals of 1000 RPM, from idle to 4000 RPM, using a pair of in-ear microphones and a digital recorder (figure 2.1)

Figure 2.1. In-ear microphone and digital recorder

To determine the characteristics of the engine sounds for later sound rendering, the captured sound samples were analyzed in the frequency domain.

2.1. SOUND SOURCE ANALYSIS

Table 2.1 shows the relationship between each vibration frequency and its source. For the six-cylinder gasoline engine in the vehicle considered in this study, the sound sources can be divided into the following categories:

1. The noise caused by fuel combustion in the engine cylinders, which depends on the pressure pulses produced by combustion. The engine pulse frequency, which is proportional to the engine rotation speed, is calculated as follows:

$$F_0 = (number\ of\ the\ cylinder) \times \frac{RPM}{120} \qquad (1)$$

   where RPM is the number of revolutions per minute of the engine. However, the intensities of the engine pulses in each cylinder are not uniform, giving rise to harmonic components at multiples of one-sixth of F0.

2. Mechanical noise caused by the revolution of the crankshaft and the vibration of solid structure. For a four-stroke engine, the fuel ignites once every two crankshaft revolutions for each cylinder. Thus, for a six-cylinder engine, the crankshaft revolution frequency is one-third of F0.

3. Aerodynamic noise caused by engine cooling and alternator fans. This type of noise is strongly correlated with combustion because the sound sources are driven by the engine crankshaft. The component frequency depends on the pulley ratio and therefore is not necessarily a multiple of F0.

Table 2.1 Relationship between each vibration frequency and its source

Component	Frequency
Pressure pulses produced by engine	F_0 *and multiples of* $\frac{1}{6}F_0$
Crankshaft	$\frac{1}{3}F_0$
Engine cooling and alternator fans	Dependent on the pulley ratio

2.2. THE DFT AND STFT

The Discrete Fourier Transform (DFT) analyzes a time-domain signal X[n] to determine that signal's frequency content X[k]. The general form of this transform is

$$\mathrm{X[k]} = \sum_{n=0}^{N-1} x[n]e^{-jn\omega_k} \quad k = 0, \ldots, N-1 \qquad (2)$$

where n is the time index (normalized time, T=1), k is the discrete frequency index, and ω_k is the frequency in radians ($\omega_k = \frac{2\pi k}{N}$). The frequency calculated in Hz is expressed in the form

$$f_k = \frac{f_s \omega_k}{2\pi} \qquad (3)$$

where f_s is the sample rate.

DFT analysis assumes that the signal is stationary. However, most signals are non-stationary. Therefore, the short-time Fourier transform (STFT) is used to determine the sinusoidal frequency and phase of a signal as it changes over time. This transform's general form is

$$X_l[k] = \sum_{n=0}^{N-1} w[n]x[n + lH]e^{-jn\omega_k} \quad l = 0,1,2 \ldots \qquad (4)$$

where w is the analysis window, l is the frame number, and H is the hop size. The window functions that can be used in STFT analysis include Rectangular, Triangular, Kaiser-Bessel, Hamming, Hann and Blackman-Harris windows, among which the Hann window is the most commonly used in signal processing and the one chosen in this study.

2.3. SOUND ANALYSIS USING THE DFT

To gain a better understanding of engine sound characteristics at different frequencies, a sound analysis was conducted using the DFT in the frequency domain. The resulting power spectrum was used to obtain the characteristics of the engine noise, which can be represented in the form of the relative amplitudes at different frequencies. Because the engine noise of interest is a low-frequency signal, with much of it lying between 0 and 500 Hz, the analysis presented in this thesis will focus on this range of the frequency domain.

Figures 2.2-2.5 show the power spectra of the engine noise for 1000 RPM, 2000 RPM, 3000 RPM, and 4000 RPM engine speed. Several observations can be made as follows:

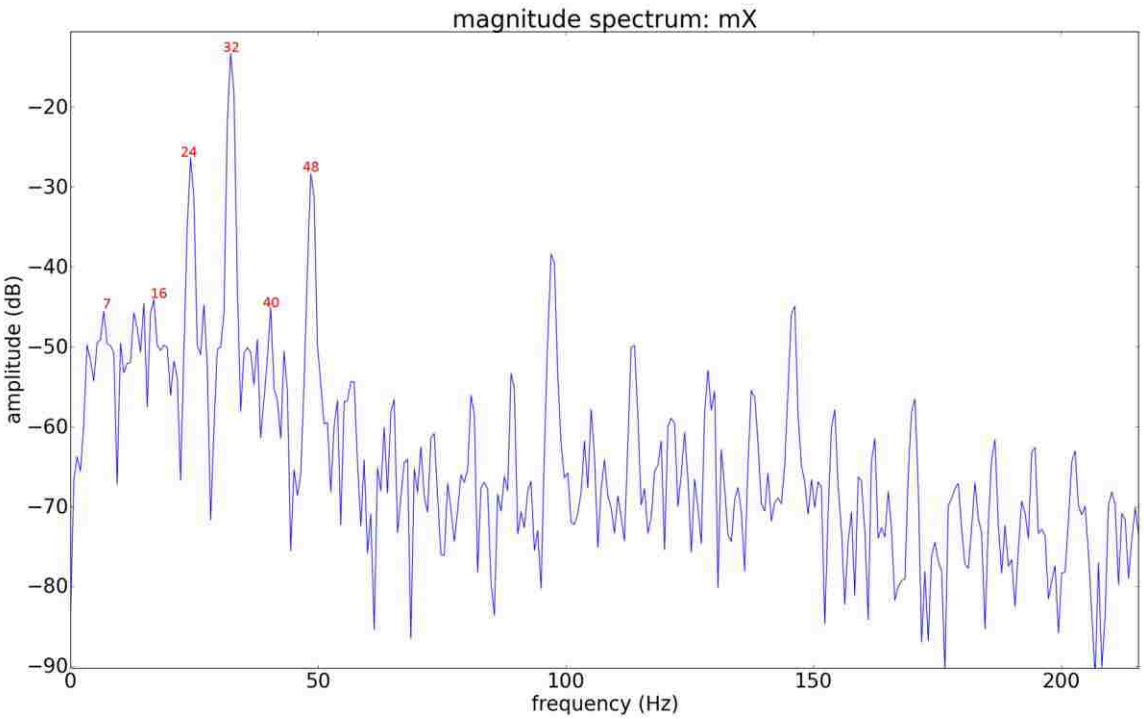


Figure 2.2. Power spectrum of the engine noise at a 1000 RPM engine speed

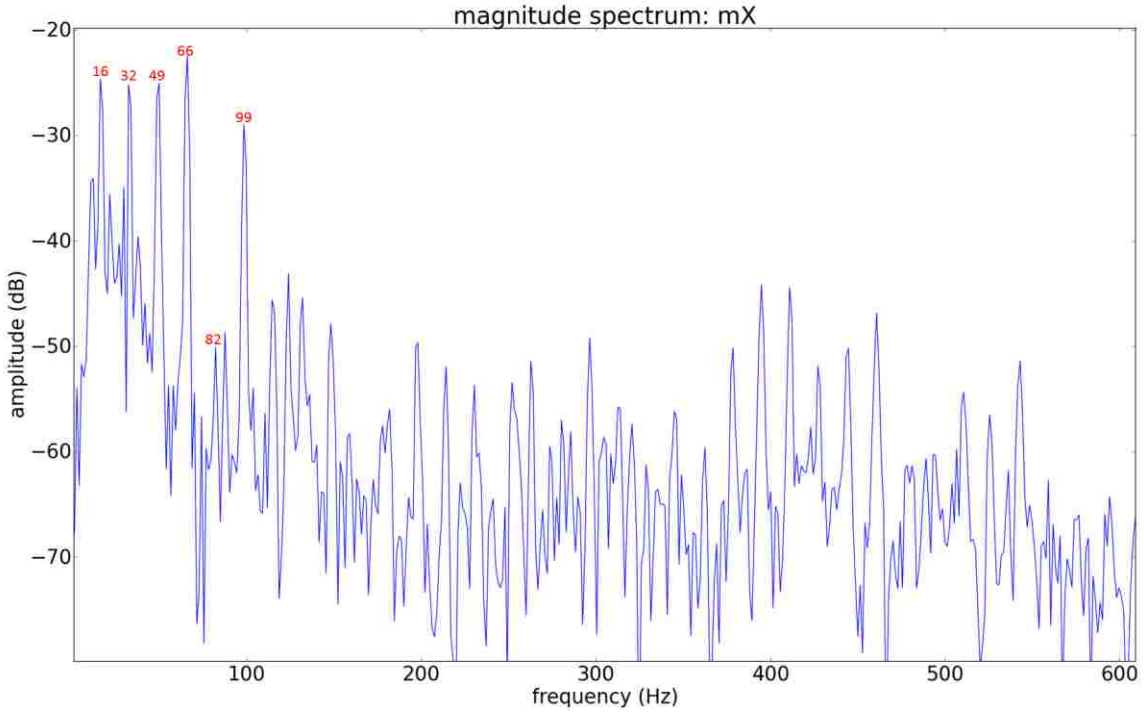


Figure 2.3. Power spectrum of the engine noise at a 2000 RPM engine speed

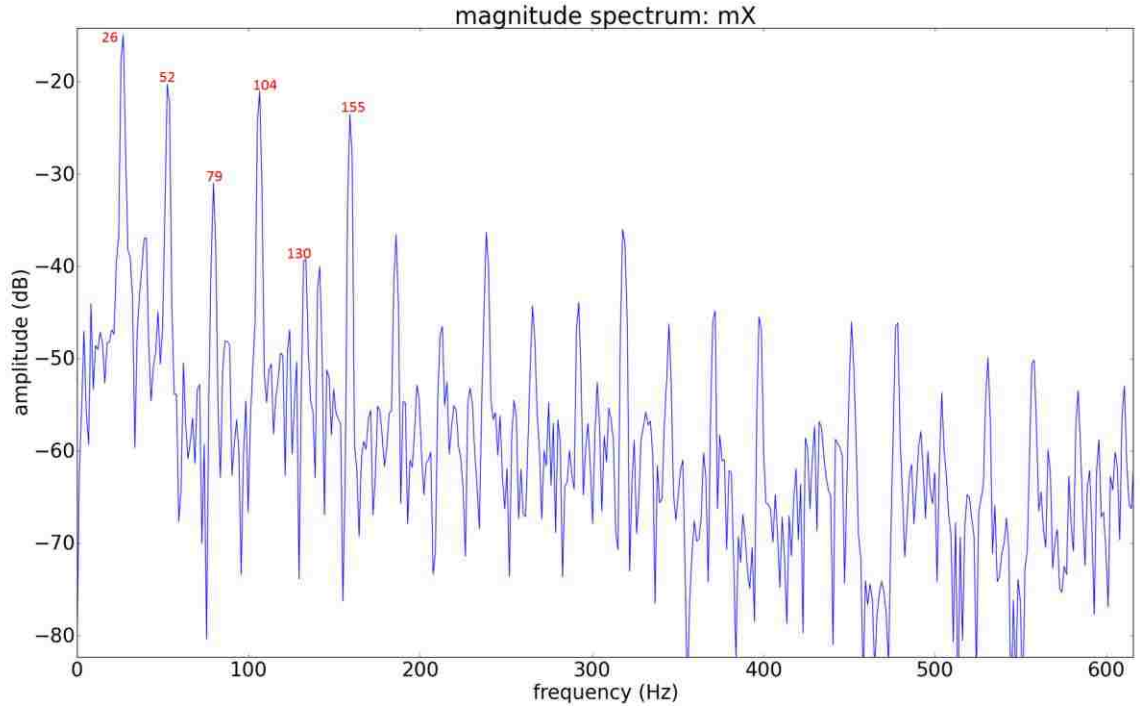


Figure 2.4. Power spectrum of the engine noise at a 3000 RPM engine speed

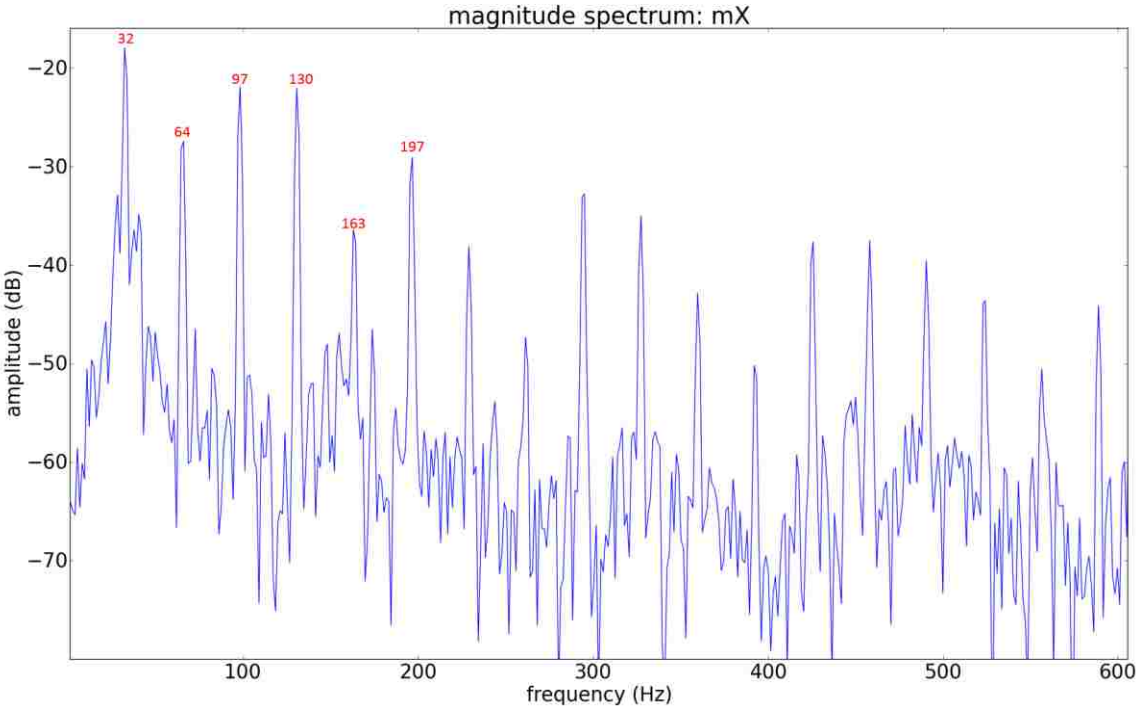


Figure 2.5. Power spectrum of the engine noise at a 4000 RPM engine speed

1. According to the power spectra, the frequencies of the 1st , 2nd and 6th major peaks marked in red numbers are close to $\frac{1}{6}F_0$, $\frac{1}{3}F_0$ and F_0. These three peaks primarily originate from the rotation of the crankshaft and combustion in the engine, as described above.
2. The 3rd , 4th and 5th peaks in the 1000,2000, 3000 and 4000 RPM spectra are the harmonic components arising from combustion in the engine and whose frequencies are approximately three times, four times and five times of $\frac{1}{6}F_0$.
3. The overall frequency of the engine sound harmonics are proportional to engine speed, and amplitude of the engine sound is also influenced by the engine speed, especially at frequencies higher than 300 Hz. A higher engine speed results in a larger amplitude of the engine sound. As the engine speed increases, the rotation and vibration of the mechanical structure increase significantly. However, the increase in the amplitude associated with combustion at lower frequencies is not significant.

2.4. SOUND SOURCE ANALYSIS USING THE STFT

Using the DFT to obtain the power spectrum in the frequency domain does not reveal any time-dependent information from the sound signal. Therefore, another method of analyzing sound is to use the STFT to calculate the spectrogram of the sound signal. Usually, the vertical and horizontal axes of this spectrogram represent the frequency and time, respectively, of the sound signal, and the color intensity at each point represents the amplitude of a particular frequency at a particular time.

Figure 2.6 shows the spectrogram of the engine sound recorded at 4000 RPM. As shown, the vertical and horizontal axes represent the frequency and time information, respectively, of the sound. The different colors in the image represent the amplitude of the engine sound; warm colors represent higher sound intensity, and cold colors represent lower sound intensity. The following observations can be made from this figure:

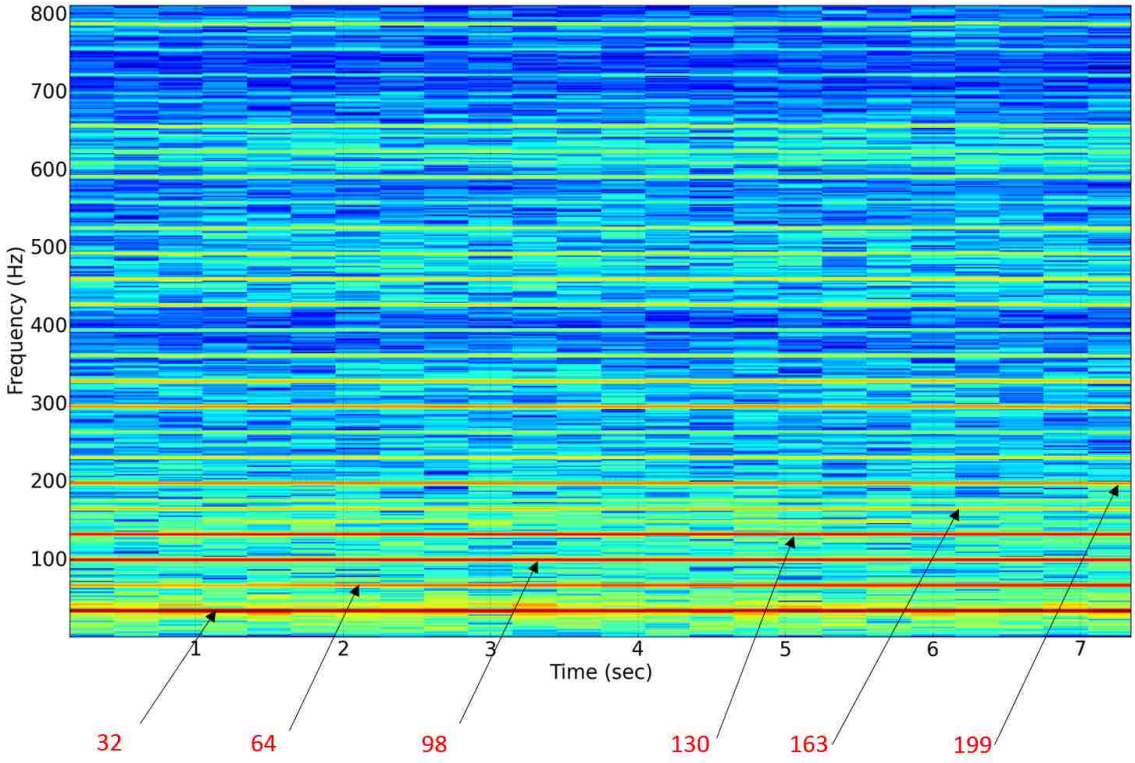


Figure 2.6. Spectrogram of the engine sound recorded at 4000 RPM

1. Because the sound sources correspond to different frequencies, the spectrogram exhibits color differences in the vertical direction. However, the distribution remains nearly constant in the horizontal direction, indicating that the sound signal was relatively stable for a period of 0-7 seconds. Thus, the engine speed during this time was also stable.

2. For frequencies in the 0-200 Hz range, the colors are warm because the related noises originate from both the engine pulses and the rotation of the mechanical struture. For frequencies higher than 400 Hz, the colors are relatively cold because the aerodynamic noise and enviromental noise are the dominant components in this frequency range.
3. The red lines near 32 Hz and 199 Hz represent the noise from engine pulses, and the red line at 64 Hz represents the noise from the rotation of the crankshaft. The two red lines at 98 Hz and 130 Hz are the harmonic components arising from the engine pulses. The frequencies of these red lines are similar to those of the peaks observed in the spectrum obtained through DFT analysis. Thus, the STFT results are consistent with the DFT results.

Figure 2.7-2.9 shows the spectrograms of the engine sound recorded at 1000 RPM, 2000 RPM and 3000 RPM; similar harmonics can be found from the spectrograms

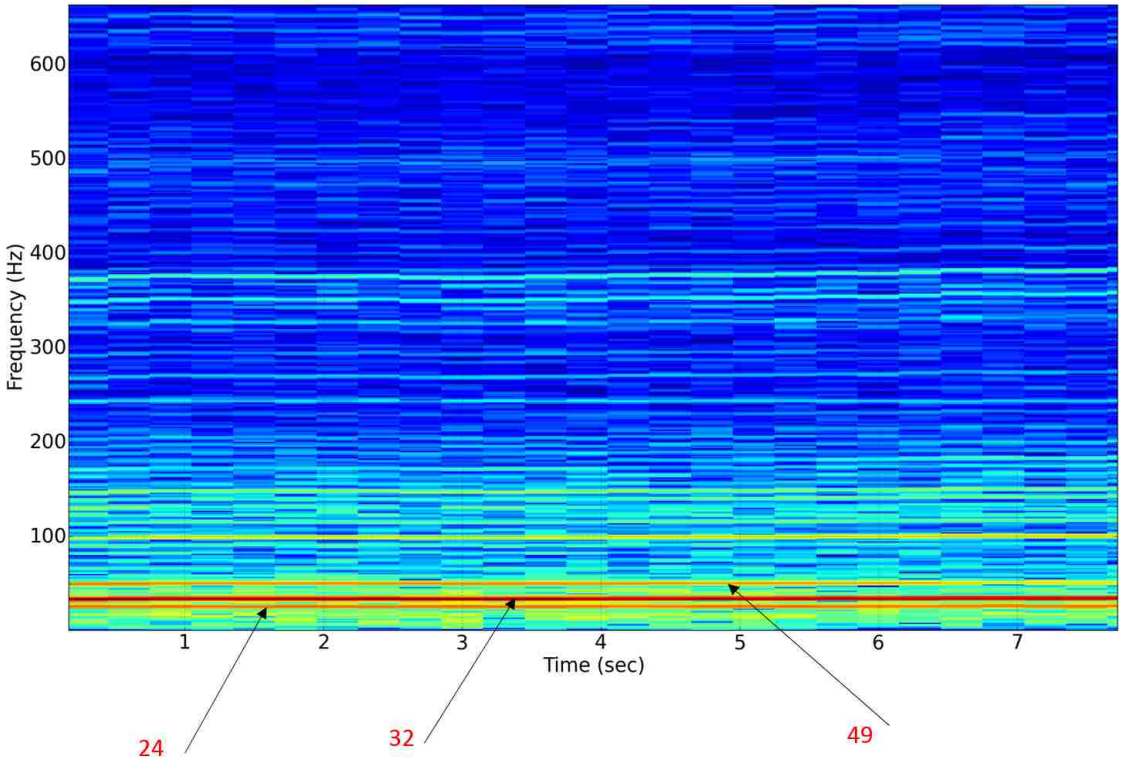


Figure 2.7. Spectrogram of the engine sound recorded at 1000 RPM

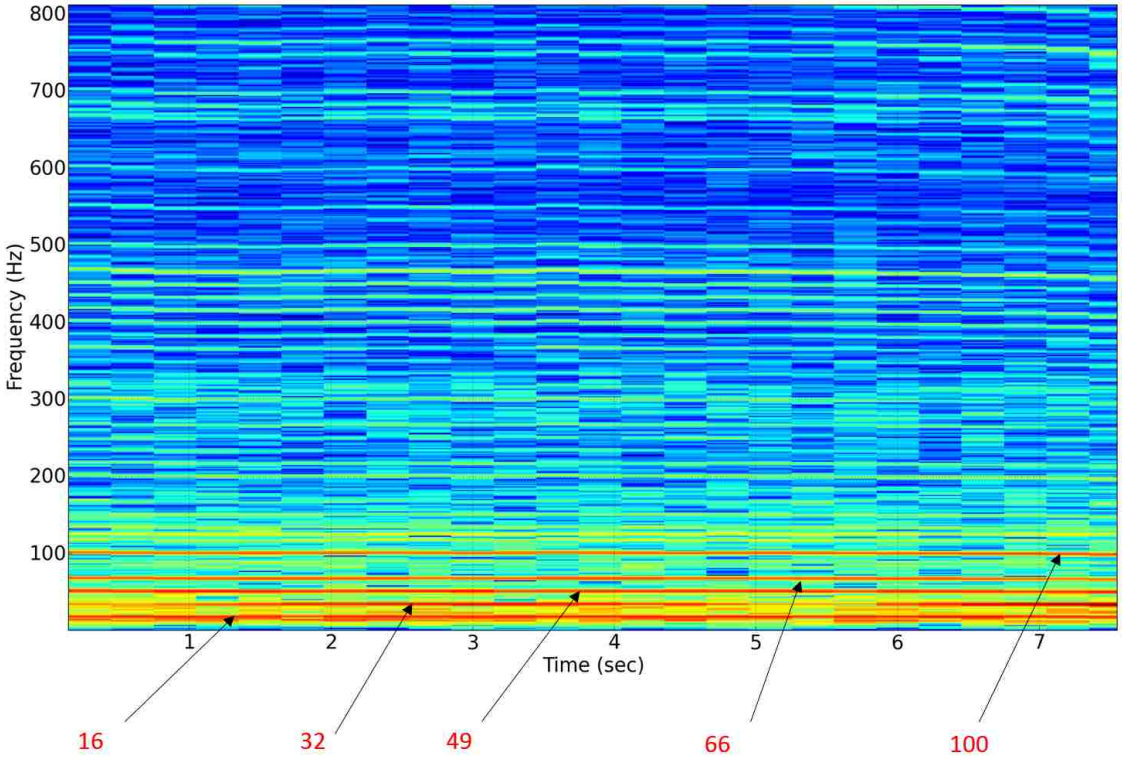


Figure 2.8. Spectrogram of the engine sound recorded at 2000 RPM

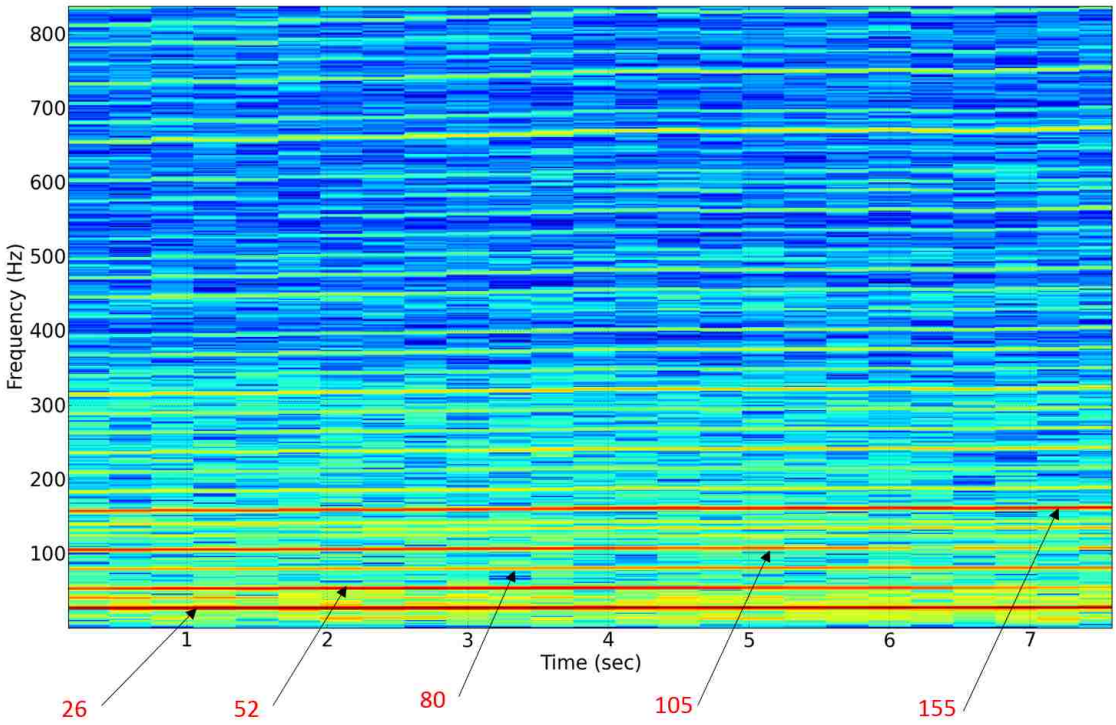


Figure 2.9. Spectrogram of the engine sound recorded at 3000 RPM

3. SOUND MODELING AND SYNTHESIS

Based on the knowledge of engine sound charateristics and the quantitative and qualitatve relations obtained in the previous chapter, sound models were developed and used to generate engine sounds for a driving simulation.

Every sound analysis and synthesis system is based on an underlying model. The sound produced by any physical system can be modeled as the sum of a set of sinusoids plus a residual. Because of the complexity of the engine sounds presented in the driving simulations, Spectral Modeling Synthesis (SMS) is applied to identify the sinusoids and residual. SMS is used to determine the mathematical models of the engine sounds at different engine speeds. The general form of SMS can be written as follows [10]:

$$\mathrm{s(t)} \approx \check{s}(\mathrm{t}) = \sum_{k=1}^{K} A_k \,\mathrm{sine}(\omega_k t + \theta_k) + e(t) \qquad (5)$$

where s(t) is the input sound; A_k, ω_k , and $\theta_k$are the instantaneous amplitude, frequency and phase, respectively, of the k^{th} sinusoid; and $e(t)$ is the noise component of the signal at time t (in seconds).

The deterministic component typically corresponds to the main harmonics of the sound. A deterministic signal is traditionally defined as anything that is not noise (i.e., an analytical signal, or perfectly predictable component, that can be predicted from measurements over any continuous interval). However, in this discussion, the class of deterministic signal considered is restricted to sums of time-varying sinusoids. Each

sinusoid models a narrowband component of the original sound and is described by an amplitude and a frequency function.

The residual consists of both the energy produced by the excitation mechanism that is not transformed by the system into stationary vibrations and any other energy component that is not sinusoidal in nature. The residual signal can be simply computed by subtracting the sinusoids from the original signal. Let r(t) denote the residual signal; it can be described as follows:

$$\mathrm{r(t)} = \mathrm{s(t)} - \sum_{k=1}^{K} A_k \, sine(\omega_k t + \theta_k) \qquad (6)$$

A stochastic signal is fully described by its power spectral density, which gives the expected signal power versus frequency. When a signal is assumed to be stochastic, it is not necessary to preserve either the instantaneous phase or the exact magnitude details of individual FFT frames. The stochastic signal is not merely the difference between the original signal and the sinusoids, it is the result of filtering white noise with an impulse response as an approximation of the residual signal. Let e(t) denote a stochastic signal; it can be described as filtered white noise as follows:

$$\mathrm{e(t)} = \int_0^t h(t,\tau)u(\tau)d\tau \qquad (7)$$

where u(t) is white noise and h(t,τ) is the response of a time-varying filter to an impulse at time t. The residual is thus modeled as the convolution of white noise with a time-varying frequency-shaping filter.

Figure 3.1 shows a block diagram of the spectral analysis. To synthesize the deterministic component, sinusoidal modeling is first applied to obtain sine frequencies, magnitudes, and phases. After the computation of the STFT for each windowed portion of the signal, a series of complex spectra are computed, from which the magnitude spectra are calculated. From each spectrum, the prominent peaks are detected, and the peak trajectories are obtained using a peak continuation algorithm. For the stochastic component, the residual signal is obtained by subtracting the deterministic components from the original signal. After applying the same window type and window size, the envelopes of the spectra are then approximated using a line-segment approximation. These envelopes form the stochastic representation.

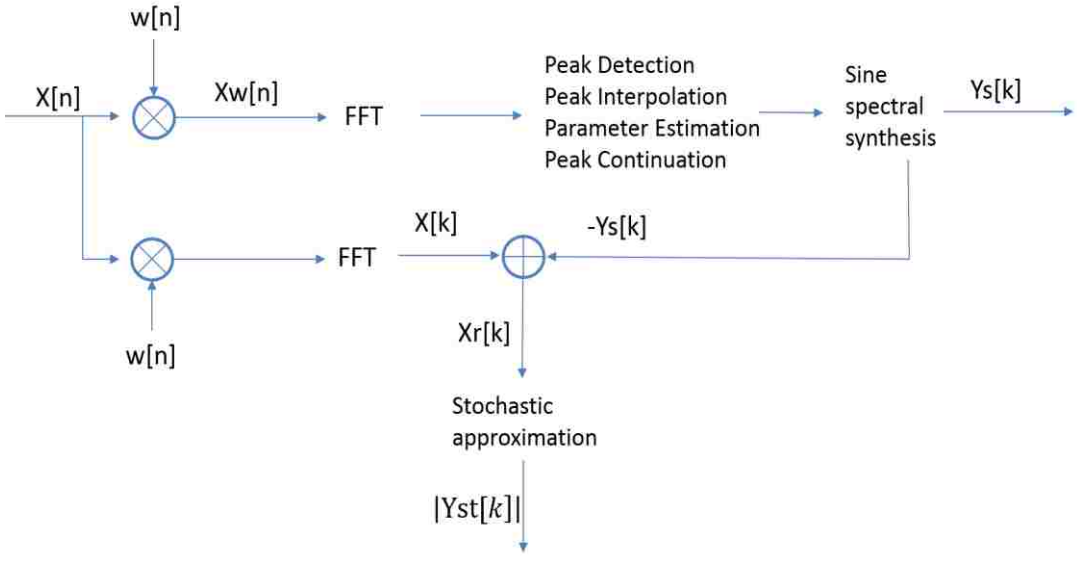


Figure 3.1. Block diagram of the spectral analysis

3.1 SINUSOIDAL MODELING

The sinusoidal modeling assumes that the most prominent peaks over time in the series of spectra produced by the STFT can be modeled by a series of time-varying sinusoids. Based on this model, a sound is generated by synthesizing a sine wave for each identified peak trajectory; the synthesized sound can be made to be perceptually equivalent to the original sound. Because most natural sounds do not have well-spaced and clearly defined peaks in their frequency domains, as in the case of the engine sound considered in this research, it is impossible to directly obtain these sinusoidal components. Practical solutions must be developed to find the sinusoidal components of input sounds.

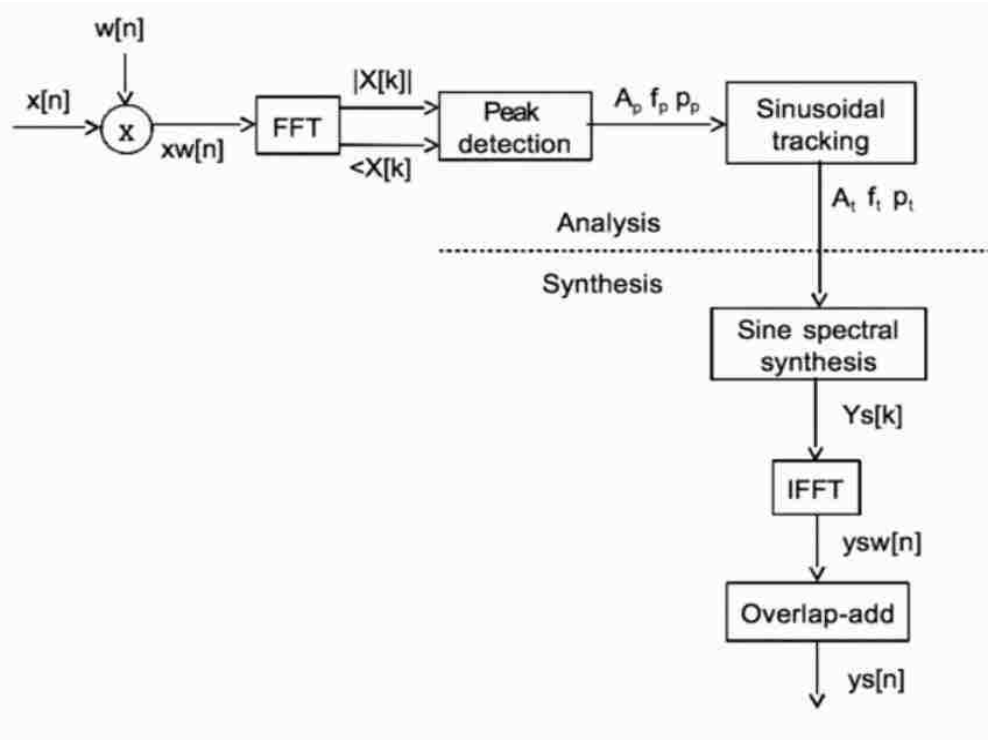


Figure 3.2. Block diagram of the sinusoidal modeling

Figure 3.2 shows a block diagram of the sinusoidal modeling method. First, the input signal X[n] is multiplied by the analysis window W[n], and then the FFT is computed to obtain the magnitude spectrum $|X[k]|$ and the phase spectrum$< X[k]$. Once the spectrum of the current frame has been computed, the next step is to detect its prominent peaks whose amplitude is above -70dB. To achieve better frequency resolution, zero padding and peak interpolation are applied during peak detection. Once the spectral peaks have been detected, sinusoidal tracking is applied to track those peaks over time and to find stable peak tracks in the spectrogram. Finally, the amplitude A_t, frequency f_t and phase p_t of each stable peak are estimated. Figure 3.3-3.4 illustrates the initial result of performing peak detection on the spectrum.

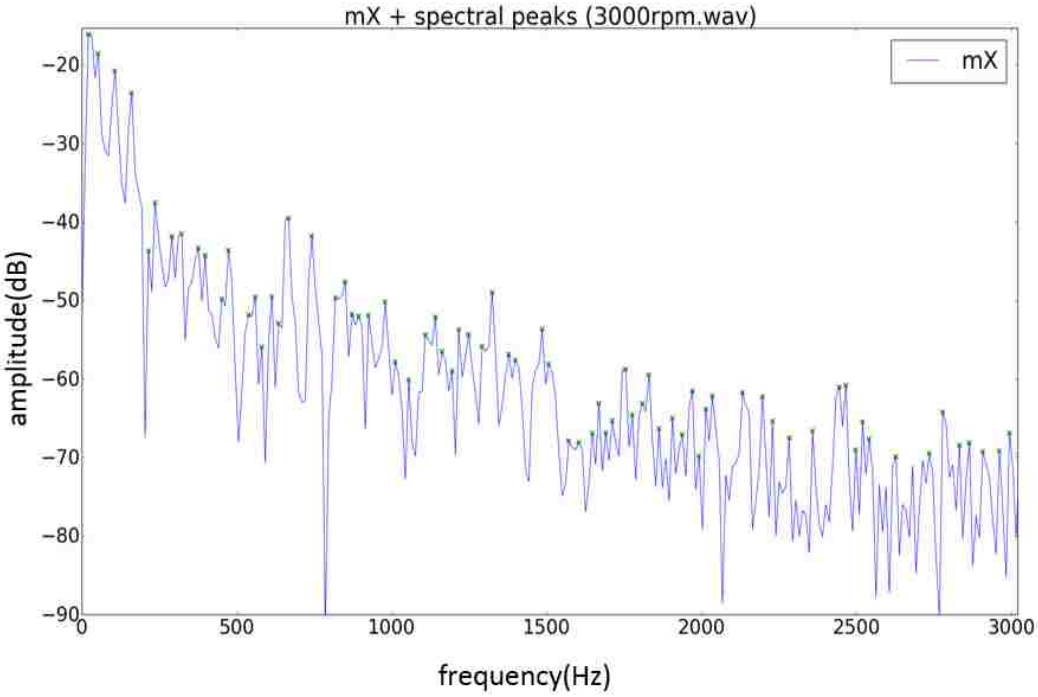


Figure 3.3. The peak detection on magnitude spectrum for one frame of the 3000 RPM engine sound

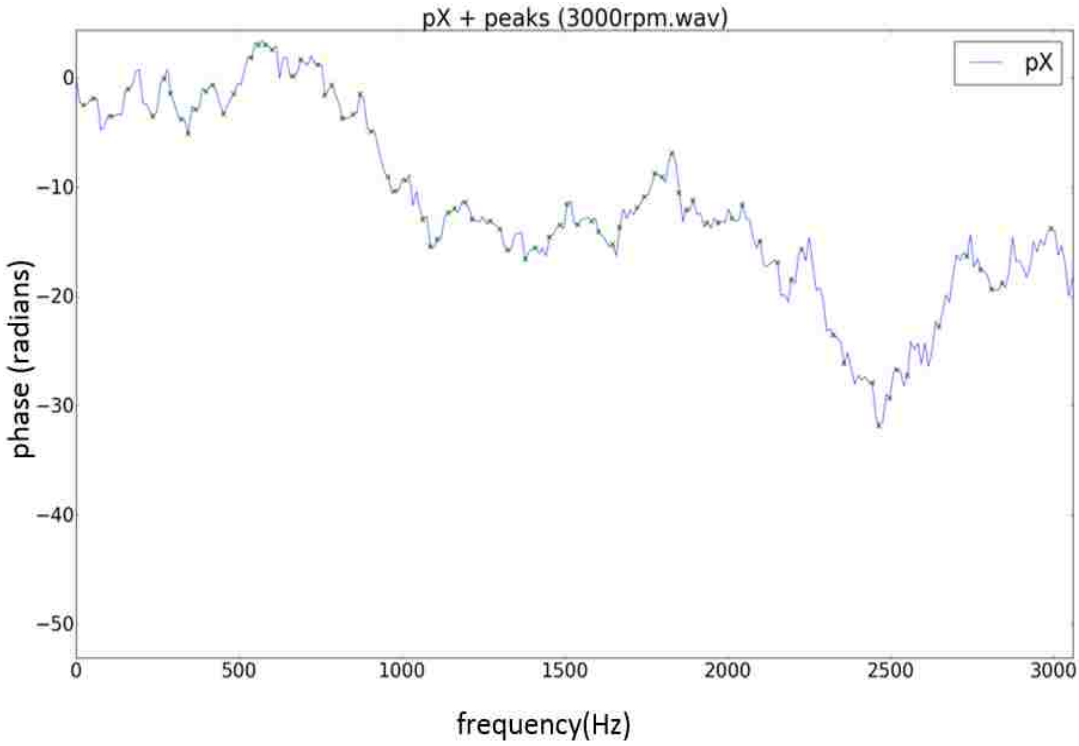


Figure 3.4. The peak detection on phase spectrum for one frame of the 3000 RPM engine sound

Based on the previously estimated sinusoidal parameters, sine spectral synthesis is applied to generate the main peaks in the spectral domain. Then, the inverse Fourier transform is computed to obtain the synthesized signal in the time domain and to remove the analysis window. Finally, the overlap-add process is applied to obtain the final synthesized sound, which reproduces the original sound relatively well. Figure 3.5 compares the result from the sinusoidal model with the original sound spectrum.

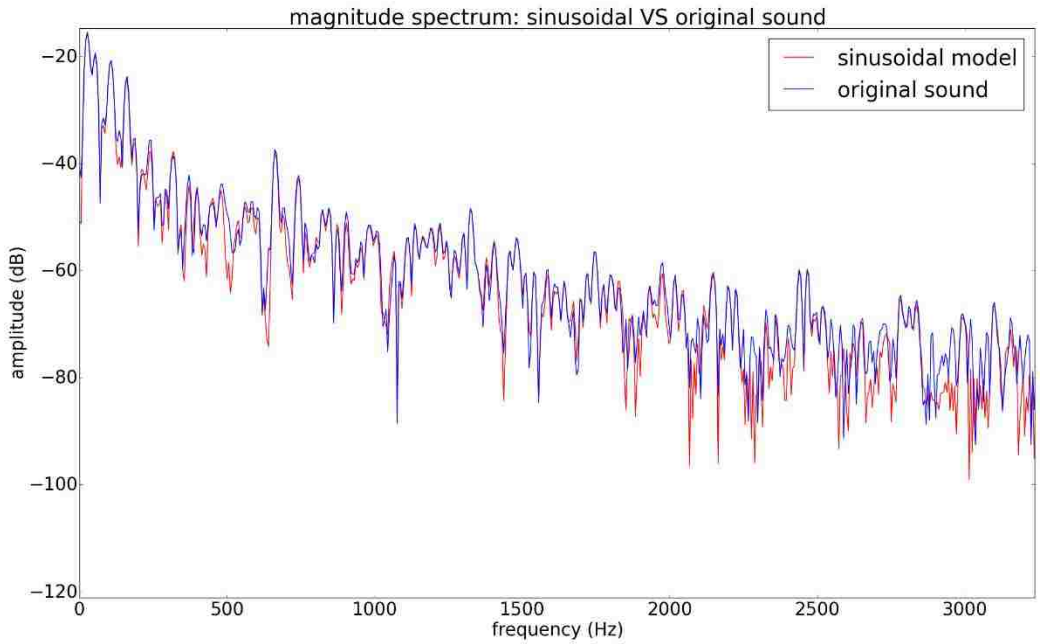


Figure 3.5 Comparison between the original sound and the sinusoidal model for 3000 RPM engine noise

3.2 STOCHASTIC MODELING

To calculate the stochastic component, we first obtain the magnitude-spectrum residual, which is done by subtracting the deterministic component from the original sound in either the time domain or the frequency domain. In this research, the deterministic subtraction was performed in the frequency domain using the same analysis window, window length, FFT size, and hop size for the STFT. Figure 3.6 shows the original sound spectrum, the sinusoidal component and the residual component.

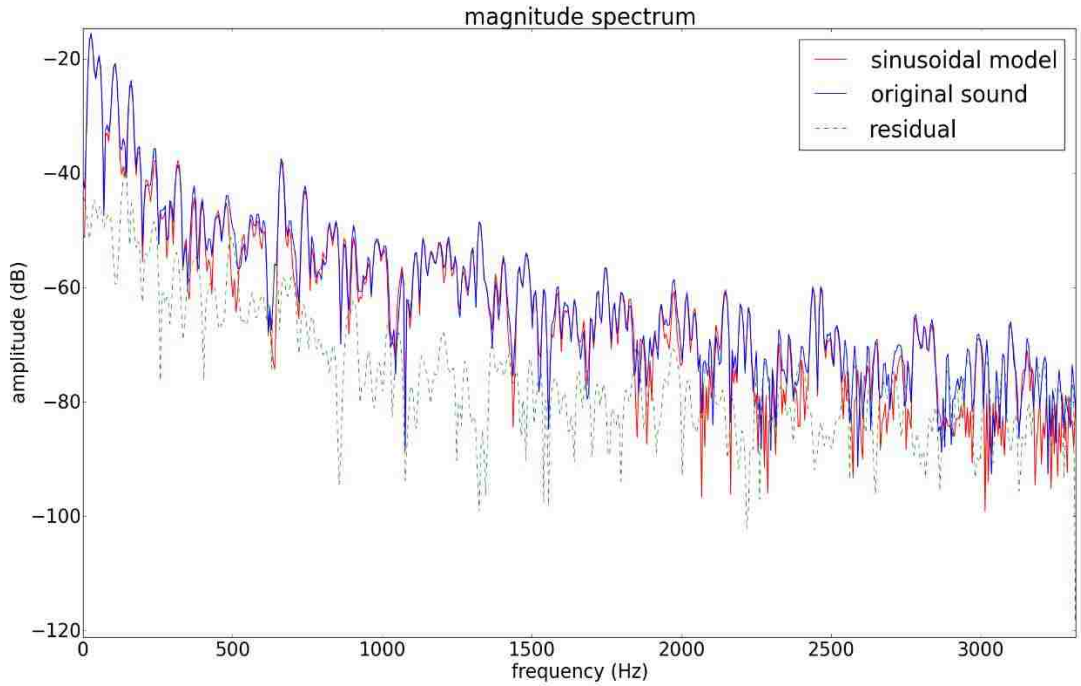


Figure 3.6 The original sound spectrum, the sinusoidal component and the residual component for 3000 RPM engine noise

After the deterministic subtraction, the stochastic approximation is applied to the residual spectrum to generate the stochastic component. The stochastic approximation assumes that the residual sound is a stochastic signal that can be fully described by its amplitude and its general frequency characteristics. Thus, it is unnecessary to retain either the instantaneous phase or the exact frequency information.

Every magnitude spectrum of the residual component can be approximated by its envelope because only its shape contributes to the sound characteristics. This type of problem is generally solved by performing curve fitting on the magnitude spectrum of the current frame [11]. In this research, a line-segment approximation [12] to the magnitude spectrum was sufficiently accurate and provided the desired flexibility. Figure 3.7 shows

an example of the result of the residual approximation for the sound recorded at 3000 RPM.

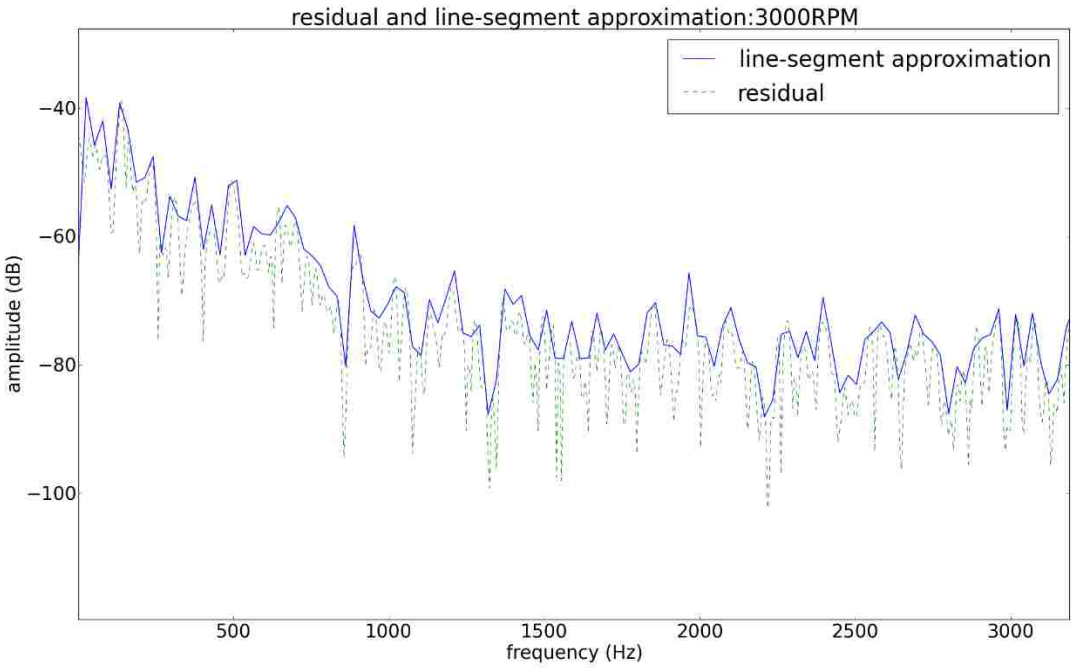


Figure 3.7 Line-segment approximation of the residual of the 3000 RPM engine sound

Compared with the sinusoidal modeling, the processing of the stochastic component is significantly simpler. Basically, the only parameters of the stochastic analysis that can be adjusted are the window length and the decimation factor.

3.3 SOUND SYNTHESIS AND SYNTHESIS RESULTS

After completion of the analysis presented above, the deterministic and stochastic components are combined to synthesize the sound. Figure 3.8 shows a block diagram of the spectral synthesis process: An additive synthesis approach is used to generate the

sinusoids Ys[k] in the frequency domain. The phase spectrum of the stochastic signal $< Yst[k]$ is created using a random number generator. The spectrum of the stochastic component Yst[k], is generated from the phase spectrum and the magnitude spectrum $|Yst[k]|$. This spectrum is summed with the spectrum of the sinusoids. Finally, the IFFT and the overlap-add process are applied to each frame.

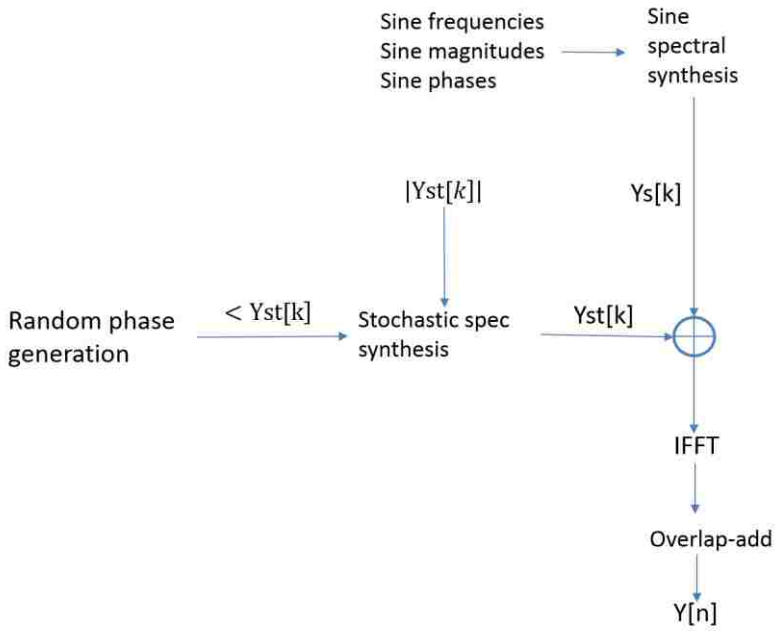


Figure 3.8 Block diagram of the spectral synthesis

The sinusoidal spectrum synthesis process is similar to that used for the sinusoidal modeling; the difference is that the phase trajectories can now be discarded. This approach sacrifices some of the flexibility of the traditional oscillator bank implementation, especially the instantaneous control of frequency and magnitude; however, the increase in speed is significant.

The stochastic spectrum is generated by approximating the residual spectrum with line segments and randomly generating the phase. The generation of the stochastic signal can be understood as the generation of a noise signal with the frequency and amplitude characteristics described by the spectral envelopes of the stochastic representation.

Instead of generating the stochastic signal by approximating the residual spectrum, the residual spectrum can be directly summed with the sinusoidal spectrum. This sine plus residual model is another option for sound synthesis. However, this model does not offer much flexibility in performing sound transformations.

Figures 3.9-3.12 compare the power spectra and spectrograms of the original 3000 RPM and 4000 RPM engine sounds and the synthesized sounds obtained using the sine plus stochastic model. The results show reasonably good similarity between the original and synthesized sounds.

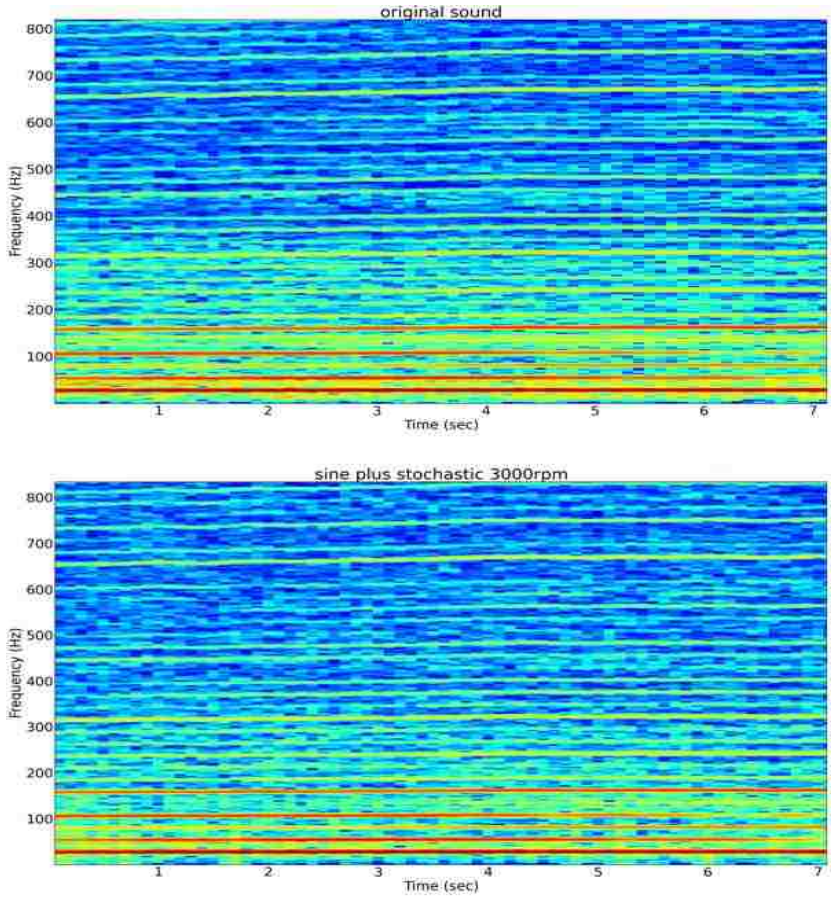


Figure 3.9 Comparison of the spectrograms of the original and SPS-synthesized 3000 RPM sounds

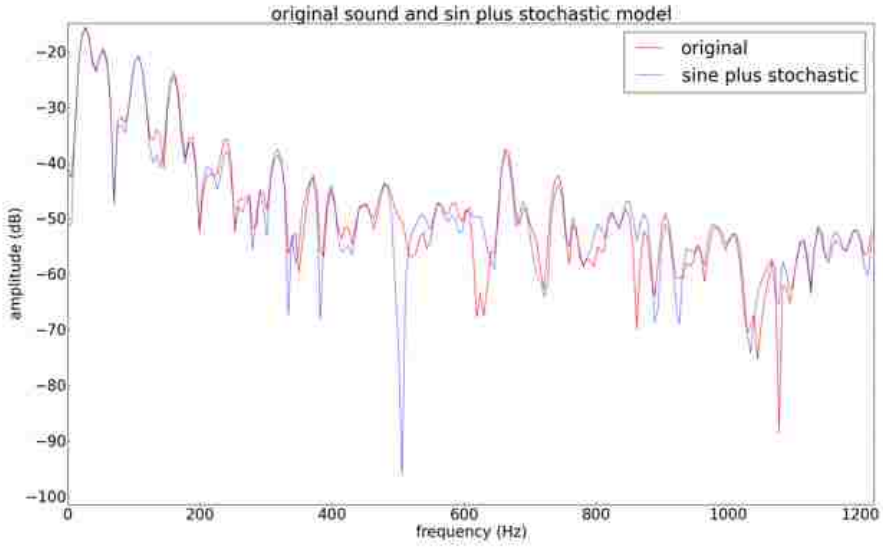


Figure 3.10 Comparison of the spectra of the original and SPS-synthesized 3000 RPM sounds

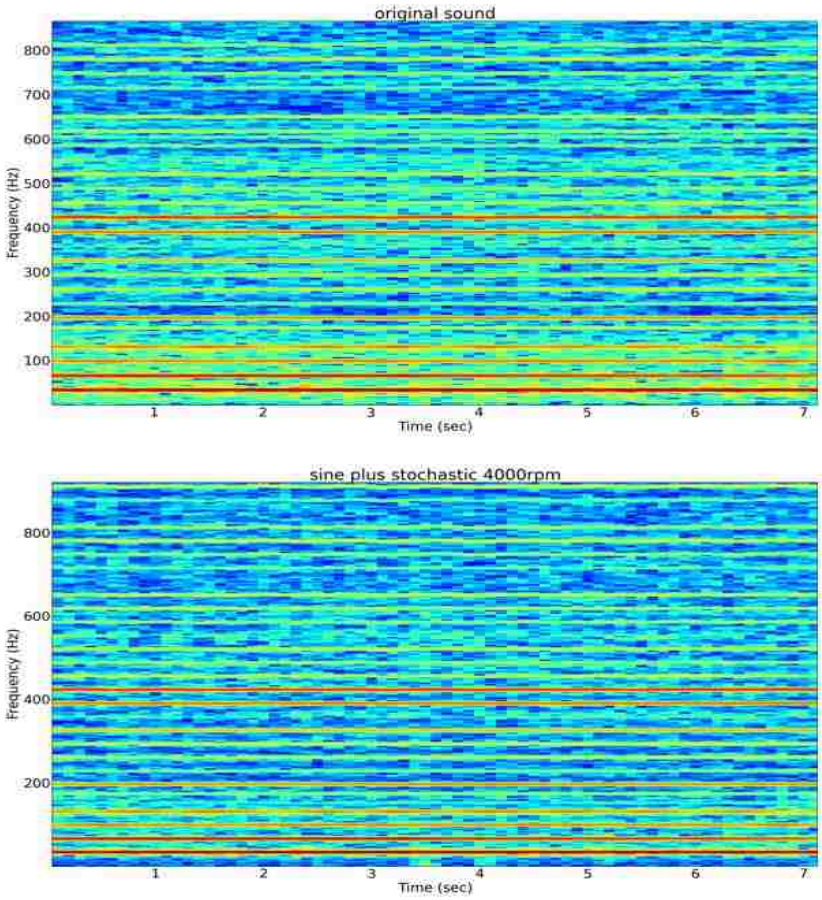


Figure 3.11 Comparison of the spectrograms of the original and SPS-synthesized 4000 RPM sounds

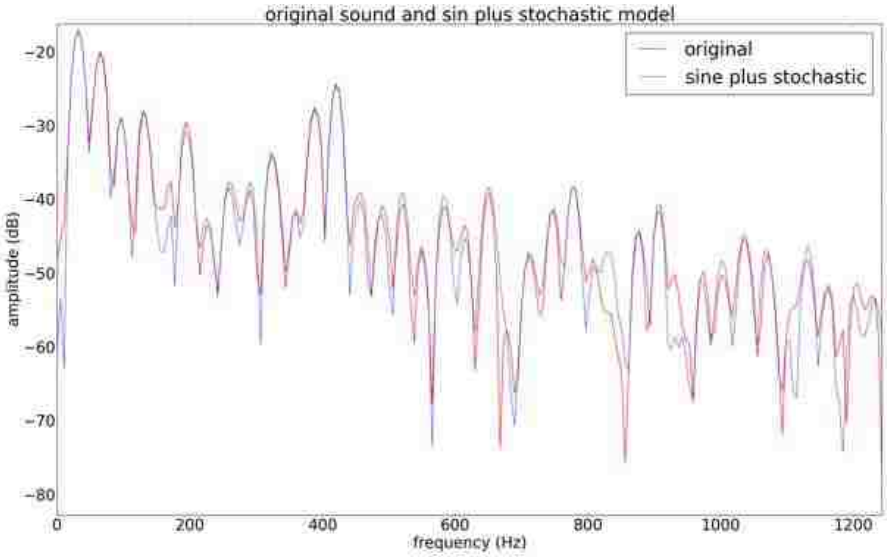


Figure 3.12 Comparison of the spectra of the original and SPS-synthesized 4000 RPM sounds

Figures 3.13-3.16 compare the power spectra and spectrograms of the original 3000 RPM and 4000 RPM sounds with those of the sounds synthesized using the sine plus residual model. The results show good similarity between the original and synthesized sounds.

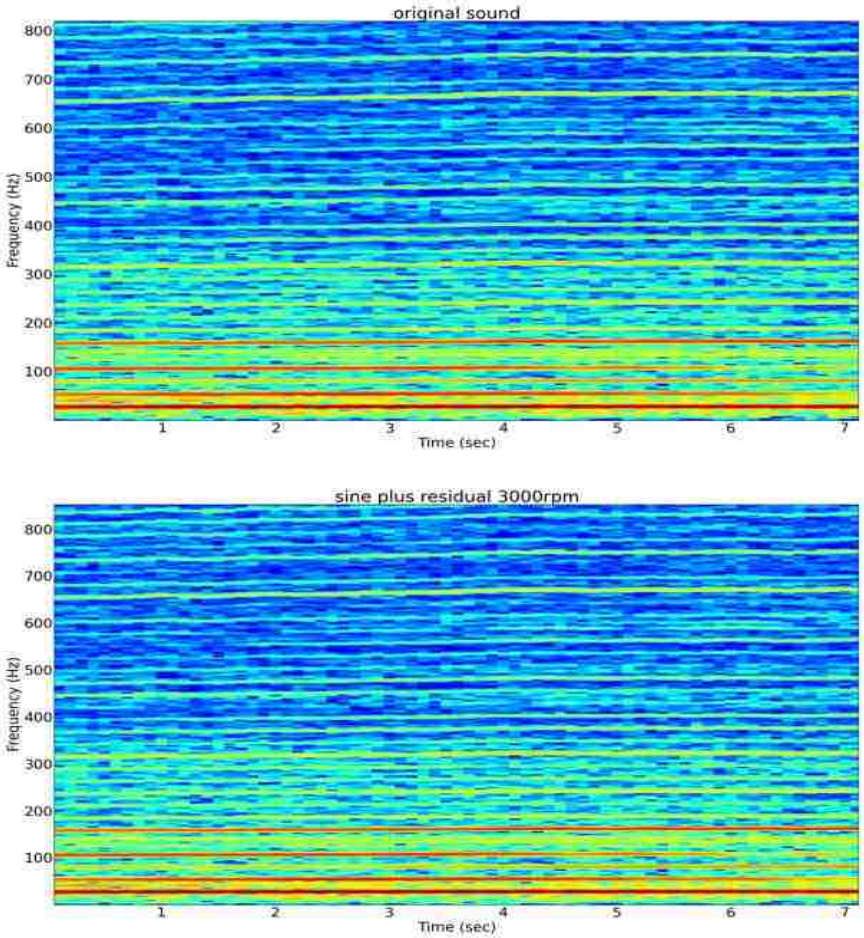


Figure 3.13 Comparison of the spectrograms of the original and SPR-synthesized 3000 RPM sounds

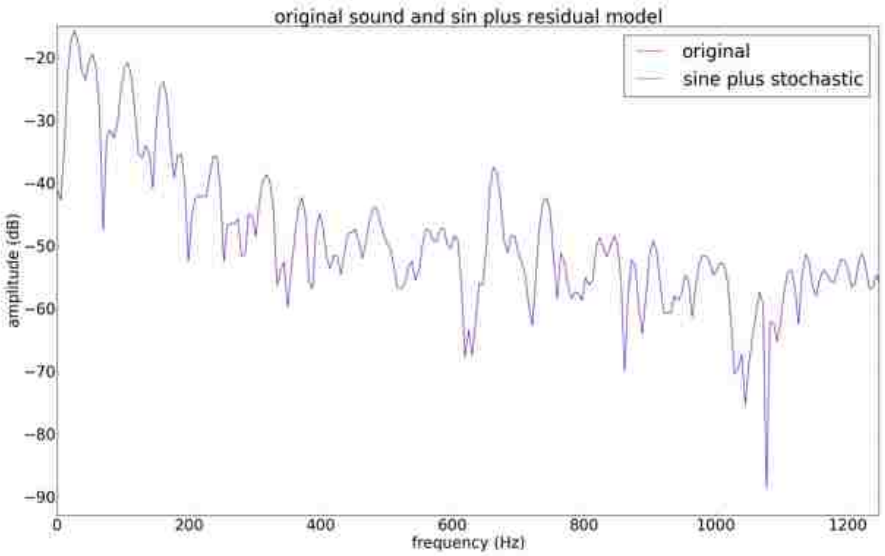


Figure 3.14 Comparison of the spectra of the original and SPR-synthesized 3000 RPM sounds

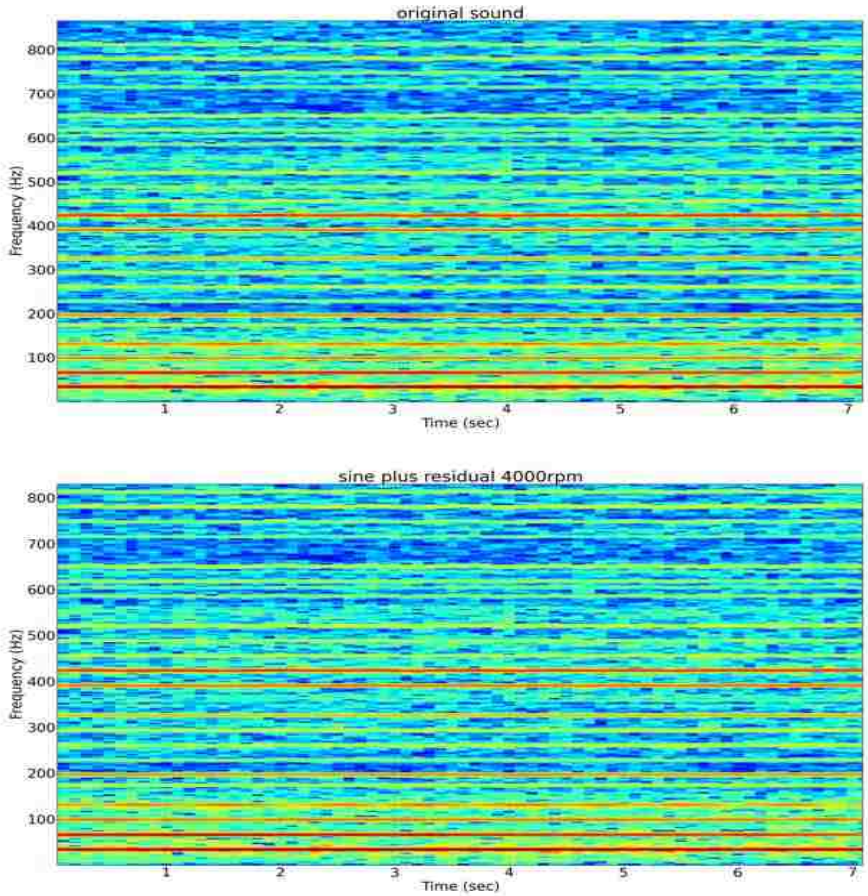


Figure 3.15 Comparison of the spectrograms of the original and SPR-synthesized 4000 RPM sounds

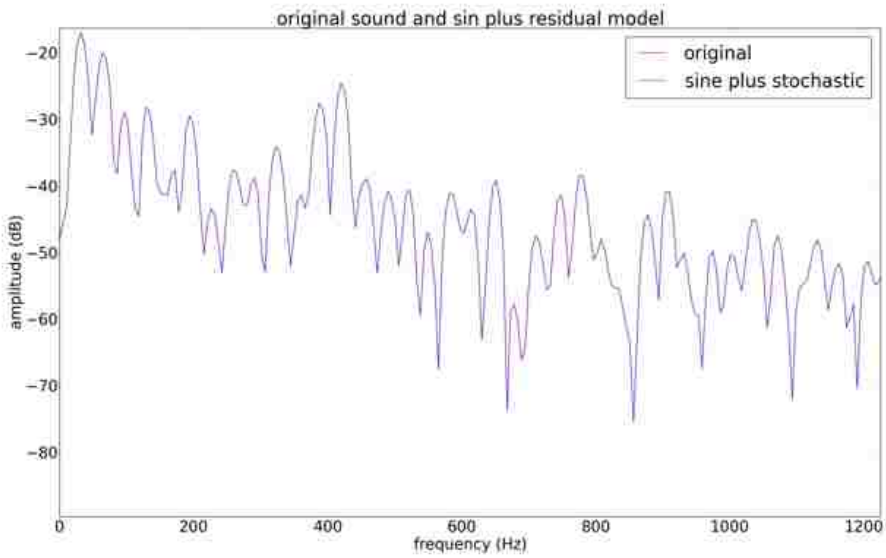


Figure 3.16 Comparison of the spectra of the original and SPR-synthesized 4000 RPM sounds

By comparing the synthesis results of the sine plus stochastic model and the sine plus residual model, it is seen that the latter model can better reproduce the original sound. However, the synthesis results obtained using the sine plus stochastic model are acceptable for driving simulations. Moreover, real-time implementation and sound transformation are easier to perform using this model.

4. SOUND TRANSFORMATION AND MODIFICATION

Because an engine is characterized by a wide range of operating speeds, it is impossible to record sample sounds for all possible engine speeds. To represent engine sounds that have not been recorded, applying transformation and modification to the resynthesized sounds represents a good solution.

As described in the previous chapter, the deterministic analysis yields a set of amplitudes A_t and frequencies f_t, and the stochastic analysis yields a set of spectral envelopes $|Yst[k]|$ that approximate the residual. Together, these representations are ideal for modification purposes and allow for a large number of sound transformations. Modifications are applied separately to the deterministic and stochastic representations.

4.1 SOUND INTERPOLATION

Unlike using the crossfading method to simulate engine sounds that have not been recorded, in our sound interpolation, the frequency components and their amplitudes are smoothly interpolated from one engine sound to another. In crossfading, the frequency components of the engine sounds are added together in such a way that the amplitude of one engine sound gradually vanishes while that of another sound gradually increases [13].

Figure 4.1 shows a block diagram of the sound interpolation process. Sound analysis is conducted using two equal length input sounds, X1 and X2, to obtain the frequencies f_{i1} f_{i2}, magnitudes A_{i1} A_{i2} and residual approximation$|Yst_1|$, $|Yst_2|$; these two sets of values are then interpolated. Based on the new frequency f_{i3}, magnitude A_{i3}, regenerated phase p_3 and stochastic residual envelope $|Yst_3|$, the sinusoidal component Ys[k] and the stochastic component Yst[k] are synthesized. Finally, these two components are combined, and the output sound is generated by applying the IFFT and the overlap-add process.

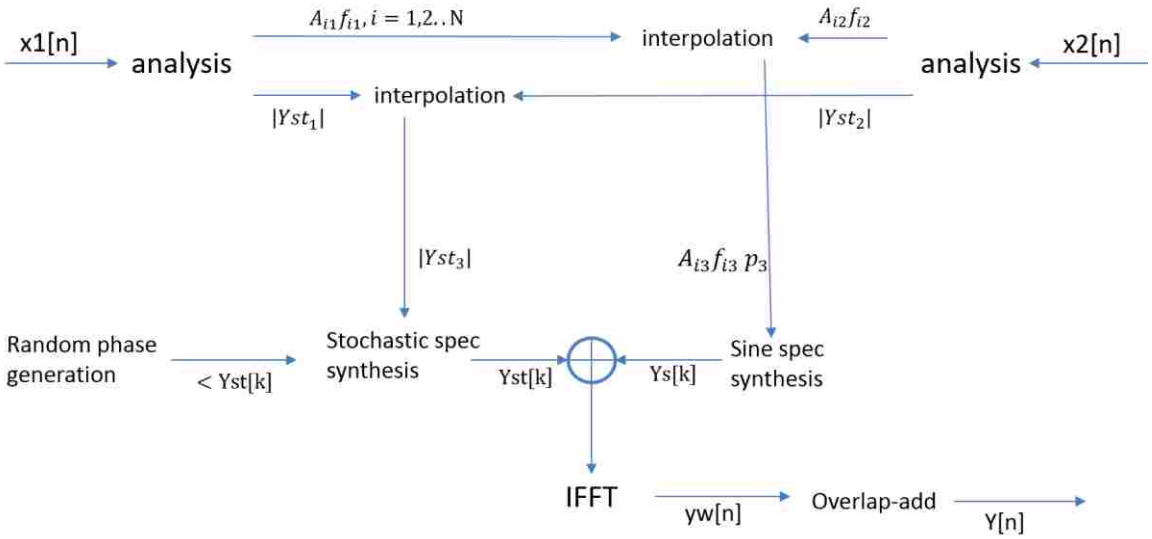


Figure 4.1 Block diagram of sound interpolation

Interpolating the representations of two sounds is markedly different from other sound transformations, such as frequency shifts and magnitude changes; this process generates a new sound with a pitch and shape that are between those of the two original

sounds. Additionally, in the stochastic component, the shape of the interpolated envelope is between those of the two input stochastic envelopes.

4.2 INTERPOLATION RESULTS

Multiple interpolated engine sounds are based on the same input sounds; for example, the 3400 RPM sound and the 3700 RPM sound are both interpolated from the 3000 RPM sound and the 4000 RPM sound. To generate different engine sounds from the same input sounds, a simple linear interpolation factor can be defined as follows:

$$\mathrm{a} = \frac{x - x_1}{x_2 - x_1} \qquad (8)$$

where x is the engine speed corresponding to the interpolated sound and x_1 and x_2 are the engine speeds corresponding to the neighboring sample sounds. Thus, the estimated parameters (frequency, magnitude, etc.) of the interpolated sound can be calculated as follows:

$$A_{i3} = A_{i1} + a(A_{i2} - A_{i1}) \qquad i = 1,2,3 \ldots \ldots N \qquad (9)$$

$$f_{i3} = f_{i1} + a(f_{i2} - f_{i1}) \qquad i = 1,2,3 \ldots \ldots N \qquad (10)$$

$$|Yst_3| = |Yst_1| + a(|Yst_2| - |Yst_1|) \qquad (11)$$

where $A_{i1}\ A_{i2}, f_{i1}\ f_{i2}$ are the magnitudes and frequencies, respectively, of the harmonics from two input sounds, and $|Yst_1|\ , |Yst_2|$ are the spectral envelopes of these input sounds. Using these inputs, the interpolated magnitudes (A_{i3}), interpolated frequencies (f_{i3}), and interpolated spectral envelopes $|Yst_3|$ are calculated. The value of N corresponds to the number of selected harmonics.

For example, to represent the engine sound at 3500 RPM, the sound analysis described in the previous chapter would first be conducted to detect the spectral peaks of the 3000 RPM and 4000 RPM sample sounds. Then, the harmonics of the spectral peaks are identified. In this case, the harmonics for 3000 RPM and 4000 RPM engine sounds are multiples of 26 Hz and 32 Hz, respectively. Finally, the parameters $(A_{i1}\ A_{i2}, f_{i1}\ f_{i2}, |Yst_1|\ , |Yst_2|)$ of these harmonics are estimated, enabling the calculation of the engine sound parameters $((A_{i3}),\ (f_{i3}),\ |Yst_3|)$ at 3500 RPM by employing the interpolation methodology illustrated in Equations (9)-(11) using a factor of 0.5 for the value of a. A graphical representation of this example is shown in Figure 4.2 where the spectrograms of the 3000 rpm, 3500 rpm, and 4000 rpm are compared.

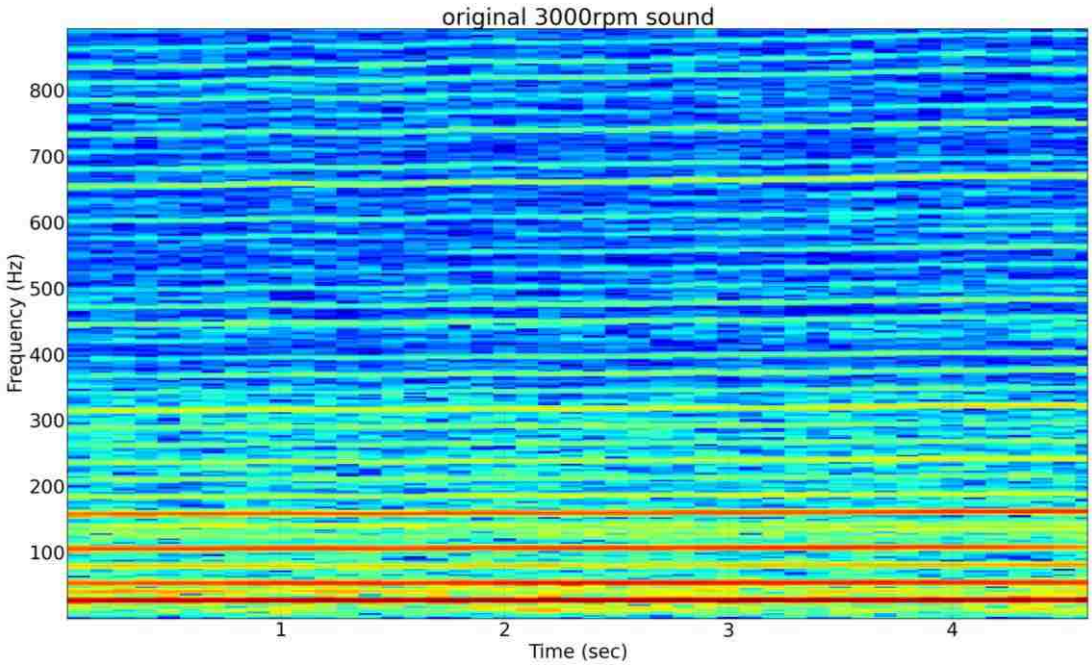


Figure 4.2 (a) Spectrogram of the 3000 RPM engine sound

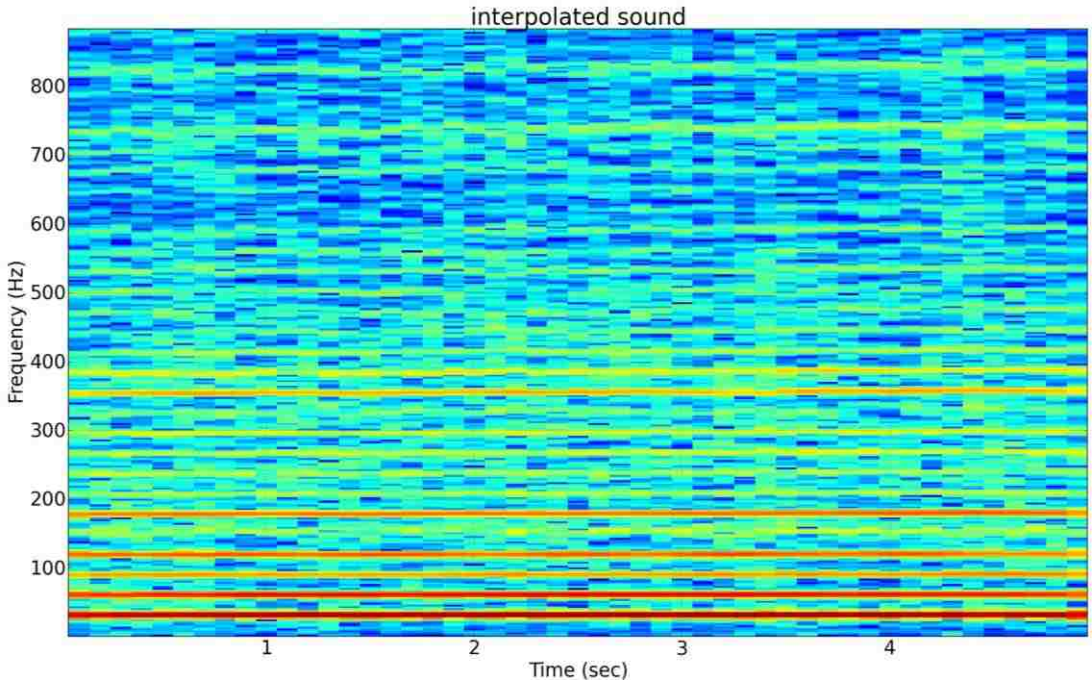


Figure 4.2 (b) Spectrogram of the interpolated sound corresponding to 3500 RPM (a=0.5)

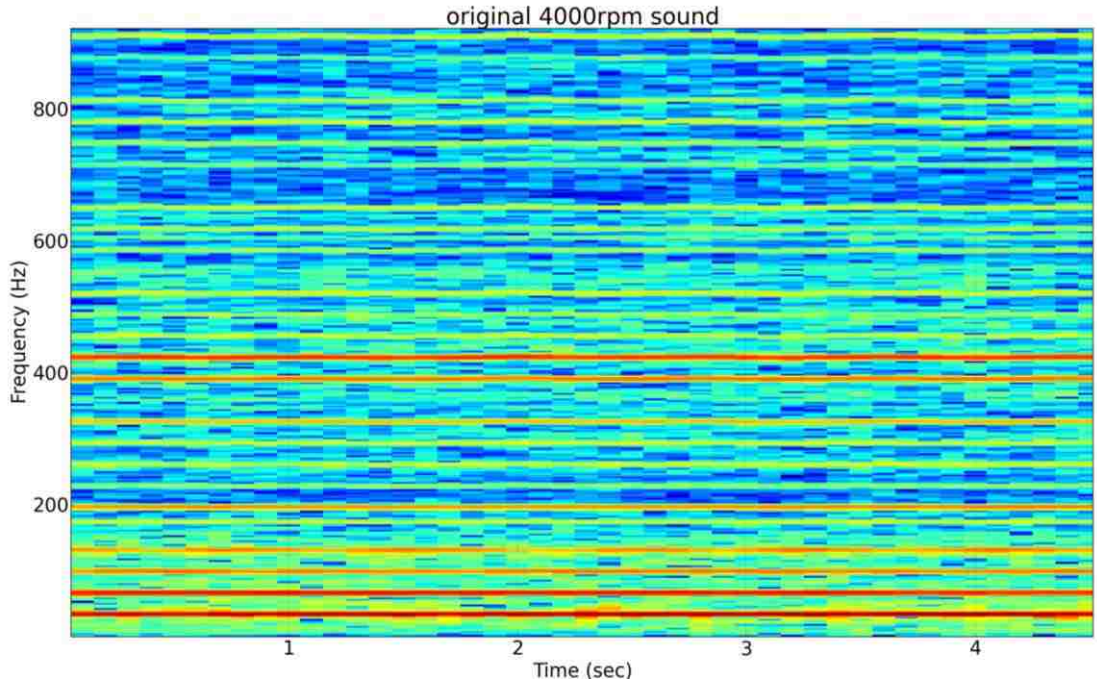


Figure 4.2 (c) Spectrogram of the 4000 RPM engine sound

The comparison of the sound interpolated with a factor of 0.5 with its neighboring sample sounds reveals that the frequencies of the main peaks in the spectrogram of the interpolated sound are slightly higher than those in the 3000 RPM spectrogram, and the main peak frequencies in the 4000 RPM spectrogram are slightly higher than those in the spectrogram of the interpolated sound. Overall, the shape of the spectrogram of the interpolated sound is between those of the 3000 RPM spectrogram and the 4000 RPM spectrogram.

Figure 4.3-4.4 compare the spectrograms and spectra of an interpolated sound with original 3500 RPM sound

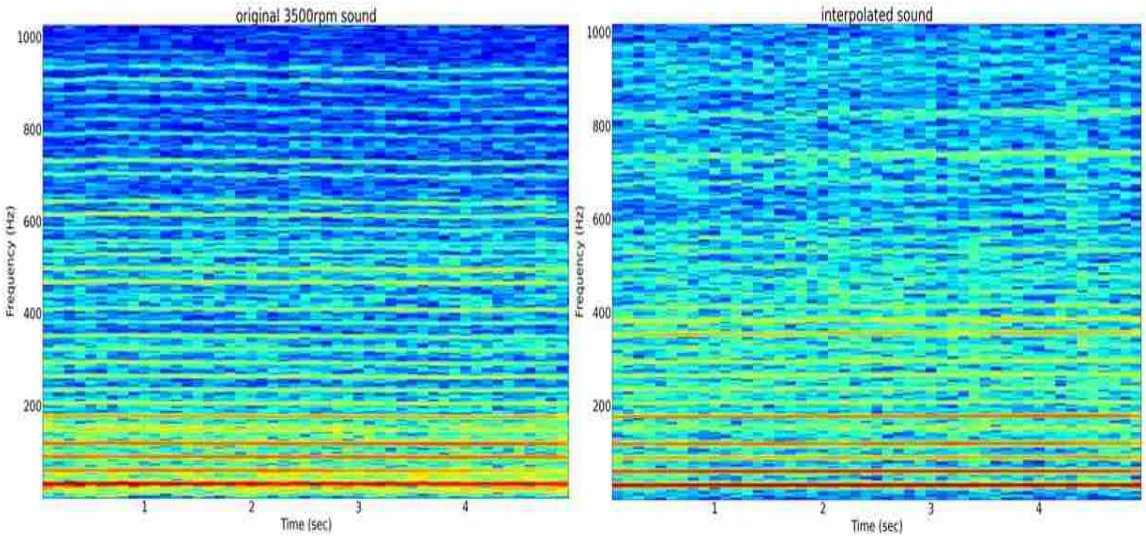


Figure 4.3 Comparison of the spectrograms of the 3500 RPM engine sound and the interpolated sound corresponding to 3500 RPM

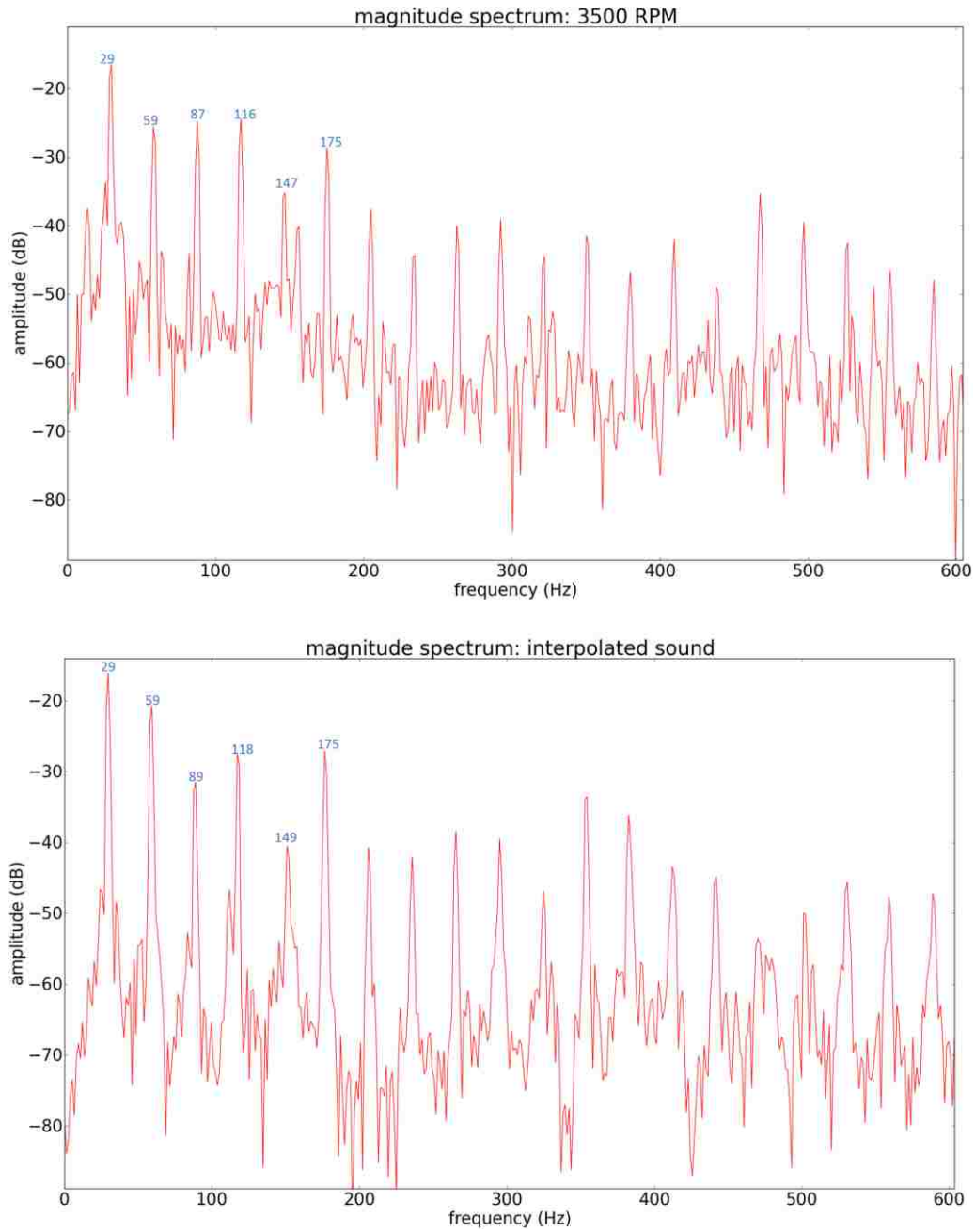


Figure 4.4 Comparison of the spectra of the 3500 RPM engine sound and the interpolated sound corresponding to 3500 RPM

Figure 4.5-4.6 compare the spectrogram and spectrum of an interpolated sound with original 2500 RPM sound

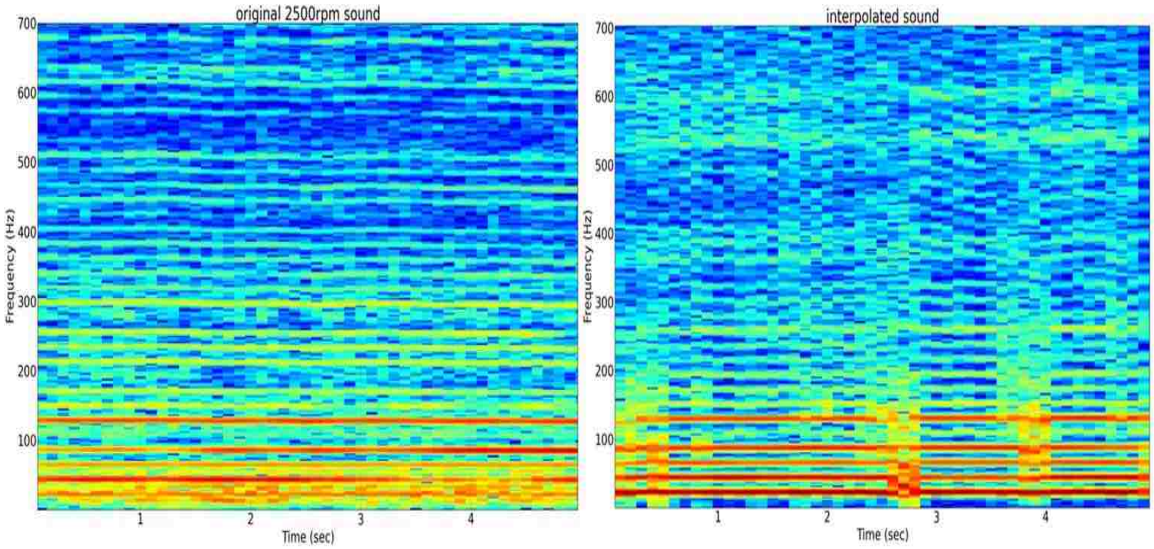


Figure 4.5 Comparison of the spectrograms of the 2500 RPM engine sound and the interpolated sound corresponding to 2500 RPM

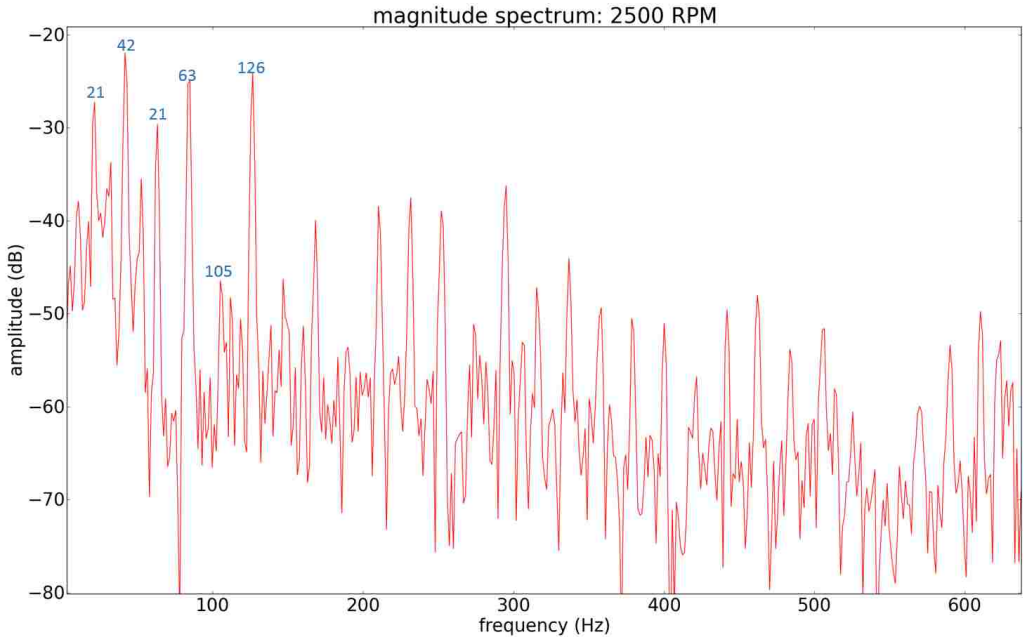


Figure 4.6 Comparison of the spectra of the 2500 RPM engine sound and the interpolated sound corresponding to 2500 RPM

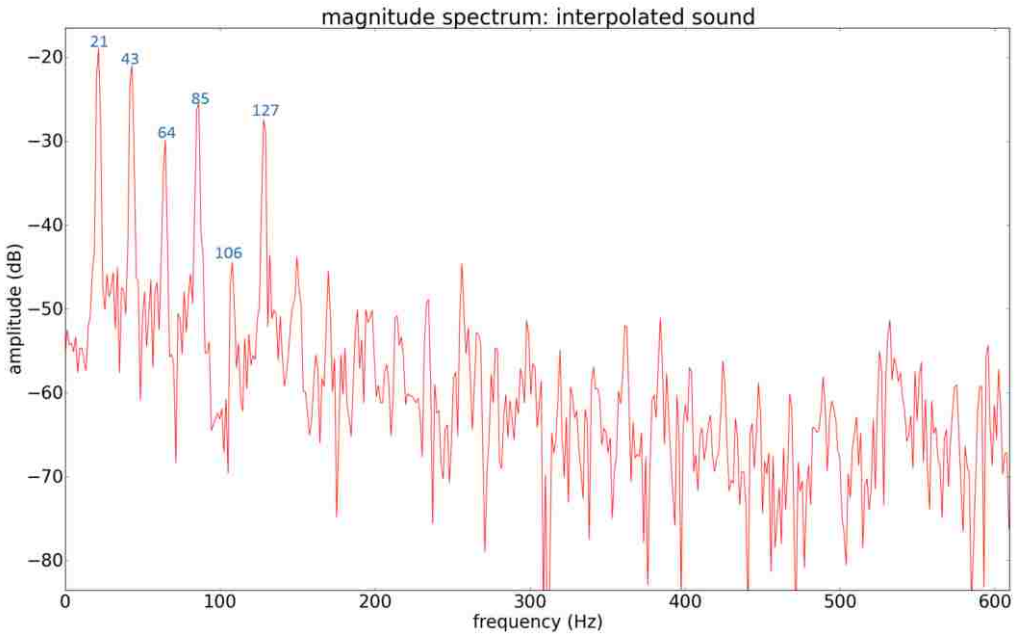


Figure 4.6 Comparison of the spectra of the 2500 RPM engine sound and the interpolated sound corresponding to 2500 RPM (cont.)

In addition to interpolating sounds with a constant interpolation factor, the input sounds can be interpolated using a time-varying factor. For example, to represent the engine sound produced when the engine speed is gradually changing from 3000 RPM to 4000 RPM, the 3000 RPM and 4000 RPM sounds must be interpolated using a time-varying factor that starts at 0 and gradually increases to 1. Figure 4.7 shows the spectrogram of a sound interpolated with a time-varying interpolation factor.

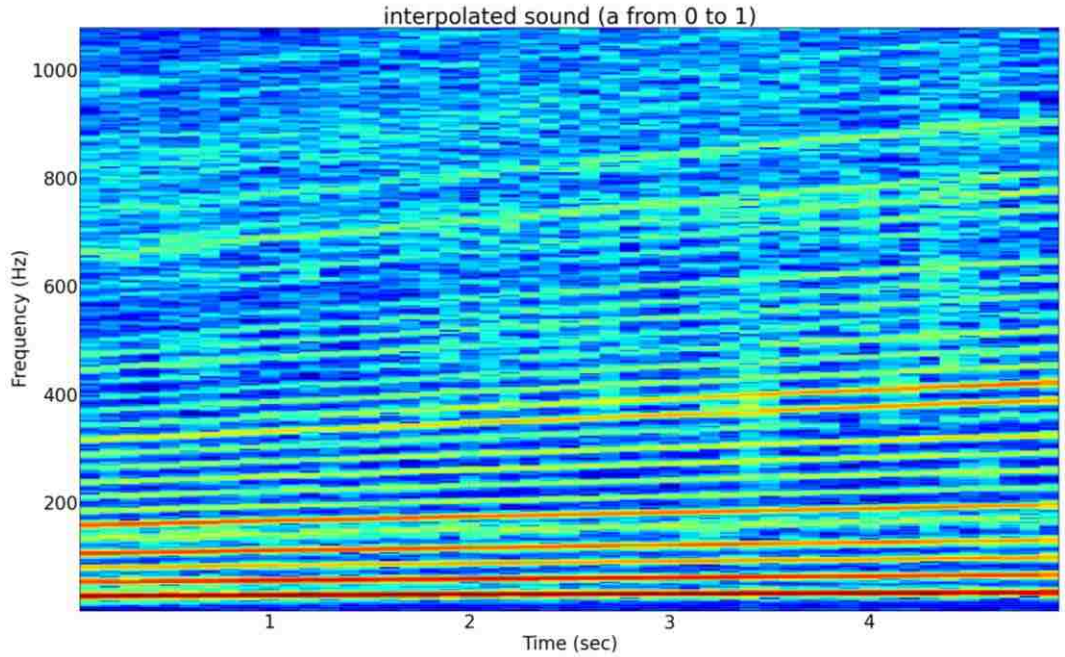


Figure 4.7 Spectrogram of a sound interpolated with a time-varying factor (starting from 0 and ending at 1)

4.3 CLICK CANCELATION

To save memory during the driving simulation, the synthesized sound can be played in a loop while the vehicle remains at a relatively stable engine speed. However, clicks will always be produced at the onset and offset of the sounds. To avoid these clicks and ensure a smooth transition between two sound fragments, the following practical method is introduced:

(1) Use the functions provided in NumPy to convert each sound fragment into an array of numbers($[y_1, y_2, y_3 \dots y_n]$).

(2) Split the array into two halves, and label the two resulting arrays as x1= ($[y_1, y_2, y_3 \dots y_a]$) and x2=($[y_{a+1}, y_{a+2}, y_{a+3} \dots y_n]$).

(3) Use the nonzero() function in NumPy to find the point y_b in x1 with the value closest to y_n.

(4) Extract the new array x3=($[y_b, y_{b+1}, y_{b+2} \ldots y_a]$) from x1.

(5) Combine x3 with x2, resulting in a new array equal to($[y_b, \ldots y_a, y_{a+1} \ldots y_n]$).

(6) Save the new array to the buffer and play it back in a loop.

Figures 4.8 compares the spectrograms of the original sound fragments and the new sound fragments generated using the above method.

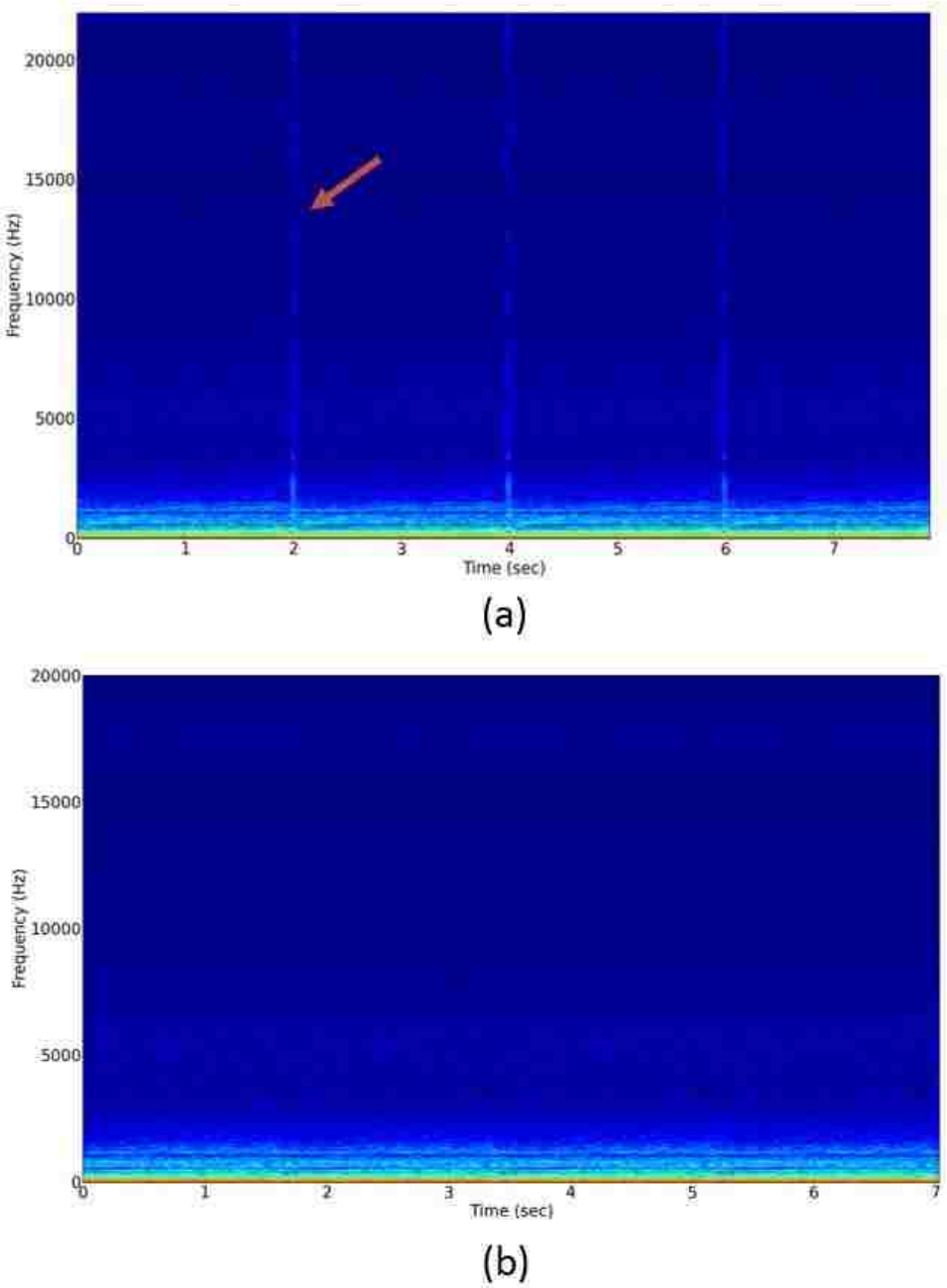


Figure 4.8 (a) Spectrogram of the original sound fragments, with clicks as indicated by the red arrow; (b) Spectrogram of the new sound fragments after click cancellation

By combining sound interpolation with click cancelation, a smooth and time-varying engine sound can be generated. Figure 4.9 shows the spectrogram of an example engine sound that starts at 3000 RPM, increases to 4000 RPM and finally returns to 3000 RPM.

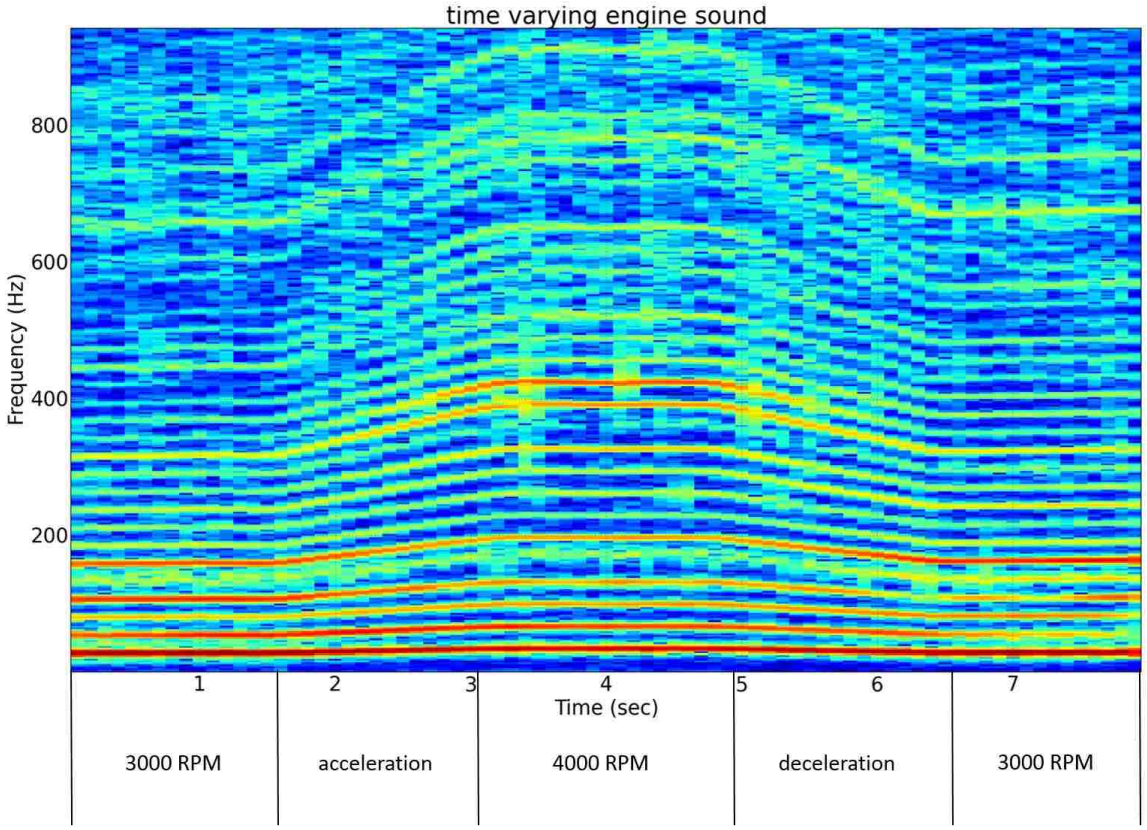


Figure 4.9 Spectrogram of a simulated time-varying engine sound

5. POWERTRAIN MODEL FOR DRIVING SIMULATIONS

Because the engine speed is the input for engine sound generation, a reasonable powertrain model is needed to simulate the engine speed in real time. Without a good model to relate engine speed and vehicle speed, it would be impossible to create a good engine sound.

5.1 FORCE ON THE GROUND VEHICLE

Tractive effort and resistance are the two primary opposing forces that determine the performance of road vehicles. The forces acting on a road vehicle are shown in Figure 5.1. By applying Newton's 2nd law for the equilibrium of forces and momentum, the vehicle acceleration can be calculated as follows [14]:

$$\text{ma} = F_t - R_a - R_{rl} - R_g \qquad (12)$$

where R_g is the grade resistance, F_t is the sum of the tractive effort delivered by the front and rear tires $(F_f + F_r)$, R_{lr} is the sum of the rolling resistances $(R_{rlf} + R_{rlr})$, and R_a is the aerodynamic resistance.

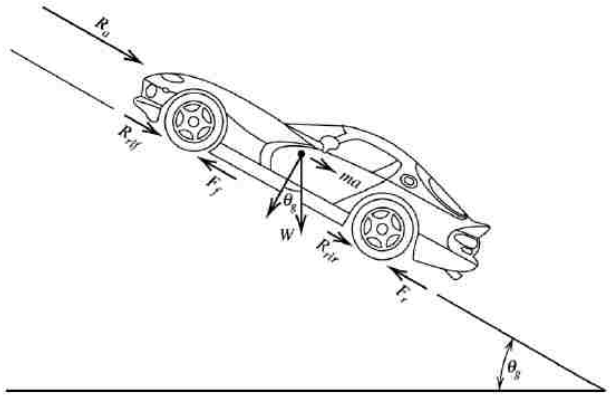


Figure 5.1 Forces acting on a road vehicle

The aerodynamic drag force generated on the front and rear of the body acts on the vehicle as a driving resistance. This resistance can be estimated as a function of the air density, the drag coefficient, the frontal area of the vehicle, and the square of the vehicle's velocity:

$$R_a = \frac{\rho}{2} C_d A_f V^2 \qquad (13)$$

where ρ is the air density, C_d is the drag coefficient, A_f is the frontal area of the vehicle, and V is the speed of the vehicle.

As the vehicle goes up or down a hill, it experiences gravitational resistance due to its weight. The grade resistance can be calculated as follows:

$$R_g = W sin\theta_g \qquad (14)$$

where W is the weight of the vehicle and θ_g is the gradient angle of the road.

The rolling resistance is the friction between the road and the tire surfaces. This resistance can be calculated in a simplified manner as a constant that depends on the vehicle's mass and the rolling resistance coefficient of the tires:

$$R_{rl} = W f_{rl} \qquad (15)$$

where f_{rl} is the rolling resistance coefficient.

5.2 DRIVELINE OF THE VEHICLE

Once the external force on the vehicle has been calculated, the tractive force generated by the driveline must be determined. A basic driveline may consist of an engine, to provide the power needed to move the vehicle, and a transmission system, to transfer that power from the engine to the wheels. In this driveline model, an internal combustion engine characterized by its torque versus rotational speed will be considered. Table 5.1 presents the look-up table for the engine torques at different throttle positions and engine speeds. [15] (USDM 2006 Impreza 2.5i 5MT, ECU value)

Table 5.1 Look-up table for the engine torque

Engine speed \ Throttle position	0.00	1.00	2.00	5.00	10.00	15.00	20.00	25.00	30.00	35.00	40.00	45.00	50.00	60.00	70.00	80.00	100.00
600	0.00	2.30	5.00	23.40	45.20	73.00	99.90	117.70	127.40	129.60	134.00	138.00	142.00	150.00	160.00	168.00	180.00
800	0.00	2.30	5.10	13.60	45.30	73.10	95.60	117.70	127.40	129.60	134.00	138.00	142.00	150.00	160.80	168.10	185.00
1000	0.00	2.70	5.80	11.00	38.80	69.20	92.90	116.30	130.10	135.10	138.50	143.00	148.20	155.40	163.90	171.70	190.00
1200	0.00	2.70	5.80	10.00	32.30	59.00	87.70	113.80	131.80	142.00	147.80	151.60	153.90	164.50	169.30	175.90	200.00
1400	0.00	2.70	5.80	10.00	25.10	51.50	82.70	108.90	129.10	143.10	154.90	159.60	162.70	171.60	174.80	177.60	205.00
1600	0.00	2.70	5.40	8.80	20.00	41.90	74.60	104.30	125.80	142.50	157.30	167.00	169.40	175.80	178.60	181.40	210.00
1800	0.00	2.50	5.10	7.80	17.00	34.20	66.40	94.00	119.10	139.70	157.90	169.20	175.00	180.20	182.10	185.50	215.00
2000	0.00	2.40	4.60	7.00	14.40	25.60	55.30	80.90	110.50	137.60	158.90	171.10	178.00	182.90	185.20	189.00	222.00
2400	0.00	2.20	3.10	5.90	12.00	20.00	37.60	67.00	99.70	133.40	160.20	174.80	182.30	187.20	193.30	198.10	237.00
2800	0.00	2.00	2.80	5.00	9.50	17.00	30.00	56.00	89.80	127.10	154.90	173.80	182.80	192.10	200.00	203.20	247.00
3200	0.00	1.80	2.50	4.40	7.80	15.00	25.00	48.30	78.40	117.40	147.10	169.00	181.50	193.70	201.20	205.40	252.00
3600	0.00	1.70	2.00	3.90	7.00	13.30	21.20	42.00	68.50	108.40	142.20	164.30	182.00	194.10	201.40	208.00	250.00
4000	0.00	1.60	1.90	3.50	6.40	12.00	19.00	36.70	60.10	101.10	138.30	162.10	182.00	196.00	205.70	210.50	240.00
4400	0.00	1.50	1.80	3.20	5.70	10.80	17.00	32.50	53.30	94.90	136.60	161.90	183.40	204.50	214.30	217.80	233.00
4800	0.00	1.50	1.70	2.90	5.30	10.00	15.30	29.10	43.10	83.00	122.90	149.80	174.60	197.10	209.60	215.10	223.00
5200	0.00	1.40	1.60	2.70	4.90	9.00	14.00	25.30	36.50	73.00	109.50	137.60	159.80	187.30	200.60	207.70	218.20
5600	0.00	1.20	1.50	2.50	4.50	8.30	12.50	22.70	32.20	64.40	96.60	125.40	150.00	174.60	190.70	199.00	212.60
6000	0.00	1.20	1.40	2.30	4.00	7.50	11.20	18.00	25.00	55.30	83.00	111.00	132.40	160.80	174.10	183.00	202.90
6400	0.00	1.10	1.30	2.20	3.50	6.50	9.80	16.00	20.00	52.70	77.90	104.30	125.10	153.60	163.50	175.60	192.00
6800	0.00	1.00	1.20	2.10	3.20	6.00	9.00	14.00	18.00	49.60	73.30	98.10	117.70	144.60	153.90	165.30	180.70
7200	0.00	1.00	1.10	2.00	3.00	5.00	7.50	13.00	16.00	46.80	69.30	92.70	111.20	136.60	145.30	156.10	170.70

In this table, the columns represent the throttle position, and the rows represent the engine speed. For a given engine speed and throttle position, the engine torque can be found using this table, and interpolation can be applied to calculate intermediate values as follows: [16]

$$T_e = F_e(\varphi_f, n_e) \qquad (16)$$

where T_e is the engine torque, φ_f is the throttle position, n_e is the engine speed, and F_e is the function representing the torque look-up table. Once the operational parameters of the engine have been defined, the next step is to define the transmission layout and the transmission system.

Figure 5.2 presents a schematic representation of the driveline for a front-wheel-drive vehicle [17]. The engine provides torque and rotational speed. This torque transmitted to the gearbox through the torque converter. In the gearbox, depending on the gear selected based on the vehicle's speed, the rotational speed and torque will vary proportionally to the gear ratio $(\xi_i, i = 1, 2, 3, 4, 5)$. Finally, the output of the gearbox is transmitted to the differential through the end gear, which also has its own transmission ratio (ε_e). The last consideration regarding the driveline is the inclusion of a transmission efficiency (η_d) to account for the torque losses generated in the driveline.

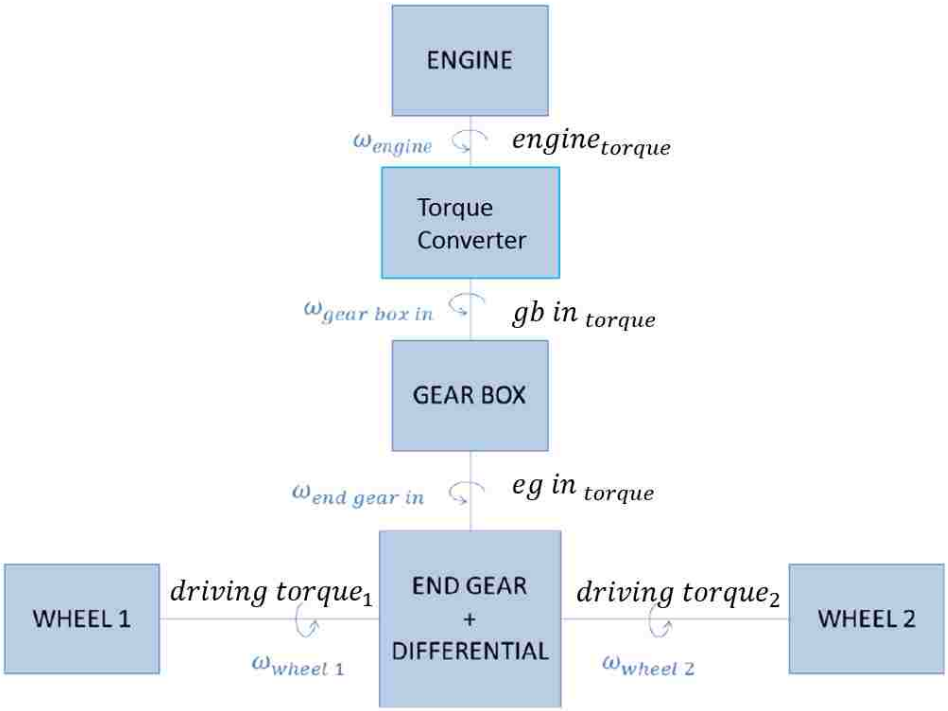


Figure 5.2 Schematic representation of the driveline

The gear ratio (ξ_i) in the gearbox usually follows a close-to-geometric progression, in which the ratio changes by a constant percentage from one gear to the next. Figure 5.3 illustrate the relationship between the engine speed and the vehicle speed [18].

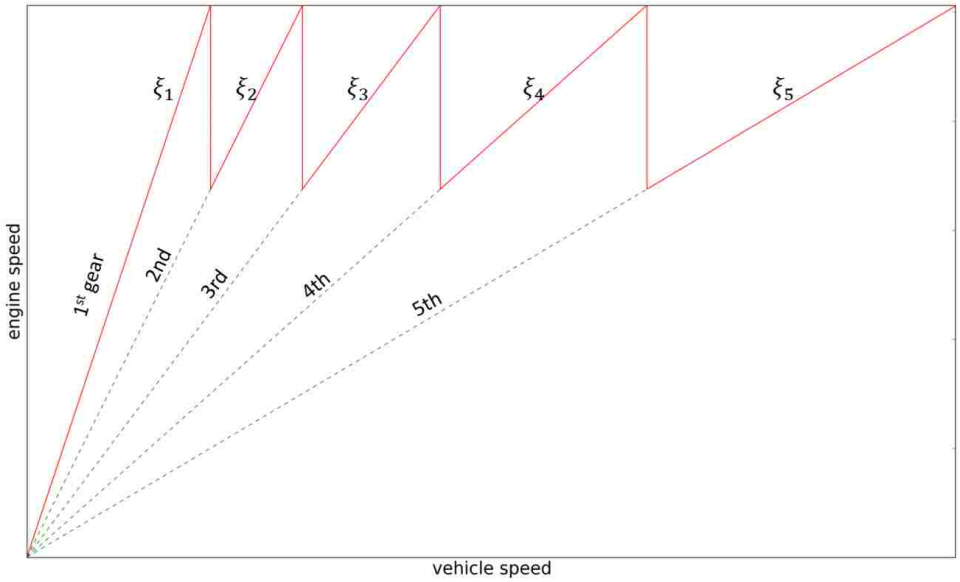


Figure 5.3 Relationship of engine speed to vehicle speed

With these terms defined, the engine-generated tractive force that reaches the wheels is given as follows:

$$F_t = \frac{T_e \xi_i \varepsilon_e \eta_d}{r} \qquad (17)$$

where ξ_i is the gearbox ratio, T_e is the engine torque, ε_e is the transmission ratio of the end gear, η_d is the transmission efficiency, and r is the tire radius.

5.3 SIMULATION BLOCK DIAGRAM AND RESULTS

When the car is in motion, it is assumed that the engine speed is equal to the wheel speed scaled by the current gear ratio and the final drive axle transmission system [19]. The engine speed can be calculated as follows:

$$n_e = \frac{V \xi_i \varepsilon_e}{2\pi r(1-i)} \qquad (18)$$

where V is the vehicle speed, n_e is the engine speed, and i is the drive axle slippage,

which is generally taken to be between 2 and 5 percent (i=0.02 to 0.05). Figure 5.4 shows the flowchart of the powertrain algorithm.

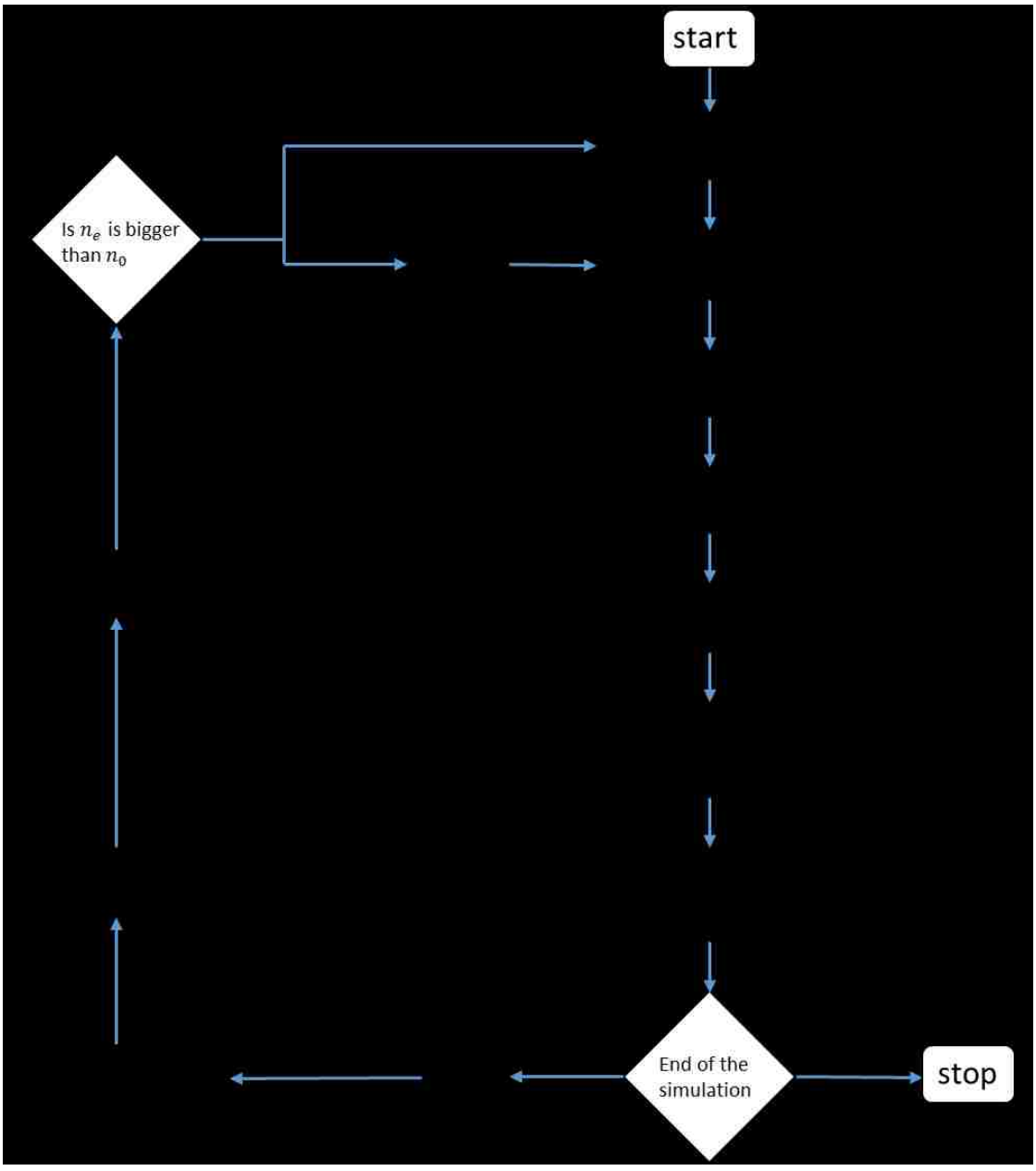


Figure 5.4 Flowchart of the powertrain algorithm

Based on the flowchart in figure 5.4, the whole simulation can proceed as follows:

1. At T=0, set an initial engine speed n_0 for the engine and obtain the throttle position φ_f from the gas petal.
2. Based on engine speed n_0 and throttle position φ_f, use equation (16) to calculate the engine torque T_e.
3. Obtain the current vehicle velocity V from the virtual environment. Based on the vehicle velocity V, select the gear from gearbox and find out current gear ratio ξ_i.
4. With gear ratio ξ_i and engine torque T_e, the current tractive force F_t can be calculated using equation (17).
5. According to equation (13), (14) and (15), the external resistance can be calculated. Then apply the tractive force and resistance on the vehicle in virtual environment.
6. Obtain the vehicle velocity V' at $T' = \mathrm{T} + 1$, select the gear from gearbox and find gear ratio ξ_i', and then use equation 18 to calculate the engine speed n_e.
7. If n_e is larger than n_0, take n_e as the engine speed at T'. If n_e is smaller than n_0, take n_0 as the engine speed at T'.
8. Based on the new engine speed at T', obtain the new throttle position and calculate the whole process again.

In this powertrain model, when the car is stationary or traveling at a very low speed in the 1st gear, it must be assumed that the torque converter is in operation and that a finite fluid slip is occurring. Under these conditions, the actual delivered torque is determined by the converter slip, the throttle position and the wheel velocity. Because the simulation of a vehicle at low speeds is very complicated and generally not required, it is assumed that the torque converter is locked at all times and that the engine speed is saturated at a constant minimum value.

Since the vehicle's parameters for a specific type of vehicle are not all available, this study selected general parameters of the vehicle for the simulation in the driving simulator. In terms of the external resistance, the rolling resistance coefficient is from 0.010 to 0.015 for ordinary car tire on concrete; the drag coefficient is from 0.25 to 0.55 for automobile and air density is 1.2kg/m^3; the projected frontal area typically ranges from 1 m^2 to 2.5 m^2 for passenger cars. Table 5.2 shows the parameters selected for the simulation.

Table 5.2 Vehicle's parameters for the simulation

Vehicle's parameter	Values
Length	4.2 m
Width	1.6 m
Height	1.3 m
Weight	1200 kg
Tire radius	0.317 m
Front area	2 m^2
Drag coefficient	0.3
Rolling resistance coefficient	0.013
Gearbox ratio (1^{st}-5^{th})	2.8 1.8 1.3 1.0 0.8
Final drive ratio	3.6
Transmission efficiency	0.92

An acceleration experiment was conducted at Missouri University of Science and Technology's driving simulator, and the virtual environment scenario was constructed by Blender3D and Blender Game Engine. Figure 5.5 shows the driving simulator and virtual environment (see Appendix A for details). For each of the experiments performed, the following conditions were implemented:

1. The terrain was completely flat with no changes in elevation.
2. A quiescent atmosphere with negligible wind speeds.

Figure 5.5 Driving simulator at Missouri University of Science and Technology

Figure 5.6 shows the simulated engine speed over time. In this figure, the times at which the engine was switched into different gears are evident. Figure 5.7 shows the simulated vehicle velocity over time, from which it can be seen that the maximum

vehicle speed is 210 km/h and the time required to accelerate from 0 to 100 km/h is 6.4s, this is comparable to the time required under real driving conditions.

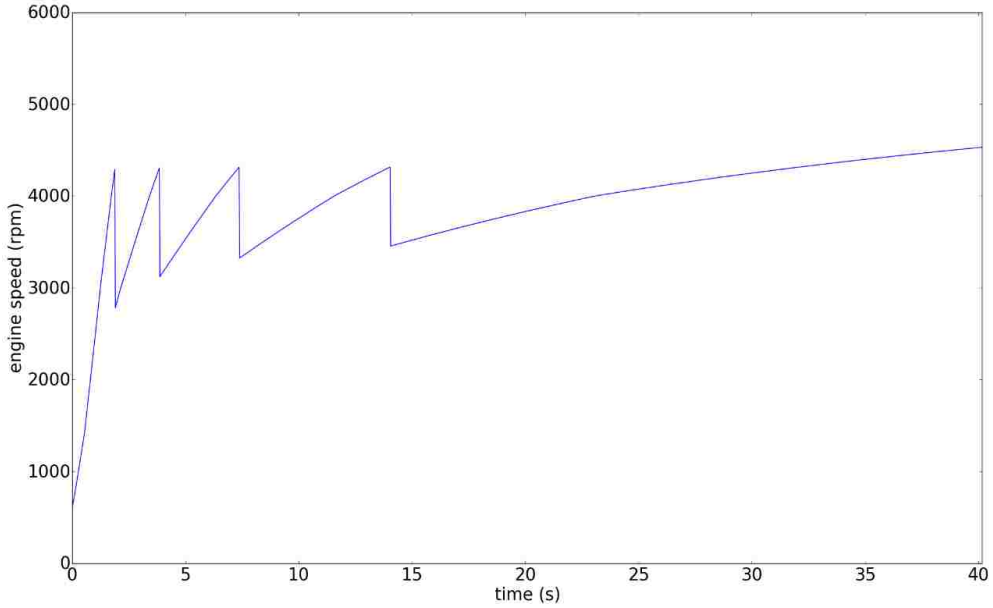


Figure 5.6 Simulated engine speed over time

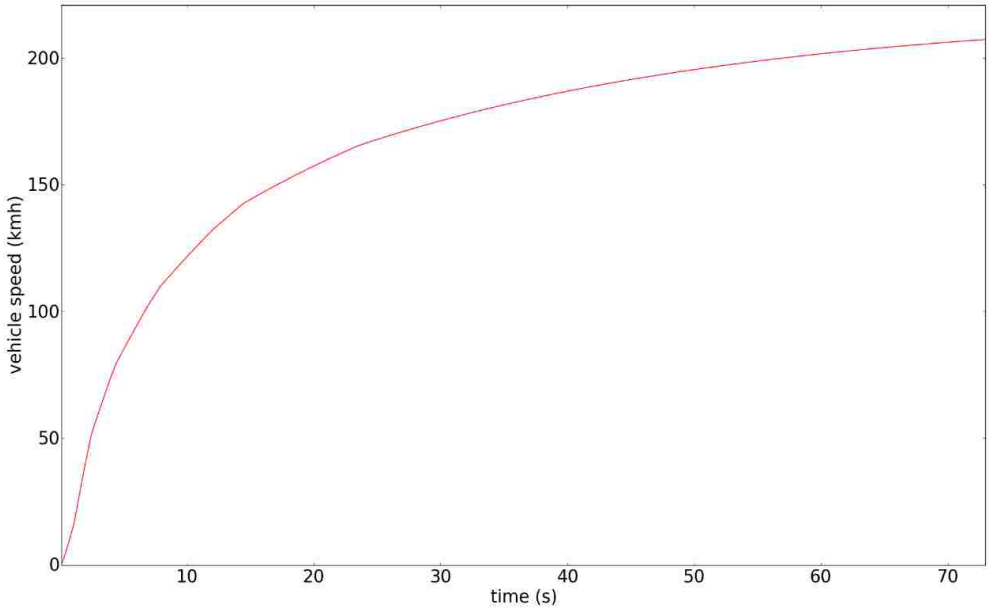


Figure 5.7 Simulated vehicle speed over time

Figure 5.8 shows the spectrogram of a simulated engine sound as the vehicle is accelerating.

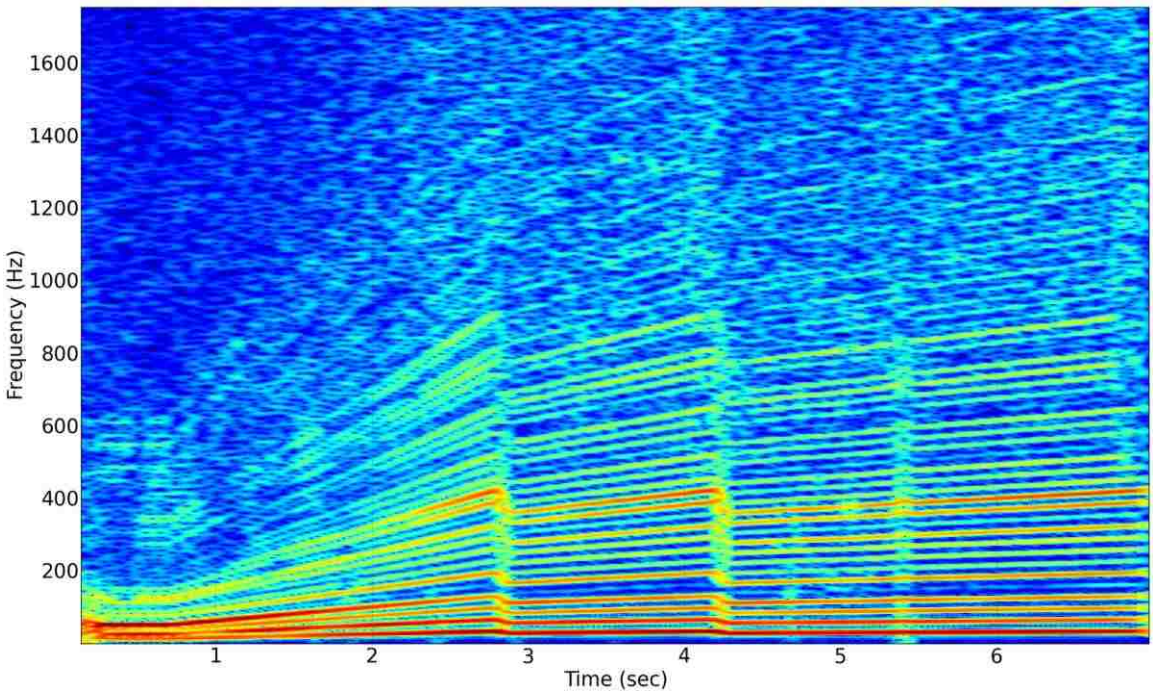


Figure 5.8 Spectrogram of a simulated engine sound

6. CONCLUSION

This thesis describes a method of generating engine sounds for a driving simulator. By analyzing sample sounds obtained using a digital recorder and in-ear microphones, spectral modeling synthesis (SMS) was used to decompose the sound samples into deterministic and stochastic components. Then, the modeled sinusoidal and stochastic signals were employed to resynthesize the sample sounds. To represent engine sounds that were not available in the database, sound interpolation was applied to two neighboring engine sounds. To remove the repeating clicks produced when playing sounds in a loop, we employed a cancellation method that involves splitting the synthesized sound and reconstructing a new engine sound snippet. Finally, a powertrain model was introduced to calculate the engine speed for the sound generation. As seen by comparing the spectrograms of the sounds, the resynthesized sounds exhibited high similarity to the recorded sample sounds. The spectrogram of the engine sound generated by interpolating two sample sounds was found to agree fairly well with the actual engine sound. Moreover, by interpolating two sample sounds with a time-varying factor, the synthesized engine sound could gradually change from one sound to another. The described method provides a practical solution for generating engine sounds of relatively good quality.

APPENDIX

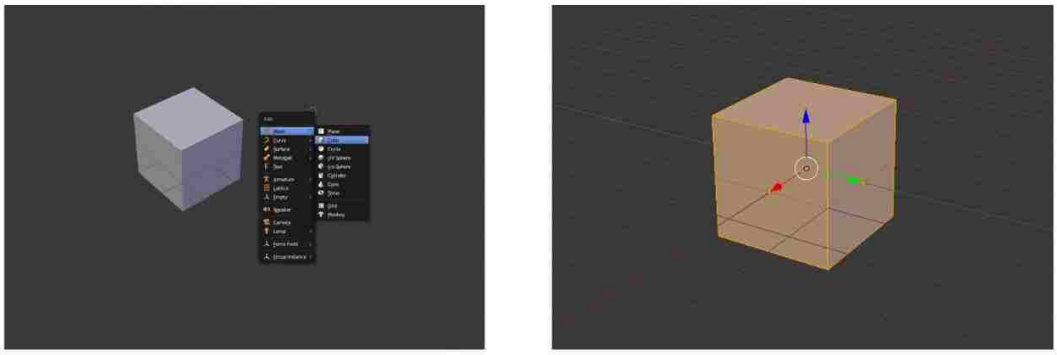
Driving Simulator Overview and Operation

1. Modeling software

The Blender 3D is a professional free and open-source 3D computer graphics software. Basically, we are using its 3D modeling and UV unwarping features to build the road, road signs, cones, etc. in the virtual environment. Alongside the modeling features it also has an integrated game engine. This game engine fully supports vehicle dynamics, including suspension stiffness, tire friction, etc. It also allows us to program the real-time interactive contents that relate to driving.

2. Objects in virtual environment

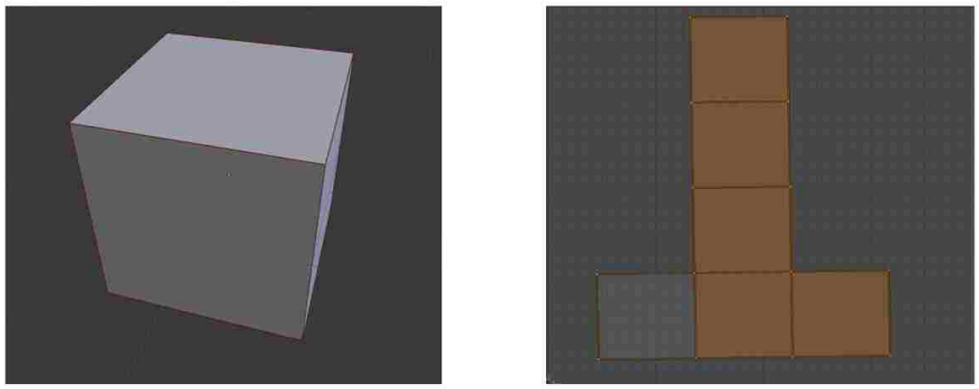
In order to build the 3D model of an object in the real world, it's better to start with some basic mesh such as plane, cube, cylinder, etc. After that we could modify the geometry of the mesh by editing its vertex, edge and face. For the object that has multiple components and complex geometry such as vehicles and trees, we download the 3D model from the internet and import them into our scenario.



(a) (b)

Figure A.1 (a) adding basic mesh. (b) the mesh in edit model

In order to make the 3D object looks real, we need to use the “UV mapping” feature to project a 2D image to a 3D mesh. The letter “U” and “V” are the names of the axes of the 2D texture. When the 3D model of an object is created, we could select the 3D model’s edge as the UV beams in edit model. Base on the UV beams, the software will unwrap the model and generate the UV map. As long as we have the UV map, we can apply the 2D texture to the 3D model.



(a) (b)

Figure A.2 (a) is the 3D cube in x-y-z coordinate with UV beams (red lines). (b) is the unwrapped mesh in p(u,v) coordinate.

3. Python script in BGE

The BGE (Blender Game Engine) using a system of graphical “logic bricks” (a combination of “sensor”, “controller”, and “actuator”) to control the movement and display objects in engine, the game engine also can be extend by binding the Python script. The movement and setup of the vehicle in our scenario is controlled by several “sensor” bricks that connect with the python scripts. The “sensor” brick will receive the input data from the gas pedal, brake and steering wheel, then these data will be processed by Python scripts and generate the force, resistance, torque, etc. on the vehicle. All the Python script programed in BGE uses Python 3.x, and there are a lot of in-build functions for vehicle setup and power transmission in BGE

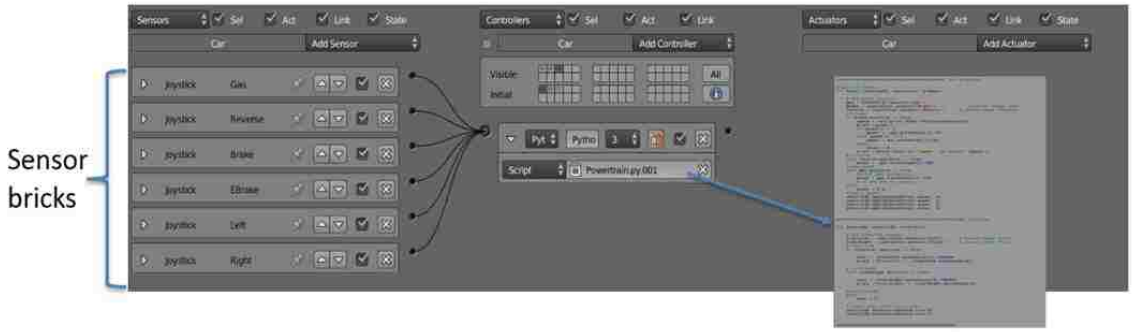


Figure A.3 use of logic brick and Python script to control the vehicle’s movement

4. Simulator operation

Step 1: Powering the Simulator.

- Facing the simulator, the projectors left to right are Projector1 (id=1), Projector2 (id=2) and Projector3 (id=3).

- The name of the master computer is fordsimdev.
- Connect projectors 1, 2 and 3 to video cards of the master computer.
- Make sure all the projectors are turned on and the master computer is turned on too.

Step 2: Configuring the Projectors

- Open the catalyst control center icon to configure the projectors.
- Adjust the display settings so that the three projectors have the same resolution.
- Open Catalyst Control Center: (a) Click on AMD Eyefinity Multi-Display. (b) Select a layout for display; for the simulation purpose select the 4x1 (4 horizontal displays setting). (c) Select the 3 projectors and the desktop to create a layout display.
- Select the resolution to be 1280*1024 on all the four screens.

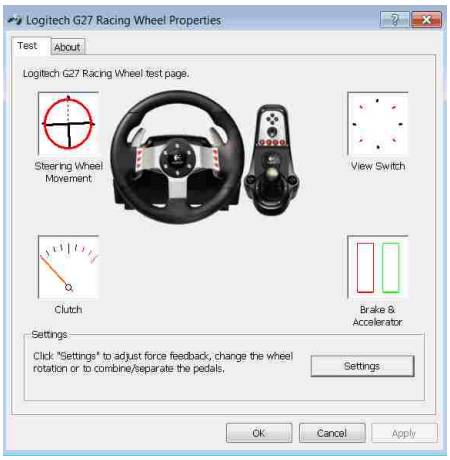
Step 3: Configuring the Arduino

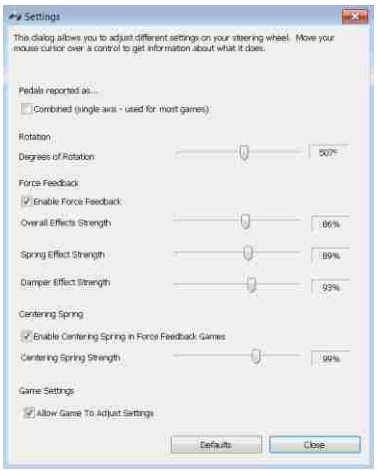
- Connect the Arduino port to one of the USB ports of the master computer.
- To switch on the Arduino, upload the program from the Arduino software.
- Next open the Python IDLE module, and open the program Adwrite. This will upload the program to configure the Arduino as a speedometer and receive and display the speed. This has to be done before the start of the simulation.

Step 4: Configuring the Steering Wheel.

- Programs → Logitech-G27 gaming profiler→ Click on Select a Game → and choose Blender as your default game engine.

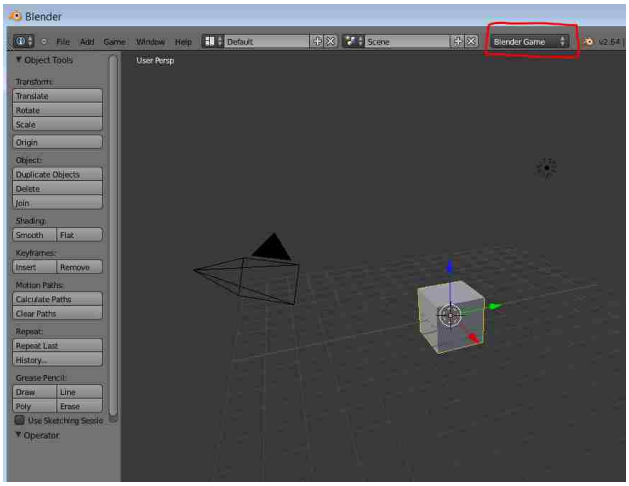
- Start→Devices and Printers→G 27 Racing wheel →Right Click→ Game controller settiings→Settings→ Apply the settings below.



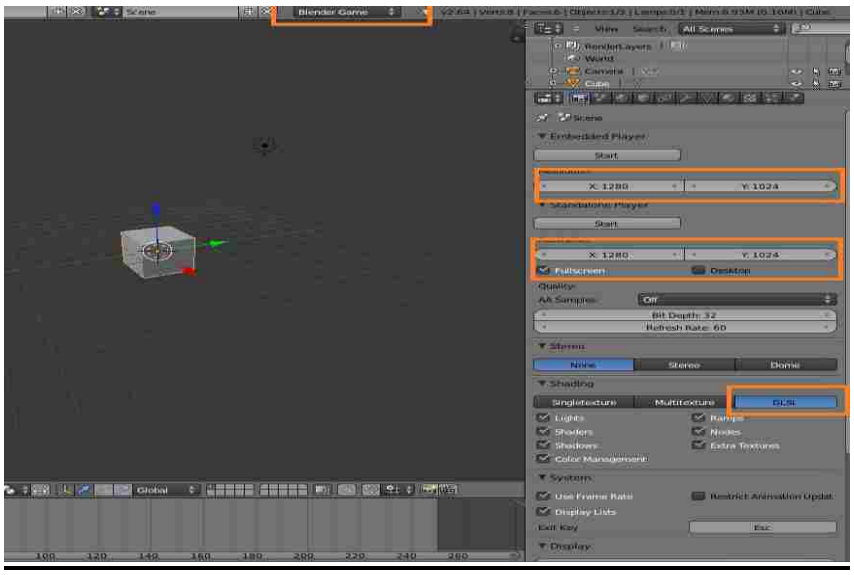


Step 5: Configuring Blender

- Start→ Open Blender→ Change the default setting into Blender game.



- Change the resolution of the embedded player into 1280x1024, and change the shading mode to GLSL.



After these initial settings are completed, select the scenario and with the mouse cursor in the 3D window press “P” for the game to begin.

BIBLIOGRAPHY

[1] Genell, A., "Perception of Sound and Vibration in Heavy Trucks," Göteborg, Sweden: Chalmers University of Technology, 2008.

[2] Sandberg, U., "Combined effect of noise, infrasound and vibration on driver performance," *Proceedings of Inter Noise* (pp. 887 – 890), 1983.

[3] Ihs, A., Andersson, J., Kircher, K., and Bolling, A., "*Trafikanternas krav på vägars tillstånd*," Swedish National Road and Transport Research Institute, VTI rapport 669, 2010.

[4] Denjean, S., and Roussarie, V., "How does interior car noise alter driver's perception of motion? Multisensory integration in speed perception" Acoustics Nantes Conference, April 2012.

[5] Merat, N., and Jamson, H., "A driving simulator study to examine the role of vehicle acoustics on drivers' speed perception," *Proceedings of the Sixth International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design,* 226-232, 2011.

[6] Horswill, M.S., and Plooy, A.M., "Auditory feedback influences perceived driving speeds", *Perception* 37, 2008.

[7] Blommer, M., and Greenberg, J., "Realistic 3D sound simulation in the VIRTTEX driving simulator," Driving Simulator Conference North America, October 2003.

[8] Heitbrink, D.A., Cable, S., "Design of a Driving Simulation Sound Engine," Driving Simulator Conference North America, September 2007.

[9] Dardas, N. H., and Georganas, N. D., "An Efficient Technique for Modeling and Synthesis of Automotive Engine Sounds," IEEE Transactions on industrial electronics, VOL. 48, 2001.

[10] Serra, X., "Musical sound modeling with sinusoids plus noise." Musical signal processing (1997): 91-122.

[11] Strawn, J., "Approximation and syntactic analysis of amplitude and frequency functions for digital sound synthesis." Computer Music Journal 4, no. 3 (1980): 3-24.

[12] Serra, X. and Smith, J., "Spectral modeling synthesis: A sound analysis/synthesis system based on a deterministic plus stochastic decomposition." Computer Music Journal 14, no. 4 (1990): 12-24.

[13] Perez-Meana; H., Gale, T., “Advances in audio and speech signal processing: technologies and applications,” Hershey, PA : Idea Group Pub., 2007, pp.80-90.

[14] Mannering, F., Kilareski, W., and Washburn, S., “Principles of highway engineering and traffic analysis”. John Wiley & Sons, 2007.

[15] http://forums.nasioc.com/forums/showthread.php?t=1537010. May. 2016

[16] Fang, T., Wei, Y. D., Zhou, X. J., and Li, P. X., "Research on vehicle dynamics simulation for driving simulator." In Advanced Materials Research, vol. 308, pp. 1946-1950. 2011.

[17] Gómez Fernández, J. "A Vehicle Dynamics Model for Driving Simulators." (2012).

[18] Gillespie, T. D. Fundamentals of vehicle dynamics. Vol. 114. SAE Technical Paper, 1992.

[19] Short, M., Pont, M.J., and Huang, Q. "Simulation of vehicle longitudinal dynamics." Embedded System Laboratory University of Leicester, Safety and Reliability of Distributed Embedded Systems. Leicester (2004).

VITA

The author of this thesis, Shuang Wu was born in January 4, 1991 in HuBei, China. He received his Bachelor degree of Engineering in Automotive Engineering from WuHan University of Technology, HuBei, China in June 2013. He joined the Master of Science program in Mechanical Engineering at Missouri University of Science and Technology, Rolla, Missouri in August 2013. The author received his Master of Science degree in Mechanical Engineering from Missouri University of Science and Technology in July 2016.