Real-time Automotive Engine Sound Simulation with Deep Neural Network

Hao Li[1], Weiqing Wang[2], and Ming Li[2]

[1] Z-one Technology co. Ltd, Shanghai, China
[2] Data Science Research Center, Duke Kunshan University, China

Abstract. This paper introduces a real-time technique for simulating automotive engine sounds based on revolutions per minute (RPM) and pedal pressure data. We present a hybrid approach combining both sample-based and procedural methods. In the sample-based technique, the sound of an idle engine undergoes pitch-shifting proportional to the ratio of current RPM to idle RPM. For the procedural technique, deep neural networks fine-tune the amplitude of the engine's pulse frequency derived from the sample-based sound. To ensure the synthesized sound does not have any clicks between the frames, we utilize a modified griffin-lim algorithm at the frame level, which, with our proposed overlap-and-add feature, can bridge the phase gap between two frames. Experimental evaluations on our self-collected database validate the efficacy of the introduced approach.

Keywords: Engine Sound Simulation; Engine Sound Synthesis; Real Time Synthesis

1 Introduction

The engine sound is one of the most overlooked aspects in driving simulation as it gives an indication of the state of the vehicle. For the in-car environment, it affects speed judgment, operator performance, alertness, and fatigue [1–4]. In addition, it can provide auditory feedback to the driver. Drivers can make decisions according to the engine sound, e.g., changing gears using the pitch of the engine sound or maintaining a steady vehicle speed. Drivers often underestimate the vehicle speed and have difficulty in maintaining a target speed if no engine sound is provided [4–9]. Leading automakers, including BMW, Audi, Ford, and Jaguar, are actively engaged in research focused on stimulating drivers' emotions and conveying distinct brand identities through vehicle sounds [10]. For the out-car noise, it can inform pedestrians and cyclists of the vehicle's approaching, avoiding many traffic accidents [11, 12].

Nowadays, electric motor-driven vehicles (EVs) are emerging due to their environmental friendliness, and fuel-efficient performance [13]. However, the electric motor usually cannot generate the sounds as an internal combustion engine. Therefore, simulation of the combustion engine sound is important in this kind of vehicle, and the vehicle sound quality is important to improve ride comfort [14–18].

In general, there are sample-based and procedural methods for engine sound synthesis [19]. The sample-based method is the most common approach, where the sound samples are looped and then resampled or pitch-shifted based on the revolutions per minute (RPM) or other signals [20–22]. Heitbrink et al. [1] use the wavetable approach to synthesize the engine sound, where crossfading is applied during playback to shift the frequencies between two sound samples.Lee et al. [23] also employ the wavetable approach that can maintain the shape of string sound waveforms and vary the pitch during acceleration. Scott et al. [24] use the deterministic-stochastic signal decomposition approach to synthesize the automotive engine sound. The deterministic component is first extracted from the original sound using the synchronous discrete Fourier transform (SDFT). Then they use a multi-pulse excited time-series method to model the stochastic component. They find that the audio quality can be improved using weighted error minimization. Van et al. [25] propose a phase vocoder-based method to simulate the engine sound for a driving simulator. They extract the acoustic features and modify these features to change the speed of the signal. Then they estimate the representation of the modified signal and finally resample the generated signal. Jan et al. [26] propose a real-time algorithm for engine sound synthesis. They extract the sound samples from a recorded engine sound within the entire engine speed range. Then they employ an extension of the pitch-synchronous overlap-and-add (PSOLA) method to locate the extraction instants of the sound samples and finally produce the engine sound. Recently, Dongki et al. [27] propose an engine sound synthesis method. They first generate a mechanical sound by summing harmonic components representing sounds from rotating engine cranks. And then they simulate a combustion noise using random sounds with similar spectral characteristics to the measured value. Finally, the mechanical sound and the combustion noise are combined to produce the engine sound.

In the procedural method, the sound is generated from some attributes of the engine sound. Stefano [19] proposes a procedural method based on the mechanics of the actual four-stroke engines. Fu et al. [28] simulate the engine motion sense sound by reading the vehicle running state data on the CAN bus of pure electric vehicles.

In this paper, a hybrid method for engine sound simulation is proposed, where both sample-based and procedural methods are employed. For the sample-based method, the pitch of the signal is shifted from the sounds of the idle engine according to the frame-level RPM, where the griffin-lim algorithm (GLA) [29] is employed. To remove the clicking between each frame, we propose the griffin-lim overlap-and-add (GLOLA) method and generate the sound given different RPM. For the procedural methods, we generate the spectrum of the engine sound with only RPM and pressure on the pedal (POP) using the deep neural network (DNN). These two spectrums generated by these two methods are finally summed up and converted to the engine sound by GLOLA.

The rest of this paper is organized as follows: Section 2 introduce the sample-based method with GLOLA; Section 3 present the DNN-based procedural method; Section 4 is the experiments and results; Section 5 concludes this paper.

2 Sample-based Method with Griffin-lim Algorithm

The most common sample-based approach in the most driving simulator is the wavetable approach [1]. In this technique, a collection of sound samples are mixed or manipulated to generate the engine sound. To be more specific, the sound samples between different speeds are recorded during a real-world drive. Next, the sounds are cross-faded depending on the vehicle's speed. However, the onset and offset of each sound will be audibly identifiable when the sounds are played in a loop, which results in repeated clicking. Therefore, the sound snippets should be faded in / out slowly to hide the repeated clicking or apply some cancellation methods to remove the repeated clicking [30].

Unlike the wavetable approach, we directly generate the engine sound at a different speed from the idle engine sound. Therefore, only a very short sound sample of the idle engine is needed (usually less than 1 second). The sound is first pitch-shifted based on the ratio between the current RPM and the idle RPM, and then playback in a loop. However, the clicks also exist as the wavetable approach does. Actually, these clicks are caused by the discontinuity of the phase between the boundaries of two sound samples. To remove such a click, we employ the griffin-lim algorithm to recover the phase near the boundaries. More specifically, we propose the griffin-lim with overlap-and-add (GLOLA) in frame-level to generate the engine sound without clicks as the traditional GLA is not originally designed for frame-level synthesis.

2.1 Griffin-lim algorithm

Griffin-lim algorithm (GLA) is a phase recovery algorithm that can recover a complex-valued spectrogram [29]. Given a real-valued amplitude $\mathbf{A}$, GLA generates the complex-valued spectrogram $\mathbf{C}$ in the following iterative projection procedure [31]:

$$\mathbf{C}^{[i+1]} = P_{\mathcal{C}}\left(P_{\mathcal{A}}(\mathbf{C}^{[i]})\right), \tag{1}$$

where $P_{\mathcal{S}}$ is the metric projection on a set $\mathcal{S}$, i is the iteration index, and $\mathbf{C}^{[0]} = \mathbf{A}$. $\mathcal{C}$ is the set of consistent complex-valued spectrograms and $\mathcal{A}$ is the corresponding spectrogram set with the same amplitude. The projection is given by:

$$P_{\mathcal{C}}(\mathbf{C}) = \mathcal{G}\mathcal{G}^{-1}\mathbf{C}, \tag{2}$$

$$P_{\mathcal{A}}(\mathbf{C}) = \mathbf{A} \odot \mathbf{C} \oslash |\mathbf{C}|, \tag{3}$$

where $\mathcal{G}$ represent short-time Fourier transform (STFT), $\mathcal{G}^{-1}$ is the pseudo inverse of STFT (iSTFT), $\odot$ denotes the element-wise multiplicatio, and $\oslash$ denotes the element-wise division. The goal of GLA is to reduce the distance

$D(\mathbf{C}, P_{\mathcal{C}}(\mathbf{C}))$ [32]:

$$\min_{\mathbf{C}} ||\mathbf{C} - P_{\mathcal{C}}(\mathbf{C})||_F^2, \tag{4}$$

where $||\cdot||$ is the Frobenius norm.

One limitation of applying GLA for engine sound simulation is that GLA is not designed for real-time synthesis at frame-level since it employs the STFT, which takes several frames of a signal as input. In addition, the phase discontinuity is not resolved yet. Actually, overlap-and-add has been inherently implemented in the STFT/iSTFT calculation, which results in a continuous signal. This motivates us to incorporate the GLA into the STFT/iSTFT process, which is the griffin-lim overlap-and-add (GLOLA) algorithm.

To synthesize the engine sound in real-time, the input of GLA becomes a frame of signal. We also perform the overlap-and-add [33] operation at the end of each frame-level iteration so that the phases are continuous between two successive frames. In the phase estimation step, we only calculate the phase of each harmonic component, which dramatically reduces the computational cost in each frame-level iteration.

2.2 Griffin-lim algorithm in frame-level

STFT/iSTFT is the process of several fast Fourier transform (FFT)/ pseudo inverse of FFT (iFFT) calculation with overlap-and-add in the frame-level. Given an amplitude $\mathbf{A}$ and let $\mathbf{C}^{[0]} = \mathbf{A}$, the traditional GLA can be composed in four steps:

1. Projecting the spectorgram $\mathbf{C}$ in $\mathcal{A}$ as Eq. 1 shows;
2. iFFT calculation on each frame of spectrogram with overlap-and-add, producing the continuous signal;
3. FFT calculation on each frame of the continuous signal, producing the spectrogram $\mathbf{C}$;
4. Assessing the distance $D(\mathbf{C}, P_{\mathcal{C}}(\mathbf{C}))$; if it is small enough, stop iteration and return the signal; otherwise, go back to step 1.

As we split the GLA step by step, it is obvious that GLA performs several FFT/iFFT calculations with overlap-and-add in one iteration since it accepts several frames as input.

To simulate the engine sound at frame-level, the GLA must take only one frame as input and iteratively estimate the phase for each frame. Therefore, we iteratively perform the FFT/iFFT, and the overlap-and-add will be applied after the distance is converged for this frame. Give an amplitude $\mathbf{a}$ of a frame, previous simulated signal $\mathbf{y}$ and let $\mathbf{c} = \mathbf{a}$, the GLOLA contains five steps:

1. Projecting the spectorgram frame $\mathbf{c}$ in $\mathcal{A}$ as Eq. 1 shows;
2. iFFT calculation on the frame, producing a frame of signal $\hat{\mathbf{y}}$

Algorithm 1 The GLOLA algorithm

```
Require: $l > o > 0$ \\ $l$ is the frame length and o is the frame overlap as required by STFT
Require: Pre−recorded sample sound $x \in \mathbb{R}^n$ at a stable idle RPM
Ensure: $\mathbf{y} = \mathbf{0} \in \mathbb{R}^o$, $\hat{\mathbf{y}} = \mathbf{0} \in \mathbb{R}^l$, $L = o$
while simulation process does not stop do
    $\hat{\mathbf{x}} \leftarrow \text{Resample}(x, \frac{\text{current RPM}}{\text{idle RPM}})$
    $\mathbf{a} = |\text{fft}(\hat{\mathbf{x}})|$
    while $D(\mathbf{c}, P_{\mathcal{C}}(\mathbf{c}))$ is not small enough do
        $\mathbf{c} = \text{fft}(\hat{\mathbf{y}})$
        $P_{\mathcal{A}}(\mathbf{c}) \leftarrow \mathbf{a} \odot \exp(\text{Angle}(\mathbf{c}) \times i)$
        $\hat{\mathbf{y}} \leftarrow \text{ifft}(P_{\mathcal{A}}(\mathbf{c}))$
        $\hat{\mathbf{y}}[0 : o] \leftarrow \hat{\mathbf{y}}[0 : o] + \mathbf{y}[L - o : L]$
        $D(\mathbf{c}, P_{\mathcal{C}}(\mathbf{c})) \leftarrow ||\mathbf{C} - P_{\mathcal{C}}(\mathbf{C})||_F^2$
    end while
    $\mathbf{y} \leftarrow \text{Append}([\mathbf{y}, \hat{\mathbf{y}}[0 : (l - o)]])$
    $L \leftarrow L + (l - o)$
end while
```

3. Adding this signal $\hat{\mathbf{y}}$ to the previous simulated signal $\mathbf{y}$ with overlap: $\hat{\mathbf{y}}_{0\sim\mathrm{o}} \leftarrow \hat{\mathbf{y}}_{0\sim\mathrm{o}} + \mathbf{y}_{(\mathrm{L-o})\sim\mathrm{L}}$, where o is the frame overlap of the STFT and L is the total length of $\mathbf{y}$. This means that we only add the overlapping part of these two signals.
4. FFT calculation on $\hat{\mathbf{y}}$, producing the spectrogram $\mathbf{c}$;
5. Assessing the distance $D(\mathbf{c}, P_{\mathcal{C}}(\mathbf{c}))$; if it is small enough, stop iteration and append this signal $\hat{\mathbf{y}}$ to the end of $\mathbf{y}$; otherwise go back to step 1.

Algorithm 1 shows a more detailed process of our engine sound simulation with GLOLA. Note that this algorithm does not exactly start from step 1 as mentioned above, but the iterative process is similar.

3 Procedural Method with Deep Neural Network

Although we obtain a simulated engine sound by GLOLA, the characteristics of the engine are not carefully considered. As mentioned in [24], the engine pulse frequency F_0 is equal to:

$$F_0 = \frac{RPM}{60} \times \frac{p}{2}, \tag{5}$$

where p is the number of cylinders. Figure 1 also shows the relationship between the RPM and the engine pulse frequency F_0. In addition, F_0 and its multiples have a higher amplitude than others. These harmonic components at multiples of F_0 have a significance at low-frequency band, which can also represent the characteristics of the engine [27]. However, even F_0 can be easily calculated from RPM, the relationships between the RPM and the amplitude on these multiples of F_0 are less obvious than the engine pulse frequency does, as shown in Figure

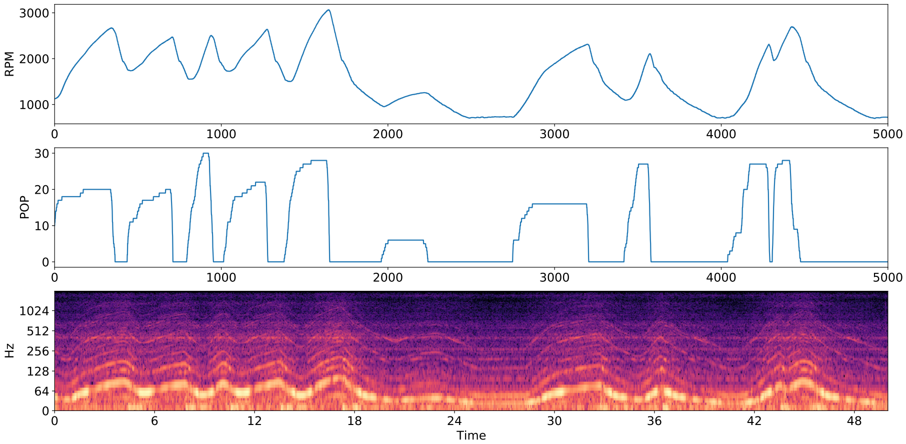

Fig. 1. The RPM, POP, and the spectrum of the original engine sound

2. But we can still notice that the amplitude is higher when the RPM goes much higher. This motivates us to use deep learning to predict the RPM.

Considering the requirement of real-time simulation, we cannot use a complex network architecture since it needs to be set on an embedded system. Therefore, we employ the deep neural network with only two fully-connected layers.

This network takes the RPM, pressure on the pedal (POP), and the first-order delta of RPM and POP as input. The first-order delta of RPM and POP can reflect the acceleration of the vehicle, which is also related to the engine sound, e.g., the engine sounds of driving at 80 MPH and accelerating to 80 MPH are different. The features are split into several frames with a sliding window size of 11, resulting in a 44-dim feature, including consecutive RPMs, POPs, and the delta of PRMs and POPs. The output is the amplitude value at the end of the sliding window on F_0 and its multiples. We train three networks with the same architecture, which predict the amplitude on F_0, $2F_0$, and $3F_0$, respectively.

After the amplitude of each multiple of F_0 is predicted, we add this amplitude to the amplitude generated by the sample-based method. Next, we employ the GLOLA to produce a new engine sound whose amplitude on the multiples of F_0 is more accurate.

4 Experiments and Results

4.1 Dataset and pre-processing

We evaluate the proposed method on several indoor recordings of engine sound from a sedan car of model MG3 manufactured by SAIC Motor that contains a four-stroke engine. The dataset contains four recordings, and the corresponding PRM and POP signals are also recorded from the system on the vehicle every 10ms. A driver presses or releases the pedal frequently to make sure that each

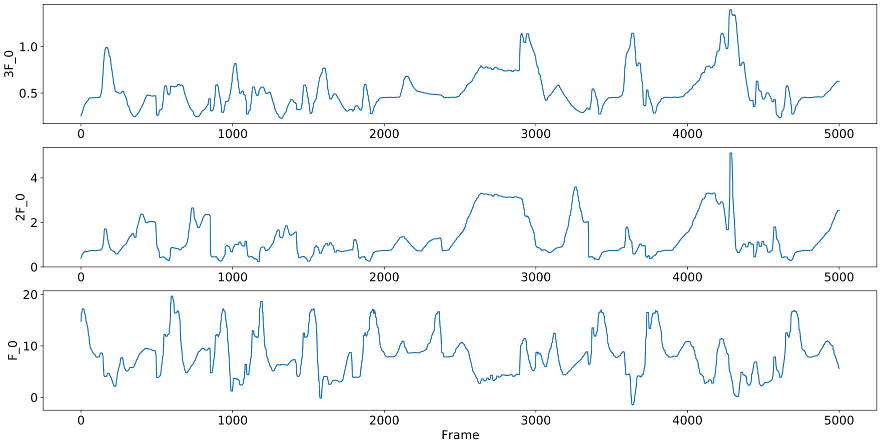

Fig. 2. The amplitude on each multiple of F_0

recording contains as much diverse information as possible. Table 1 shows some metadata of each recording. The sample rate is 44100 Hz, and the duration of each recording is about 10 to 15 minutes. In our experiments, we take the last 50-second audio of the 4-th recording as the testing data for synthesis, and the remaining recordings are used for training the deep neural network.

Table 1. Metadata of the dataset

ID	Duration (s)	range of RPM	range of POP
1	548.76	719.14∼4296.59	0∼37
2	964.91	690.68∼3218.65	0∼36
3	911.35	713.99∼4925.18	0∼55
4	902.14	696.22∼6283.91	0∼228
Total	3327.16	690.68∼6283.91	0∼228

We first downsample the recordings to 4000 Hz. In addition, we compute the STFT with frame length of 100ms and frame shift of 10ms, which means $l = (4000 \times 0.1) = 400$ and $o = (4000 \times 0.09) = 360$ as metioned in algorithm 1.

4.2 Sample-based method

For the sample-based method, we choose the continuous sound samples whose pressure on the pedal (POP) is zero. Since we simulate the engine sound in frame-level and the frame length is 400, we only need no more than $400 \times \frac{6283.91}{690.68} \approx 4000$ idle engine sound samples, which is a 1-second signal.

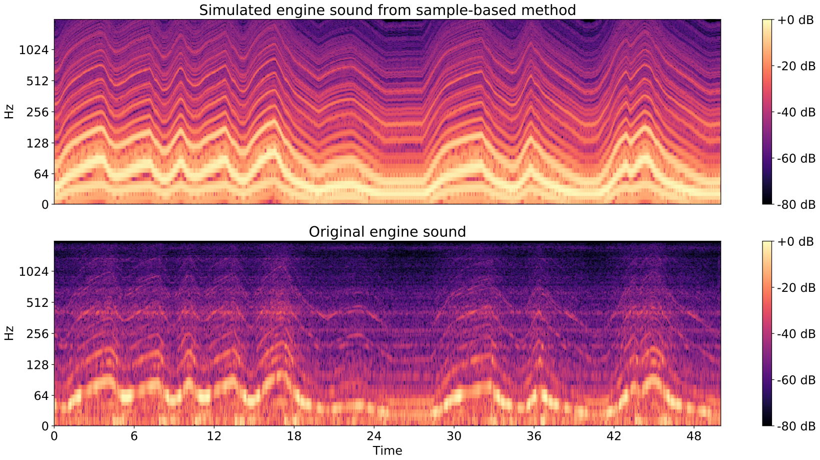

Fig. 3. The specturm of the original engine sound and the simulated engine sound from sample-based method

For a coming frame with an RPM, the idle engine sound samples are first pitch-shifted (resampled) by the ratio between the RPM of the current frame and the idle RPM, which is the $\hat{\mathbf{y}}$ in the algorithm 1. Each time a frame of 400 sound samples is simulated, and only the first 40 samples are played as the frame shift is 40 (10ms). The distance $D(\mathbf{c}, P_{\mathcal{C}}(\mathbf{c}))$ usually becomes stable after 40 iterations, so we stop the iteration after 40 iterations.

Figure 3 shows the spectrograms of the original engine sound and the engine sound simulated by our sample-based method. Although they look similar, the amplitude on the engine pulse frequency F_0 and its multiples cannot match the original amplitude, as shown in figure 4. The energies on the engine pulse frequency are too high to keep the characteristics of the engine.

4.3 Procedural method

As the amplitude of the engine sound simulated by the sample-based method is not entirely accurate, we have explored the use of deep neural networks to predict the amplitude at the engine pulse frequency F_0 and its multiples.

We specifically predict the amplitude at F_0, $2F_0$, and $3F_0$. For each of these multiples of F_0, a separate neural network is trained to predict the amplitude. Consequently, three networks with identical architecture have been trained. Each network consists of two fully connected layers, with both the first and second layers containing 128 neurons each. The input dimension is set at 44, encompassing 11 continuous samples of RPM, POP, and the delta of RPM and POP. The output dimension is 1, representing the amplitude at one of the multiples of F_0, corresponding to the last sample of the input RPM features. Each model undergoes optimization using the Adam optimizer with mean square error (MSE) loss over 100 epochs. The batch size is 128, with a learning rate of 0.001.

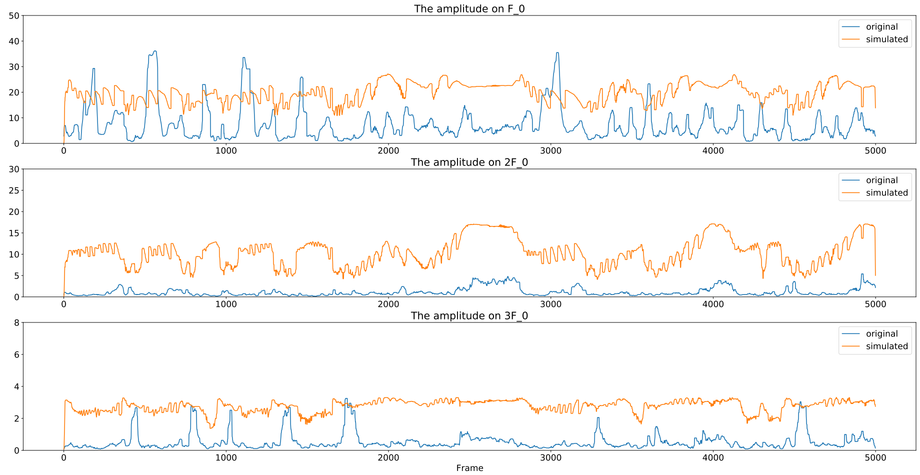

Fig. 4. The amplitude of the original engine sound and the simulated engine sound from sample-based method

Figure 5 displays the spectrograms of the original engine sound alongside the sound simulated by our procedural method. Additionally, Figure 6 presents a comparison of the amplitude between the original engine sound and the sound simulated by the procedural method.

4.4 Hybrid process of synthesis

In the sample-based method, whenever an RPM signal is received every 10ms, the idle engine sound is resampled according to this RPM and then converted to **a** using FFT.

These are then combined with the preceding 10 samples of RPM and POP, and the delta of these samples is computed. These features are subsequently concatenated to form the input for the neural network. Three distinct neural networks then predict three amplitude values separately. These predicted amplitude values are added to **a** to enhance the accuracy of the amplitude on the multiples of F_0.

The overall simulation process remains consistent with the algorithm outlined in 1, with the exception that the three predicted amplitude values are now added to **a**.

Figure 7 illustrates the spectrograms of both the original engine sound and the engine sound synthesized by this hybrid process. The result is notably more similar to the original when compared with the sample-based method alone. The synthesized samples are available on GitHub[1].

[1] https://github.com/karfim/EngineSound

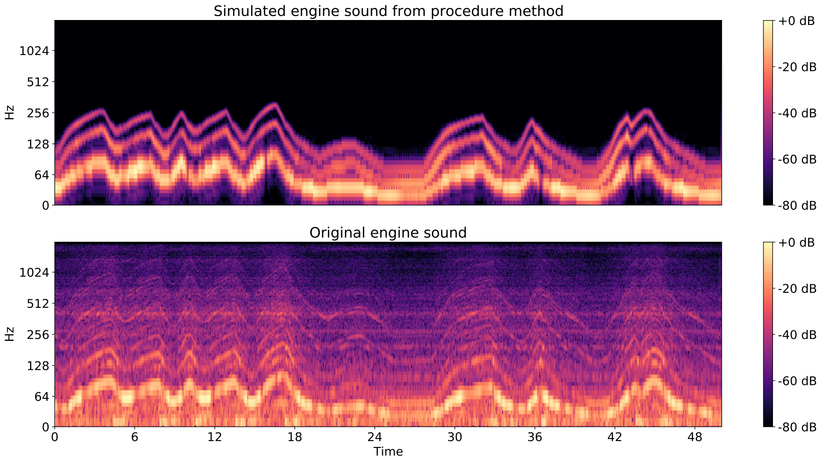

Fig. 5. The specturm of the original engine sound and the simulated engine sound from procedure method

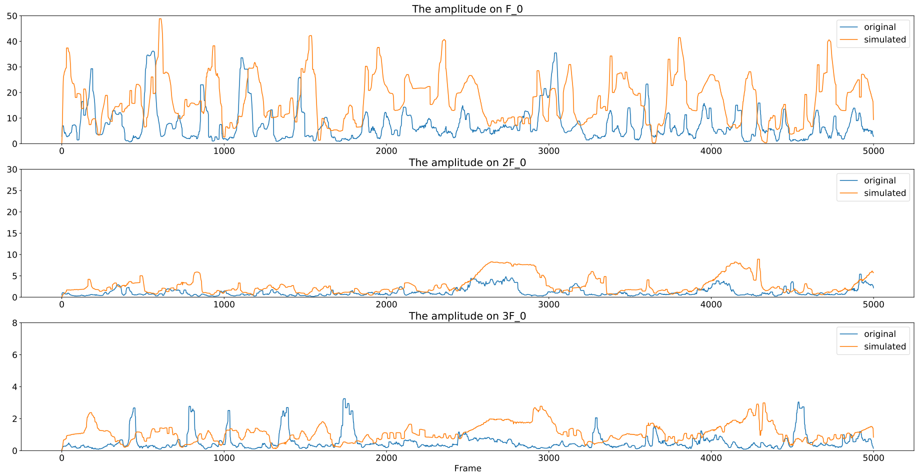

Fig. 6. The amplitude of the original engine sound and the simulated engine sound from procedure method

4.5 Time cost

During the simulation process, all experiments are conducted on an Inter Xeon Silver CPU with a single core of 2.2GHz. The pitch shift process and deep neural network prediction take 0.4ms and 0.8ms. The GLOLA takes 6∼7ms for a frame after 40 iterations. Therefore, the total time cost is about 8ms, and it is smaller than the frame shift of 10ms. This means that it is possible to simulate the engine sound in real-time. In practice, we can use a large frame shift, e.g., 50∼100ms,

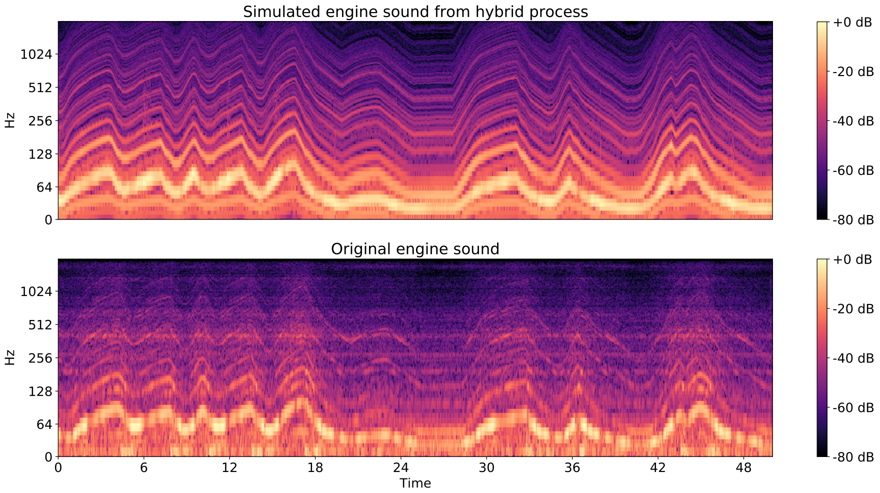

Fig. 7. The specturm of the original engine sound and the simulated engine sound from hybrid process

as engine sound does not need such a high resolution as the speech does. Thus the time cost will be more negligible.

5 Conclusions

In this paper, a hybrid method for engine sound synthesis is proposed, which consists of the sample-based and procedure methods. For the sample-based method, the engine sound can be synthesized in frame-level only given RPM and a 1-second idel engine sound. Next, the amplitude on the engine pulse frequency F_0 and its multiples can be refined by the DNN-based procedure methods. In this procedure method, several deep neural networks predicts the amplitude on the multiples of F_0, and these amplitude values will be added to the spectrum of the resampled signal. Finally, we propose the GLOLA to synthesize the signal in frame-level without any clicks since we incorporate the GLA into the process of STFT/iSTFT. Therefore, no further action is needed to perform clicking cancellation. Also, this method is very fast, which has good potential to synthesize the engine sound with a neural network on the vehicles in real-time. In the future, we are going to collect more data to train a better network. Besides, some techniques, such as quantization and network pruning, can be used to compress the model to further reduce the model size. We will also investigate how to predict the RPM signal from other captured information, e.g., vehicle speed, and then perform the engine sound synthesis using the estimated RPM signal, which is more suitable to the real applications.

Acknowledgements Thanks Hao Li to contribute his own car in data collection.

References

1. D. A. Heitbrink and S. Cable. Design of a driving simulation sound engine. In *Driving Simulation Conference, North America 2007*, 2007.
2. James J Gibson and Laurence E Crooks. A theoretical field-analysis of automobile-driving. *The American journal of psychology*, 51(3):453–471, 1938.
3. Candida Castro. *Human factors of visual and cognitive performance in driving.* CRC Press, 2008.
4. Robert C McLane and Walter W Wierwille. The influence of motion and audio cues on driver performance in an automobile simulator. *Human factors*, 17(5):488–501, 1975.
5. Natasha Merat and Hamish Jamson. A driving simulator study to examine the role of vehicle acoustics on drivers' speed perception. In *Sixth International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design*, 2011.
6. Mark S Horswill and Annaliese M Plooy. Auditory feedback influences perceived driving speeds. *Perception*, 37(7):1037–1043, 2008.
7. Sebastien Denjean, Vincent Roussarie, Richard Kronland-Martinet, Solvi Ystad, and Jean-Luc Velay. How does interior car noise alter driver's perception of motion? multisensory integration in speed perception. In *Acoustics 2012*, 2012.
8. Leonard Evans. Speed estimation from a moving automobile. *Ergonomics*, 13(2):219–230, 1970.
9. Mark S Horswill and Frank P McKenna. The development, validation, and application of a video-based technique for measuring an everyday risk-taking behavior: Drivers' speed choice. *Journal of Applied psychology*, 84(6):977, 1999.
10. May Jorella Lazaro, Sungho Kim, Minsik Choi, Kichang Kim, Dongchul Park, Soyoun Moon, and Myung Hwan Yun. Design and evaluation of electric vehicle sound using granular synthesis. *Journal of the Audio Engineering Society*, 70(4):294–304, 2022.
11. Stefanie M Faas and Martin Baumann. Pedestrian assessment: Is displaying automated driving mode in self-driving vehicles as relevant as emitting an engine sound in electric vehicles? *Applied ergonomics*, 94:103425, 2021.
12. Enes Karaaslan, Mehdi Noori, JaeYoung Lee, Ling Wang, Omer Tatari, and Mohamed Abdel-Aty. Modeling the effect of electric vehicle adoption on pedestrian traffic safety: An agent-based approach. *Transportation Research Part C: Emerging Technologies*, 93:198–210, 2018.
13. Michael S Wogalter, Rachelle N Ornan, Raymond W Lim, and M Ryan Chipley. On the risk of quiet vehicles to pedestrians and drivers. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 45, pages 1685–1688. SAGE Publications Sage CA: Los Angeles, CA, 2001.
14. Verena Wagner-Hartl, Bernhard Graf, Markus Resch, and Paco Langjahr. Subjective evaluation of ev sounds: A human-centered approach. In *Human Systems Engineering and Design: Proceedings of the 1st International Conference on Human Systems Engineering and Design (IHSED2018): Future Trends and Applications, October 25-27, 2018, CHU-Université de Reims Champagne-Ardenne, France 1*, pages 10–15. Springer, 2019.

15. Florian Doleschal and Jesko L Verhey. Pleasantness and magnitude of tonal content of electric vehicle interior sounds containing subharmonics. *Applied Acoustics*, 185:108442, 2022.
16. Sang-Kwon Lee, Gun-Hee Lee, and Jiseon Back. Development of sound-quality indexes in a car cabin owing to the acoustic characteristics of absorption materials. *Applied Acoustics*, 143:125–140, 2019.
17. Jae Hyuk Park, Hansol Park, and Yeon June Kang. A study on sound quality of vehicle engine sportiness using factor analysis. *Journal of Mechanical Science & Technology*, 34(9), 2020.
18. Kun Qian, Zhichao Hou, and Dengke Sun. Sound quality estimation of electric vehicles based on ga-bp artificial neural networks. *Applied Sciences*, 10(16):5567, 2020.
19. Stefano Baldan, Helene Lachambre, Stefano Delle Monache, and Patrick Boussard. Physically informed car engine sound synthesis for virtual and augmented environments. In *IEEE 2nd VR Workshop on Sonic Interactions for Virtual Environments*, 2015.
20. Heather Konet, Manabu Sato, Todd Schiller, Andy Christensen, Toshiyuki Tabata, and Tsuyoshi Kanuma. Development of approaching vehicle sound for pedestrians (vsp) for quiet electric vehicles. *SAE International Journal of Engines*, 4(1):1217–1224, 2011.
21. Thomas Kuppers. Results of a structured development process for electric vehicle target sounds. In *Aachen Acoustic Colloquium 2012*, pages 63–71, 2012.
22. Oliver Engler, Marcus Hofmann, Roman Mikus, and Torsten Hirrle. Mercedes-benz sls amg coupé electric drive nvh development and sound design of an electric sports car. In *15. Internationales Stuttgarter Symposium: Automobil-und Motorentechnik*, pages 1295–1309. Springer, 2015.
23. Jaeyoung Lee, Junwoo Lee, Dooil Choi, and Jongin Jung. String engine sound generation method based on wavetable synthesizer. In *Audio Engineering Society Convention 154*, 2023.
24. Scott A Amman and Manohar Das. An efficient technique for modeling and synthesis of automotive engine sounds. *IEEE Transactions on Industrial Electronics*, 48(1):225–234, 2001.
25. T Janse Van Rensburg, MA Van Wyk, AT Potgieter, and W-H Steeb. Phase vocoder technology for the simulation of engine sound. *International Journal of Modern Physics C*, 17(05):721–731, 2006.
26. Jan Jagla, Julien Maillard, and Nadine Martin. Sample-based engine noise synthesis using an enhanced pitch-synchronous overlap-and-add method. *The Journal of the Acoustical Society of America*, 132(5):3098–3108, 2012.
27. Min Dongki, Park Buhm, and Park Junhong. Artificial engine sound synthesis method for modification of the acoustic characteristics of electric vehicles. *Shock and Vibration*, 2018:1–8, 2018.
28. Jianghua Fu, Chunrong Zhu, and Jintao Su. Research on the design method of pure electric vehicle acceleration motion sense sound simulation system. *Applied Sciences*, 13(1):147, 2022.
29. Daniel Griffin and Jae Lim. Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243, 1984.
30. Shuang Wu. Engine sound simulation and generation in driving simulator. Master's thesis, Missouri University of Science and Technology, 2016.

31. Yoshiki Masuyama, Kohei Yatabe, Yuma Koizumi, Yasuhiro Oikawa, and Noboru Harada. Deep griffin–lim iteration. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 61–65. IEEE, 2019.
32. Jonathan Le Roux, Hirokazu Kameoka, Nobutaka Ono, and Shigeki Sagayama. Fast signal reconstruction from magnitude stft spectrogram based on spectrogram consistency. In *Proc. DAFx*, volume 10, pages 397–403, 2010.
33. Xinglei Zhu, Gerald T Beauregard, and Lonce L Wyse. Real-time signal estimation from modified short-time fourier transform magnitude spectra. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(5):1645–1653, 2007.