KU LEUVEN

STADIUS
Center for Dynamical Systems, Signal Processing and Data Analytics

Citation/Reference	Damiano S., Bondi, L., Guntoro, A., van Waterschoot T. (2024) **A Framework for the Acoustic Simulation of Passing Vehicles Using Variable Length Delay Lines** *EURASIP Journal on Audio, Speech and Music Processing*, vol. 2024, no. 49
Archived version	Author manuscript: the content is identical to the content of the published paper, but without the final typesetting by the publisher
Published version	https://asmp-eurasipjournals.springeropen.com/articles/10.1186/s13636-024-00372-4
Journal homepage	https://asmp-eurasipjournals.springeropen.com
Author contact	stefano.damiano@esat.kuleuven.be

(article begins on next page)

A Framework for the Acoustic Simulation of Passing Vehicles Using Variable Length Delay Lines

Stefano Damiano[1*], Luca Bondi[3], Andre Guntoro[2], Toon van Waterschoot[1]

[1*]Department of Electrical Engineering (ESAT), STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics, KU Leuven, Kasteelpark Arenberg 10, Leuven, 3001, Belgium.
[2]Robert Bosch LLC, 2555 Smallman St, Pittsburgh, PA 15222,United States.
[3]Robert Bosch GmbH, Robert-Bosch-Campus 1, Renningen, 71282,Germany.

*Corresponding author(s). E-mail(s): stefano.damiano@esat.kuleuven.be;
Contributing authors: luca.bondi@us.bosch.com; andre.guntoro@de.bosch.com; toon.vanwaterschoot@esat.kuleuven.be;

Abstract

The sound produced by vehicles driving on roadways constitutes one of the dominant noise sources in urban areas. The impact of traffic noise on human activities and the related investigation on modeling, assessment and abatement strategies fueled the research on the simulation of the sound produced by individual passing vehicles. Simulators enable in fact to promote a perceptual assessment of the nature of traffic noise, and of the impact of single road agents on the overall soundscape. In this work we present *TrafficSoundSim*, an open-source framework for the acoustic simulation of vehicles transiting on a road. We first discuss the generation of the sound signal produced by a vehicle, represented as a combination of road/tire interaction noise and engine noise. We then introduce a propagation model based on the use of variable length delay lines, allowing to simulate acoustic propagation and Doppler effect. The proposed simulator incorporates the effect of air absorption and ground reflection, modelled via complex-valued reflection coefficients dependent on the road surface impedance,

as well as a model of the directivity of sound sources representing the passing vehicles. The source signal generation and the propagation stages are decoupled, and all effects are implemented using finite impulse response filters. Moreover, no recorded data is required to run the simulation, making the framework flexible and independent on data availability. Finally, to validate the framework capability to accurately simulate passing vehicles, a comparison between synthetic and recorded pass-by events is presented. The validation shows that sounds generated with the proposed method achieve a good match with recorded events in terms of power spectral density and psychoacoustics metrics, as well as a perceptually plausible result.

Keywords: acoustic simulation, traffic noise synthesis, variable length delay lines, road acoustic propagation, Doppler effect

1 Introduction

The noise produced by public and private transportation systems is a prominent feature of the urban soundscape. In any major city, vehicles such as cars, trucks, trains, aircraft etc. are in fact the source of an incessant background noise that, on one hand, has a major impact on the health and quality of life of people [19], and on the other is a source of information on events happening in urban areas [22]. To improve the livability of cities, several noise abatement techniques have been proposed. Since road traffic constitutes the prominent noise source, solutions to mitigate it have been specifically designed via the production of quieter engines and the manufacture of tires and asphalt mixtures designed to limit the rolling sound. Simultaneously, urban studies aiming at diverting the traffic flow to reduce vehicle density in urban areas, as well as the installation of sound barriers, have proven effective to reduce the overall sound intensity [19, 39].

To ease the study and assessment of traffic noise, engineering methods that mostly aim at the evaluation of average sound pressure levels (SPL) [31, 32] have been developed. Whilst effective for estimating the long-term averaged acoustic pressure,

such tools are usually insufficient to capture the complexity of the highly non-stationary urban soundscape, including individual vehicle pass-bys, impulsive noises and emergency sounds such as horns and sirens.

For this reason, auralization techniques have been exploited to provide the opportunity to *listen* to the sound that would be received by a listener in a specific location [20–22, 26, 39, 40]. According to the definition provided in [46], auralization refers to the technique of creating audible sound from numeric data. The auralization of a virtual environment involves three main steps: i) the definition of a virtual acoustic scene containing a set of sound sources and a listener; ii) the simulation of sound propagation given the defined scene and an acoustic propagation model; iii) (optionally) the spatial audio rendering to reproduce the sound perceived by the listener (i.e., binaural rendering). Although auralization techniques are largely exploited in indoor acoustics studies and in the early phases of the design of closed spaces (e.g. concert halls, recording studios, public spaces), the lack of well established techniques for the auralization of outdoor environments poses a limitation to its adoption for the analysis of urban areas and the assessment of traffic noise. In recent years, many studies have proposed methodologies to implement the different components involved in the auralization of traffic noise, from the definition of source signals to emulate passing vehicles [4, 20, 22, 28, 39, 40, 43], to the development of models for acoustic propagation on a road [10, 15, 16, 28, 39], to the design of a spatial audio reproduction setup [15, 22, 33].

Auralization tools can be useful both to reproduce the urban acoustic soundscape and to extract some information that it contains. The analysis of traffic noise, in particular, has been exploited in the design of algorithms to enhance the perceptual abilities of autonomous vehicles by means of acoustic sensing [38] and in acoustic vehicle counting systems [9, 14]. In these use-cases, auralization tools can become useful to generate

data to train data-driven techniques [9]. The reproduction of traffic soundscapes, instead, has found applications in virtual reality and computer games [22].

With the goal of developing a traffic sound auralization tool, in our previous work [10] we introduced an open-source simulator of acoustic propagation on a road based on the use of variable length delay lines. A delay line allows to effectively simulate the sound produced by a moving source with arbitrary trajectory and speed as well as the Doppler effect, particularly relevant in road scenarios due to the relatively high speed that vehicles can reach. The simulator incorporates a model of air absorption and asphalt reflection, implemented via finite impulse response (FIR) filters, and supports user-defined source signals, trajectories and speeds. Compared to more complex propagation models relying on wave-based methods, *pyroadacoustics* is based on geometrical acoustics and comes with a lower computational complexity.

This work extends [10] by introducing *TrafficSoundSim*, a framework for the acoustic simulation of passing vehicles. *TrafficSoundSim* is composed of two independent modules: the source synthesis module, responsible for the generation of source signals to represent vehicular sources, and the propagation module, responsible for the simulation of acoustic propagation on a road. For the design of the source synthesis module, we adopt a sound source description inspired by the Harmonoise model [30], according to which the sound produced by road vehicles can be split into two main contributions, namely road/tire interaction (or rolling) sound and engine (or propulsion) sound. Every road vehicle can then be acoustically described by two vertically stacked point sources, emitting a mixture of rolling and propulsion sound, both broadband in nature.

In Harmonoise, the sound produced by a passing vehicle is synthesized in short-time windows, within which the source is assumed to be static and propagation effects are simulated jointly with the signal generation. This model comes with two main drawbacks. First, whilst effective for characterizing road/tire interaction noise,

the broadband noise description disregards both the *texture* of a real engine sound, having rumbling and rattling components, as well as prominent tonal characteristics connected to the engine load and speed, and to the speed of the vehicle.

To overcome this limitation, similarly to [22] we introduce a more accurate engine sound generation procedure based on physical modelling, known as the Baldan model [4], on top the Harmonoise method. Both the original Harmonoise and the extendend Harmonoise and Baldan source models have been implemented in the simulator, and a choice between them can be made based on psychoacoustics considerations and on the target application. Second, the joint source signal and acoustic propagation synthesis that is performed within Harmonoise inevitably leads to artifacts, caused by the source *jumping* between successive discrete positions corresponding to the short-time windows used in the synthesis process. Moreover, the simulation of a signal received by multiple receivers entails performing the entire simulation from scratch per each receiver. *TrafficSoundSim*, on the contrary, is designed with a modular structure to keep the source synthesis and propagation modules decoupled. On the one hand, this allows to generate the stationary source signal once, and to apply the propagation effects multiple times only in presence of multiple microphones. On the other hand, the use of delay lines reduces the artifacts as the source position is updated sample by sample.

As an additional contribution, we extend the *pyroadacoustics* propagation module: first, we introduce an improved implementation of the ground reflection filtering inspired by the Delany Bazley model [12]. Finally, to simulate the directivity of moving sound sources representing passing vehicles, we introduce an FIR implementation of the Harmonoise directivity function. Differently from the original Harmonoise model, where directivity is implemented at the source signal generation time, the proposed implementation is part of the propagation module, as it represents an effect caused

by the movement of the source. This way, the modularity principle of the proposed framework is respected.

The rest of the paper is organized as follows. In Sec. 2 we discuss the existing literature on the auralization of passing vehicles, with a focus on the Harmonoise and Baldan models that constitute the backbone of *TrafficSoundSim*. In Sec. 3 we introduce the passing vehicles simulation framework by describing the source synthesis module (Sec. 3.1), the propagation module (Sec. 3.2) and the complete simulation setup (Sec. 3.3). We then show how simulations compare to real pass-by recordings to discuss the effectiveness of the framework in Sec. 4, and we finally draw conclusions in Sec. 5.

2 Related Work

The existing frameworks for the auralization of vehicle pass-by events consist of three main building blocks, responsible for the generation of the source signal, the propagation of sound from the source to the receiver, and (optionally) the spatial sound reproduction (i.e., binaural rendering). Since in this work we are only interested in the first two, in this section we focus on summarizing the main existing methods adopted for their design.

Several models have been proposed in the literature to generate synthetic source signals for the auralization of passing vehicles. According to the ISO-362 standard, two separate components contribute to the sound produced by moving vehicles, namely the engine (or propulsion) noise, and the road/tire interaction noise [6, 23]. These contributions are usually simulated separately using different techniques, that can be categorized into physical modelling synthesis [3–5, 27, 34], spectral-shaping [33, 40], recording-based synthesis [16, 17, 36, 39, 43], engineering methods [31, 32], and combinations of those [15, 22, 24, 28].

Physical modelling techniques have been used for the generation of both road-/tire [5] and propulsion [4] noise. These methods consist in the definition of a parametric model of the physical behaviour of the sound generation process. For road-/tire interaction noise, the SPERoN and HYRoNE models have been proposed [5], based on deterministic calculations of the contact forces between the rubber surface of the tire and the asphalt road, eventually complemented with a statistical description of the non-linear dependency of the contact pressure on the road profile. Notwithstanding their elevate accuracy, these models require a difficult parameter tuning and come with a high complexity that hinders their practical usage for auralization. For propulsion noise, a physical model of the four-stroke engine has been developed, known as the Baldan model [4]. This relatively small model involves few parameters that describe the engine physics (e.g. number of cylinders, engine speed and load) and can be used to emulate the sound produced by different vehicle types (cars, trucks, motorbikes) with real-time capabilities.

Spectral-shaping synthesis techniques aim at designing the spectrum of the target signal using additive or subtractive synthesis, or a combination of both (spectral modelling synthesis). In [40] the propulsion noise component is generated using spectral modeling synthesis: this model employs a set of parameters to describe driving dynamics (engine speed, load and gear) and allows to simulate accelerating vehicles. As a main drawback, recordings performed in a controlled environment are needed to obtain the propulsion noise synthesizer parameters; this limits the applicability of the model to specific vehicle types for which high quality recordings are available.

Recording-based approaches rely on the analysis of one (or a set of) real recorded pass-by event(s) to generate a synthetic source signal to be used for simulation purpose. Two main techniques have been adopted. The first one consists in extracting the source signal by inverse-filtering a recorded pass-by event [16] to remove the effects of propagation, air absorption, ground reflection, source directivity and Doppler shift,

and thus obtain a stationary signal that contains both contributions from road/tire interaction and propulsion. This method depends on the availability of a pass-by recording from which the inverse filters parameters can be extracted, and the generated signal is tailored to match the recording conditions. To extend the design space, granular synthesis has been proposed [36]. This method relied on the analysis of a large set of recordings to extract the grains needed to synthesize different vehicles, roads, speeds and gears. The lack of large recording datasets constitutes the main drawback of these techniques.

Engineering models, such as Nord-2000 [32] and Harmonoise [31], aim at estimating the average sound pressure levels produced by passing vehicles. These methods describe both road/tire and engine noise using empirical equations, whose parameters have been extracted by regression analysis performed on a large dataset of traffic sounds, and are available in tables. The two source signal components are synthesized by filtering broadband noise using one-third octave filterbanks to match the sound pressure levels predicted by the aforementioned equations. These simple models are recording-free and produce accurate results for the simulation of the road/tire interaction sound. However, the averaging operation causes the engine sound to lack in texture (rattling, rumbling and tonal components) compared to other methods.

Finally, hybrid approaches have also been proposed. In [24], the Harmonoise model is used to simulate the road/tire interaction noise, and a modified version of the Baldan model [4] is adopted for engine simulation: this combination leads to an improved accuracy as compared to the standard Harmonoise technique, without significantly impacting the computational load. In [15], instead, the inverse filtering method [16] is combined with the Baldan [4] model with the same purpose.

For the simulation of acoustic propagation, existing simulators targeting traffic auralization rely either on frequency-domain processing [31, 33], time-domain processing [10, 24, 40], or a combination of both [21, 36].

Frequency-domain techniques, introduced in [31, 33], are designed with the aim of simulating propagation effects including geometrical spreading, air absorption, attenuated ground reflections and source directivity by analysing their individual impact on the sound pressure levels produced by the source in one-third octave frequency bands. The effects are applied as sound pressure level correction factors, depending on the relative position of the source and the receiver, at the source signal generation time. Sound synthesis is performed on short-time windows in which the source is assumed to be static.

Different techniques use a combination of time- and frequency-domain processing to consider the Doppler effect [21, 36]. In [36] the synthesized time-domain source signal is fed to a fractional delay line to model acoustic delay and Doppler effect. The delayed signal is then processed via frequency-domain filters to apply frequency-dependent gains to simulate sound attenuation and source directivity.

The last group of propagation methods are entirely based on time-domain processing of the signal generated by the source synthesis module [10, 24, 40]. These methods rely on the use of fractional delay lines to simulate acoustic propagation delay and Doppler effect for both the direct sound and the reflection from the ground surface, whereas the attenuation due to ground and air absorption are implemented as FIR filters in the time domain. These techniques come with a low computational complexity and acoustic propagation is decoupled from the source signal generation.

In our previous work [10] we introduced the open-source *pyroadacoustics* simulator, that implements sound propagation on a road by means of variable length delay lines and FIR filtering. The model, described in Sec. 3.2, allows to simulate the direct sound emitted from an omnidirectional source and the reflection from the ground surface, together with sound attenuation due to spherical propagation and air absorption, and comes with a low computational complexity.

In this paper we extend pyroadacoustics to build an end-to-end framework for the simulation of vehicle pass-by events. *TrafficSoundSim* is implemented as an open-source python package [8] and can be used to simulate the sound produced by both cars and heavy vehicles (HV). This work introduces the following contributions:

1. we propose a *modular* structure that combines state-of-the-art source signal generation methods, namely the Harmonoise [31] road/tire interaction noise model and the Baldan physical engine noise model [4], with a sound propagation module built on the pyroadacoustics simulator. The modularity guarantees that all propagation effects (sound attenuation, air absorption, asphalt reflection, moving source directivity) are independent on the source signal generation, leaving the proposed framework open for extension to account for different source types (e.g., electrical engines).
2. We introduce an improved ground reflection model in pyroadacoustics, based on the *Delany and Bazley* model [12], and show that it results in an improved realism of synthesized sounds in terms of both error and psychoacoustics metrics.
3. We propose an implementation of the Harmonoise moving sound source directivity based on digital filtering performed in the propagation module. This implementation respects the modular structure of the proposed method and enhances extensibility to different sound sources.
4. We introduce in pyroadacoustics an implementation of common directivity patterns for sound sources and receivers.

To provide an overview of the existing traffic sound simulation tools, in Tab. 1 we compare the features and capabilities of some of the discussed state-of-the-art methods, including *TrafficSoundSim*. The strength of the proposed framework lies in the modular, user-controllable and extensible propagation module, designed in a physically accurate way and allowing to generate signals for arbitrary receiver configurations (i.e., arbitrary multi-channel microphone array geometry and directivity).

Feature	Kaczmarek [33]	Maillard et al. [36]	Pieren et al. [40]	Fu et al. [24]	Dreier et al. [15]	*TrafficSoundSim*
Source model	Harmonoise	Rolling+propulsion	Modified Harmonoise	Harmonoise	Rolling+propulsion	Harmonoise
Rolling sound	Additive synth.	Granular synth.	Spec. modeling	Harmonoise	Inv. filtering	Harmonoise
Propulsion sound	NA	Granular synth.	Spec. modeling	Baldan	Baldan	Baldan
Trajectory	Rectilinear	Arbitrary	Arbitrary	Arbitrary	Arbitrary	Arbitrary
Doppler	Yes	Yes	Yes	Yes	Yes	Yes
Reflections	Ground	None	Ground	Ground Buildings	Ground Buildings	Ground
Air absorption	Yes	Yes	Yes	Yes	Yes	Yes
Need recordings	No	Yes	Yes	No	Yes	No
Real-time	Yes	No	No	Yes	Yes	No
Receiver	Binaural	Mono, Stereo Ambisonics, Binaural	Multi-channel	Binaural	Arbitrary	Arbitrary
Open-source	No	No	No	No	No	Yes

Table 1 Comparison of features between state-of-the-art simulators and the proposed framework. Features include the adopted source model, how rolling and propulsion sounds are simulated, the type of trajectory that the source can travel upon, the inclusion of Doppler effect, reflections and air absorption, whether recordings are needed to extract auralization parameters (e.g., for generating source signal), real-time capabilities, allowed receiver geometry configuration and source code availability.

This feature, together with its open-source release, enables *TrafficSoundSim* to be used in several applications ranging from auralization to data generation for training automatic traffic monitoring systems [9]. To validate the model, a comparison between simulated and recorded pass-by events in semi-controlled conditions is shown in Sec. 4, and audio demos are provided as additional files. The comparison has been carried out using traditional error metrics and psychoacoustics sound quality measures, both expressing a close similarity between simulated and recorded signals. Moreover, *TrafficSoundSim* has been successfully used to generate data for training a neural network to target acoustic vehicle counting in [9]. Although this application is relevant to assess the effectiveness of *TrafficSoundSim*, its discussion is outside the scope of this work and we refer to [9] for the details.

3 Framework Description

In this section, we describe the *TrafficSoundSim* framework by first introducing the source synthesis and propagation modules, and then discussing the vehicle pass-by simulation set-up.

3.1 Source signal synthesis

According to the Harmonoise model [31], two main components contribute to the sound produced by a passing vehicle, namely the road/tire interaction noise, caused by the rolling motion of the tires on the rough ground surface, and the propulsion noise, produced by the engine. Following this description, in the source synthesis module of *TrafficSoundSim* we separately generate the two components and combine them to create the source signals to be used as input for the propagation module. Note that the propulsion noise is mostly relevant for cars powered by thermal engines, whereas electric cars are not considered in our discussion. Nevertheless, since the generation of engine and rolling noise is decoupled, this structure would facilitate extending the proposed framework to electric vehicles as well.

3.1.1 Road/tire interaction noise

To synthesize the signal produced by road/tire interaction, we rely on the Harmonoise model [31]. Accordingly, the sound produced by the rolling tires has broadband characteristics (thus tonal components caused by resonances, as well as vibrations produced e.g. by cobblestone pavements or non-smooth ground surfaces are not taken into account). The rolling noise component is generated by filtering white noise using a one-third octave filterbank with $N_b = 27$ bands, with center frequencies from 25 Hz to 10 kHz. The Harmonoise model predicts the SPL emitted in each band as a function of vehicle speed and type (i.e., light vehicles, medium vehicles, heavy vehicles and other vehicles), by means of the empirical equation

$$L_{\mathrm{WR}}(v, f) = a_{\mathrm{R}}(f) + b_{\mathrm{R}}(f) \log(\frac{v}{v_{\mathrm{ref}}}) \,, \tag{1}$$

where f is the frequency, v is the speed of the vehicle, $v_{\mathrm{ref}} = 70\,\mathrm{km/h}$ is a reference speed, and the regression parameters a_{R} and b_{R} are provided in one-third

octave bands in the above mentioned frequency range. Correction coefficients, although not incorporated in the current description for the sake of clarity in presentation, can be introduced to accommodate variations in road surfaces and atmospheric conditions [22]. Moreover, a correction factor depending on the instantaneous source position can be used to model source directivity: however, to keep the source signal generation decoupled from the propagation effects, directivity is implemented in the propagation module in *TrafficSoundSim*. The discrete time-domain road/tire interaction noise signal is then computed as

$$s_{\mathrm{R}}[n] = \sum_{i=1}^{N_b} p_0 10^{\frac{L_{\mathrm{WR}}(v,f_i)}{20}} \cdot s_{\mathrm{noise},i}[n]\,, \tag{2}$$

where $p_0 = 2 \times 10^{-5}\,\mathrm{Pa}$ is the hearing threshold and $s_{\mathrm{noise},i}[n]$ is the white noise in the frequency band i. In practice, the signal $s_{\mathrm{noise},i}[n]$ is obtained by passing a broadband white noise signal $s_{\mathrm{noise}}[n]$ through the above-mentioned one-third octave filterbank. We underline that, differently from [24], we do not apply road surface correction factors, nor directivity correction factors. The latter, in particular, will be implemented in the propagation stage using digital filters, as discussed in Sec. 3.2.2.

3.1.2 Propulsion noise

For the synthesis of propulsion noise, we adopt the Baldan model [4], a physical model of the 4-stroke engine that allows to generate a realistic-sounding engine with a low computational complexity. This method is based on modelling three separate components that are responsible for most of the emitted sound:

1. The intake system is responsible for letting a mixture of air and fuel into the cylinder chamber via a valve that opens at regular time intervals. Acoustically, this is simulated by means of a digital waveguide [44] with fixed feedback on the free end and variable feedback on the valve end, connected to the cylinder.

The intake system receives its primary signal from the cylinder component, and is fed at the free end with a secondary signal consisting of a low-pass filtered white noise, that represents the turbulence in the air/fuel mixture caused by the aspiration from the opening valve. The free end yields the first output of the model, represented by the signal $s_{\text{in}}[n]$.

2. One or more cylinders with periodic motion, together with the fuel ignition mechanism, constitute the core of the engine. These components are controlled by the two main input parameters of the model, namely the engine speed (expressed in revolutions per minute - RPM) and by the engine load, and produce the tonal components of the engine, together with the rattling sounds produced by mechanical vibrations. The cylinders are connected to a second model output that represents the engine vibrations, represented by the signal $s_{\text{vibr}}[n]$.
3. The exhaust system represents the component that expels gas into the environment. It is simulated by a set of digital waveguides that represent the exhaust pipes, the muffler elements and the final outlet, fed with filtered white noise to represent the gas flow, and is connected to the cylinder module. The exhaust yields the third output of the model, represented by the signal $s_{\text{ex}}[n]$. Note that for the simulation of a passing vehicle, the exhaust system constitutes the major contribution to the overall produced engine noise [22].

Finally, the engine sound is obtained by summing the three components as

$$s_{\text{E}}[n] = A_1 s_{\text{in}}[n] + A_2 s_{\text{vibr}}[n] + A_3 s_{\text{ex}}[n]\,, \tag{3}$$

where A_1, A_2 and A_3 represent relative amplitude weights. As the Baldan model does not provide guidelines to tune these parameters, they can be tuned manually according to the desired listening position [22].

3.1.3 Source signal generation

To generate the source signals that will be provided to the propagation module, we combine the road/tire and engine noise components following Harmonoise [22]. Accordingly, passing vehicles can be modelled using two vertically stacked point sources that emit a mixture of road/tire and propulsion noise. The lower source (LS) is located at a height of 0.01 m for all types of vehicles, whereas the higher source (HS) is located at 0.3 m for cars and 0.75 m for heavy vehicles such as trucks and buses. The road-/tire and propulsion signal powers are split with relative weights of $[0.8, 0.2]$ between the two sources, with the LS containing most of road/tire noise power and vice versa, as [22]

$$\begin{aligned} s_{\mathrm{LS}}[n] &= \frac{1}{\sqrt{5}}\tilde{s}_{\mathrm{E}}[n] + \frac{2}{\sqrt{5}}s_{\mathrm{R}}[n] \\ s_{\mathrm{HS}}[n] &= \frac{2}{\sqrt{5}}\tilde{s}_{\mathrm{E}}[n] + \frac{1}{\sqrt{5}}s_{\mathrm{R}}[n]\,, \end{aligned} \tag{4}$$

where $s_{\mathrm{LS}}[n]$ and $s_{\mathrm{HS}}[n]$ represent the time-domain signal assigned to the lower and higher source, respectively, $s_{\mathrm{R}}[n]$ represents the road/tire noise signal generated as in (2) and $\tilde{s}_{\mathrm{E}}[n]$ is a modified version of $s_{\mathrm{E}}[n]$ defined in (3), normalized to the Harmonoise predicted noise power. To combine the signals as in (4), in fact, the relative power of the two components generated from road/tire interaction and propulsion must match the power predicted by Harmonoise.

We normalize the engine noise signal to match the Harmonoise engine power following the procedure described in [22]. First, we compute the engine noise predicted by Harmonoise: similarly to (1), the SPL produced by the engine noise can be computed as

$$L_{\mathrm{WP}}(v, f) = a_{\mathrm{P}}(f) + b_{\mathrm{P}}(f) \log \frac{v - v_{\mathrm{ref}}}{v_{\mathrm{ref}}}\,, \tag{5}$$

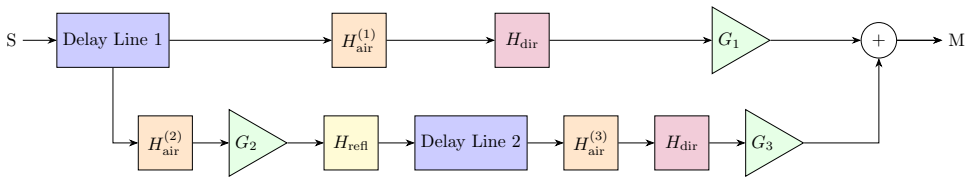

Fig. 1 The architecture of the proposed propagation module: acoustic delay and Doppler effect are simulated using delay lines; air absorption for the paths source-receiver, source-reflection point on ground surface and reflection point-receiver, as well as ground reflection effect are simulated via FIR filters (H_{air} and H_{refl}, respectively); H_{dir} models sound source directivity in accordance with the Harmonoise description; attenuation caused by spherical propagation is modelled via gain elements G_1, G_2, G_3.

where the regression coefficients a_{P} and b_{P} are given in one-third octave bands. We then compute the time-domain Harmonoise engine signal as

$$s_{\text{P}}[n] = \sum_{i=1}^{N_b} p_0 10^{\frac{L_{\text{WP}}(v,f_i)}{20}} \cdot s_{\text{noise},i}[n]\,, \tag{6}$$

where $s_{\text{noise},i}[n]$ is a subband white noise signal in the frequency band i, obtained by passing $s_{\text{noise}}[n]$ through a one-third octave filterbank with $N_b = 27$ bands, similarly to (2). Although the signal $s_{\text{P}}[n]$ can be directly used as engine noise source signal, it lacks the texture of a real engine: being a sum of filtered white noise components, it does not contain tonal components nor rattling and rumbling sounds produced by mechanical vibrations [22]. In *TrafficSoundSim*, instead, we exploit $s_{\text{P}}[n]$ to calibrate the power of the more accurate $s_{\text{E}}[n]$ to match the Harmonoise predictions and be able to apply (4). Therefore, we compute the normalized engine noise $\tilde{s}_{\text{E}}[n]$ as

$$\tilde{s}_{\text{E}}[n] = s_{\text{E}}[n] \cdot \frac{\sum_{i=0}^{N_s} |s_{\text{P}}[i]|^2}{\sum_{i=0}^{N_s} |s_{\text{E}}[i]|^2} \tag{7}$$

where N_s is the total length of the signals s_{P} and s_{E}. The signals obtained using (4) represent the output of the source synthesis module, and are passed over to the propagation module.

3.2 Sound propagation

The proposed sound propagation module, whose architecture is depicted in Fig. 1, is built on top of our open-source *pyroadacoustics* simulator [10], available at [7]. *Pyroadacoustics* is designed to emulate the propagation of the sound emitted by a single source moving with arbitrary speed and trajectory on a non-urban road, in which the main reflecting surface is represented by the ground. An arbitrary number of stationary virtual microphones, arbitrarily distributed in the space, provide the output of the simulator (i.e., they constitute the listening positions).

In *pyroadacoustics* the propagation delays, together with the Doppler effect, are implemented by means of variable length delay lines. Delay lines are simple signal processing blocks used to introduce a delay on the input signal they are provided with, and can be implemented using a circular buffer with a defined length N, as depicted in Fig. 2. The input signal is written in the buffer at the position of a write-pointer (*wptr*), that gets incremented by one every computation instant; the output is read at the position of a read-pointer, delayed by M samples with respect to the write-pointer. The delay M can vary over time as long as the condition $M \leq N$ is met, yielding a *variable length* delay line, that can be used to simulate acoustic propagation when relative motion is present between a sound source and a receiver. Given a signal $x[n]$ emitted by a moving source, whose instantaneous position is denoted by $\mathbf{r'}[n]$, and the signal $y[n]$ received by a static listener at position $\mathbf{r}$, the corresponding delay $M[n]$ is given by

$$M[n] = \frac{\|\mathbf{r} - \mathbf{r'}[n]\|_2}{c} f_s \,, \tag{8}$$

where f_s denotes the sampling frequency, c is the speed of sound in the air, and $\|\cdot\|_2$ is the vector ℓ_2-norm. For the clarity of the presentation, the time index n will be omitted from the expression of the delay $M[n]$, that will anyhow always be considered time-dependent. Since M can take non-integer values in a real scenario (e.g. when the

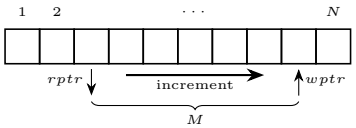

Fig. 2 Circular buffer implementation of an M-sample delay line, with read (*rptr*) and write (*wptr*) pointers moving in the direction specified by the arrow [10].

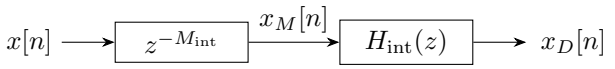

Fig. 3 Implementation of delay line with fractional delay $M = M_{\text{int}} + \Delta$, where $M_{\text{int}} = \lfloor M \rfloor$ is the integer part and $\Delta \in [0, 1)$ is the fractional part: the integer delay line of M_{int} samples is followed by an interpolator H_{int} [10].

relative distance between a moving source and a receiver varies continuously), interpolated reads can be performed using an interpolation filter cascaded to the delay element, as depicted in Fig. 3. Linear, all-pass and sinc interpolation filters have been implemented in *pyroadacoustics*. Note that, since the position of the read and the write-pointer is tied to a physical model of sound propagation, a minimum (and maximum) delay interval is always present between them; thus, the read-pointer never surpasses the write-pointer [44]. The variable delay, together with the interpolated reads, guarantees that the Doppler effect is simulated without any further processing, making this architecture effective for the accurate simulation of fast moving sources. In *pyroadacoustics* two delay lines are employed: the first one has two read-pointers and is used to model the direct path and the path connecting the source position to the reflection point on the ground surface. The second one, instead, is used to model the path from the reflection point to the microphone position.

The delayed and interpolated signal $x_D[n]$ that exits a delay line is passed through a FIR filter H_{air} that emulates the distance- and frequency-dependent effect of air absorption, and an attenuation gain G is added to account for spherical propagation (see Fig. 4). The air absorption filter is designed from its amplitude response, derived from the air absorption model described in the ISO 9613-1 standard [29], as discussed in [10].

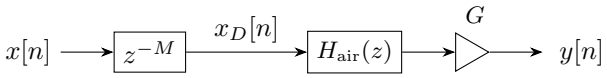

Fig. 4 Simulation scheme of acoustic propagation: the input signal $x[n]$ is passed through a variable length delay line, an attenuation filter to emulate air absorption, and a gain element to account for spherical propagation [10].

3.2.1 Ground effect

The sound reflection produced by the ground surface is modelled in the propagation module, as shown in Fig. 1. In particular, sound reflection is implemented as a time-variant FIR filter H_{refl}, that takes as input the sound emitted by the source, delayed and filtered according to the distance between the source and the reflection point, computed using the law of reflection. In [10] the filter design originates from the real-valued, normal-incidence reflection coefficient: although the method is lightweight and allows to characterize several ground surface materials for which normal-incidence reflection coefficients are available [41], it neglects the phase effects caused by the complex impedance of rigid surfaces. To overcome this limitation we hereby introduce an improved model of ground reflection, inspired by [26] and based on the *Delany and Bazley* model [2, 12].

To model the ground effect we assume a single, specular reflection path (i.e., scattering and diffusion are not included) and a locally reacting ground surface, implying that the complex-valued ground impedance Z does not depend on the angle of incidence ϑ. The plane wave reflection coefficient is defined as [35]

$$R(\vartheta) = \frac{Z\cos\vartheta - Z_0}{Z\cos\vartheta + Z_0}\,, \tag{9}$$

where Z_0 is the characteristic impedance of air and Z is the complex-valued impedance of the road surface, that can be expressed using the single-parameter Delany Bazley model as [12]

$$Z = 1 + 9.08\left(\frac{f}{\sigma}\right)^{-0.75} - 11.9\jmath\left(\frac{f}{\sigma}\right)^{-0.73}\,, \tag{10}$$

where $\jmath$ denotes the imaginary unit, while the parameter σ indicates the flow-resistance and can be computed for different materials. The spherical wave reflection factor can then be expressed as [2]

$$Q(\vartheta) = R(\vartheta)\,(1 - R(\vartheta))\,F(w)\,, \tag{11}$$

where w represents the numerical distance

$$w = \sqrt{\jmath kd\left(1 + \frac{1}{Z}\cos\vartheta - \sqrt{1 - \left(\frac{1}{Z}\right)^2}\sin\vartheta\right)}\,, \tag{12}$$

d being the distance between the source and the reflection point and $k = 2\pi f/c$ being the wave number. Finally, the function $F(w)$ of the numerical distance w is defined as

$$F(w) = 1 + \jmath\sqrt{\pi}we^{-w^2}\mathrm{erfc}(-\jmath w)\,, \tag{13}$$

where we used the complementary error function

$$\mathrm{erfc}(\xi) = 1 - \frac{2}{\sqrt{\pi}}\int_0^{\xi} e^{-\chi^2}\,d\chi\,. \tag{14}$$

This method allows us to compute the complex-valued reflection coefficient $Q(\vartheta)$ for frequencies in the range $[0, f_s/2]$. We then implement the ground reflection effect using a FIR filter, by interpreting $Q(\vartheta)$ as the frequency response of the desired filter, and using the inverse Fourier transform to obtain its impulse response.

3.2.2 Directivity model

The Harmonoise method includes a model of horizontal source directivity, representing the *horn* effect produced by the rolling wheels, and vertical source directivity,

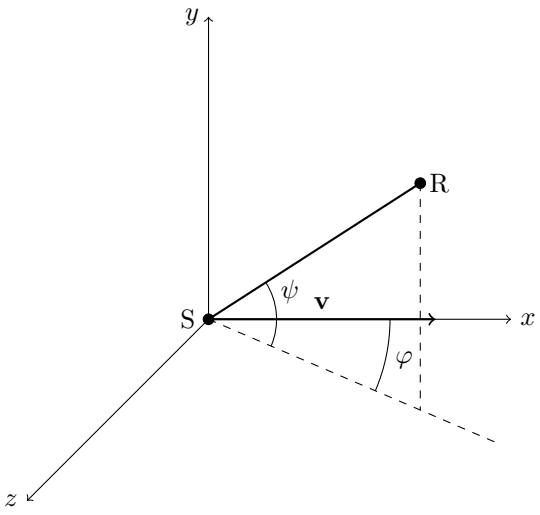

Fig. 5 Geometry for the derivation of the directivity functions: S denotes the sound source, moving at speed **v**, while R denotes the static receiver.

to account for the screening effect caused by the car body [31]. Both the directivity effects are defined as SPL correction factors applied during the source signal generation, governed by closed-form functions of the horizontal angle φ and vertical angle ψ depicted in Fig. 5. Since the angles φ, ψ depend on the relative position of the car and receiver, that varies over time, the directivity computation is performed in each synthesis-window, and is thus entangled to the signal generation.

To keep the source synthesis and propagation modules uncoupled, we propose an implementation of horizontal directivity based on FIR filtering in the propagation module (see Fig. 1). Since we consider a scenario where the receiver is located at the side of a road, at a relatively small height (smaller than 3 m), the vertical directivity has a minor contribution on the overall signal and, hence, it is currently not included in *TrafficSoundSim*. Note that, to consider sensors located on top of the road or listeners at higher positions (e.g., on high buildings or aircrafts), the impact of vertical directivity would instead be non-negligible and, thus, should be implemented. The flexibility of the framework and its open-source distribution easily allow this extension.

According to Harmonoise, the horizontal directivity always affects the signal emitted by the LS, while it is relevant for the HS only for heavy vehicles [31]. For the LS, the directivity correction coefficient, expressed as a variation in the SPL in one-third octave frequency bands, is defined as

$$\Delta L_H(\varphi,\psi) = \begin{cases} (-2.5 + 4|\cos\varphi|)\sqrt{\cos\psi} & f \in B \\ 0 \qquad \text{otherwise}\,, \end{cases} \tag{15}$$

where $B = [800, 6300]$ Hz. We implement this effect by using a time-varying bandpass filter added to the propagation module. In particular, for each simulation instant:

1. The angles φ, ψ are computed based on the (instantaneous) relative position of the source and receiver;
2. A bandpass FIR filter $H_{\text{dir}}(\varphi, \psi)$ is designed, with gain in the pass band computed using (15). Note that, to reduce the computational burden of computing a filter for each simulation instant, directivity filters are pre-computed and stored in a table when initializing the framework;
3. The source signal is filtered using $H_{\text{dir}}(\varphi, \psi)$.

For the highest source in HVs, the directivity correction is expressed as [31]

$$\Delta L_H(\varphi,\psi) = (1.546(\pi/2-\varphi)^3 - 1.425(\pi/2-\varphi)^2 + 0.22(\pi/2-\varphi) + 0.6)\sqrt{\cos\psi}\,, \tag{16}$$

that can be implemented as a gain element in the propagation module (since it does not depend on frequency but only on the angles φ, ψ).

In addition to the source directivities, first-order common microphone directivity patterns have been implemented. Assuming the receiver directivity to be invariant to rotations with respect to the horizontal plane, the first-order directivity pattern is

described by the formula

$$D(\alpha) = a + (1 - a)\cos\alpha\,, \tag{17}$$

where α represents the angle between the orientation of the microphone and the line connecting it to the position of the source, and a is a parameter that can be used to define different patterns (e.g., omnidirectional, $a = 1$, subcardioid, $a = 0.75$, cardioid, $a = 0.5$, supercardioid $a = 1/3$, hypercardioid, $a = 0.25$ and figure-8, $a = 0$).

3.3 Simulation setup

The overall structure of *TrafficSoundSim* is depicted in Fig. 6. The simulator consists of three independent blocks that communicate via shared parameters and features a unidirectional signal flow.

The first block constitutes the interface of *TrafficSoundSim*, and is responsible for the collection and storage of the parameters to set up the simulation. These parameters include: i) environmental parameters (temperature, pressure, relative humidity, flow resistivity of the ground surface material); ii) microphone parameters (position and directivity pattern of each microphone); iii) source parameters (type of vehicle –car or HV–, trajectory, speed, engine speed, engine load). The parameters are shared among the blocks.

The second block constitutes the source synthesis module, responsible for the generation of source signals as described in Sec. 3.1. The road/tire interaction noise generation model has been implemented in Python as part of *TrafficSoundSim*, whereas for the Baldan model, adopted to generate engine noise, we rely on an open-source Rust implementation [11]. The road/tire interaction and engine noise are then combined according to the Harmonoise model to produce the source signals of the LS and HS that are then handed over to the third block.

The third and last block constitutes the propagation module, responsible for the simulation of acoustic propagation, air absorption, ground effect and source directivity,

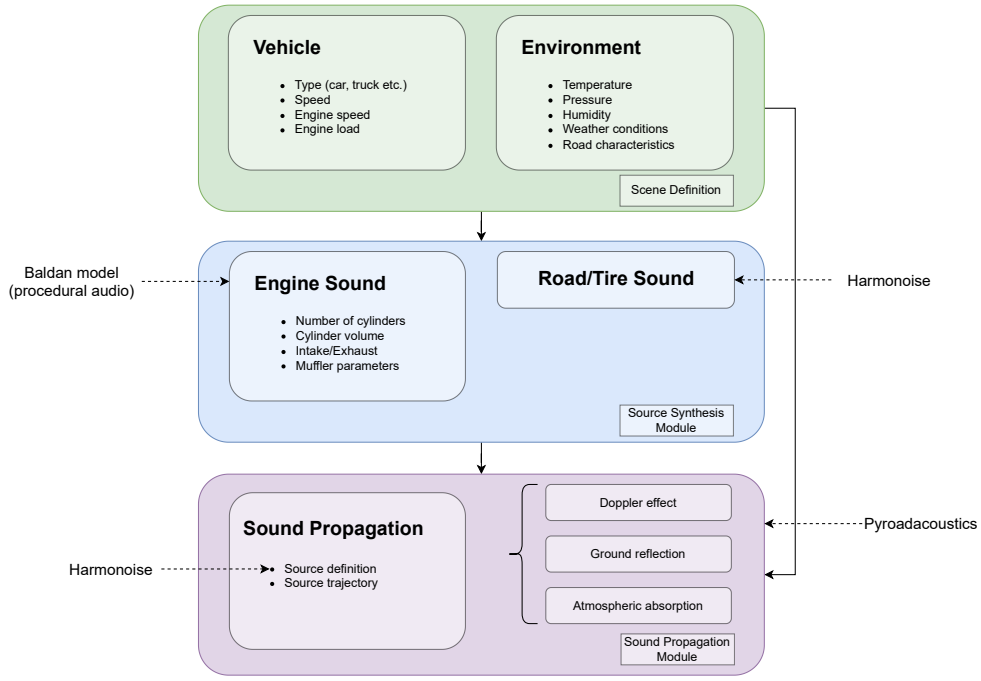

Fig. 6 *TrafficSoundSim* is composed of three blocks: the interface of the model, that collects the description of the acoustic scene to be simulated; the source synthesis module, responsible for the generation of source signals based on the provided input parameters; the sound propagation module, simulating acoustic propagation and producing the output signal.

as described in Sec. 3.2. This module takes as input the LS and HS signals and propagates them in the scene defined in the first block: variable-length delay lines are used to simulate Doppler effect and acoustic delay, whereas ground reflection, air absorption and source directivity are implemented using FIR filters. This block returns a single-channel signal for each microphone present in the acoustic scene.

4 Results and Discussion

To illustrate the capabilities of *TrafficSoundSim* we run simulations of pass-by events and compare synthetic and real signals. The simulations are designed starting from real-world labelled recordings, collected by Bosch on a two-lane German municipal road. The recordings have been obtained with an omnidirectional microphone deployed

on the sidewalk, at 2.7 m height and 3.5 m horizontal distance from the closest edge of the closest lane. The recording microphone has a flat magnitude response in the frequency range $f \in [100, 5000]$ Hz. For higher frequencies, as a consequence of the microphone enclosure, a lowpass filtering effect and some artifacts caused by resonance are observed. The recorded audio is accompanied by metadata for each passing vehicle, including its type (car or HV), speed, pass-by timestamp, and direction of transit (vehicles drive from left to right in the closest lane, and from right to left in the other one). Using this information, we set up simulations that emulate the target recordings: to this end, we assume that vehicles have a constant speed throughout the pass-by, as no turns, crossroads or traffic lights are present nearby the microphone position; moreover, we only consider recordings in which isolated events are registered. We define an event to be isolated if no other pass-by is registered in an interval of 60 s temporally centered on the pass-by instant (i.e., the moment in which the vehicle is closest to the microphone). A total of 5 isolated pass-by events have been identified for each class (car and HV) and used to run simulations for evaluation purposes.

To generate synthetic pass-bys, we use the metadata to set up simulations using the procedure described in Section 3.3. We adopt a sampling frequency $f_s = 16$ kHz and set the following parameters for the definition of the acoustic scene: a temperature $T = 20\,^{\circ}$C, in line with the average day temperature of Southern Germany in Autumn, when the data was collected; an asphalt flow resistance $\sigma = 20\,000 \cdot 10^3$ Ns/m^{-4} [42]. We heuristically set the atmospheric pressure $p = 1$ atm and the relative humidity to 50%. To assess the source synthesis module, we simulate each pass-by event using three different models. First we define a *baseline* (BL) model, that represents each vehicle using a single point source with a height corresponding to the Harmonoise LS (i.e., 0.01 m). To generate the BL source signal, we use linear predictive (LP) modeling [37] to match the BL spectrum with that of a 1 s-long segment of the recorded signal considered in each experiment, starting 1 s after the pass-by instant (in order to avoid

modeling the prominent Doppler shift around the pass-by instant). To this end, we design a third order linear predictor by analysing the segment of the recorded signal, and use it to filter a white noise signal to create the spectrally matched source signal. We then define the *Harmonoise* (HM) model, corresponding to the characterization of each vehicle using a HS and a LS as described in Sec. 3.1.3; in this model, the engine and tire/road components are generated by filtering white noise as described, respectively, in (2) and (6) [31]. Finally, the *Harmonoise + Baldan* (HM+BD) model denotes the complete source signal generation model described in Sec. 3.1. Here, the road/tire interaction noise is generated via the Harmonoise synthesis process (see eq. (2)), while the propulsion noise is generated using the Baldan model.

Since no information on the engine characteristics is available for the recorded pass-bys, we choose to use 4 cylinders for cars, 6 for HVs, and we manually tune the engine speed and load parameter. Moreover, we set the amplitude parameters of the three engine components (intake, exhaust and vibration) introduced in (3) as follows: since the vibration components mostly contributes to the low-frequency rumbling noise perceived in the cabin of the vehicle, and we consider for the current simulations a recording position located outside of it where its impact is limited, we set $A_2 = 0.1$ in all simulations. As done for the engine speed and load, we manually tune A_1 and A_3. After defining the trajectories (i.e., start and end point, direction of movement) using the labels, we simulate 50 s-long audio segments. To match the microphone magnitude response, we apply a second-order bandpass filter to all simulated signals, with bandpass range between 100 Hz and 5000 Hz.

To assess the effectiveness of *TrafficSoundSim*, we first analyze the power spectral density (PSD) computed using the Welch's method [47]. Accordingly, given an audio signal $x[n]$, we first compute the periodogram as [45]

$$P_x(m, \omega_k) = \frac{1}{M_w} |\mathrm{FFT}_{N_{\mathrm{fft}},k}(x[n])|^2 = \frac{1}{M_w} \left| \sum_{n=0}^{N_{\mathrm{fft}}-1} x[n] w[n + mR] e^{-j\frac{2\pi nk}{N_{\mathrm{fft}}}} \right|^2 , \quad (18)$$

where ω_k is the kth frequency bin, the mth frame of signal $x[n]$ is extracted using the $M_w = 2f_s$ samples-long Hann window w, $R = f_s/2$ denotes the hop size and $\mathrm{FFT}_{N_{\mathrm{fft}}}$ is the N_{fft}-point fast Fourier transform (FFT), where we use $N_{\mathrm{fft}} = M_w$. If we average P_x over a defined number K of frames of $x[n]$ we obtain the PSD

$$\mathrm{PSD}(x, \omega_k) = \frac{1}{K} \sum_{m=0}^{K-1} P_x(m, \omega_k)\,. \tag{19}$$

We perform the evaluation as follows. First, we compute the PSD of a 4 s-long audio segment extracted from the real and synthetic audio signals, temporally centered on the pass-by instant. This measure is used to evaluate the quality of the signal in a time interval close to the pass-by instant, thus excluding the signal tails (note that here we consider as tails the parts of the signal that are more than 2 s distant from the pass-by instant). Then, we trace a temporal stack of periodograms for the whole 50 s-long signal including tails, as discussed in the following. To quantitatively assess how close the simulated signals are to the recorded ones, we adopt two error metrics, namely the root mean squared error (RMSE) between the PSDs of two signals, defined in dB-scale as

$$\mathrm{RMSE}(x_{\mathrm{sim}}, x_{\mathrm{real}}) = \sqrt{\frac{1}{P} \sum_{k=1}^{P} \Big[20 \log_{10}(\mathrm{PSD}(x_{\mathrm{real}}, \omega_k)) - 20 \log_{10}(\mathrm{PSD}(x_{\mathrm{sim}}, \omega_k))\Big]^2}, \tag{20}$$

where P denotes the length of the PSD vector (i.e., the total number of frequencies on which the PSD is computed), and the mean absolute error (MAE) between the power spectral densities in dB

$$\mathrm{MAE}(x_{\mathrm{sim}}, x_{\mathrm{real}}) = \frac{1}{P} \sum_{k=1}^{P} |20 \log_{10}(\mathrm{PSD}(x_{\mathrm{real}}, \omega_k)) - 20 \log_{10}(\mathrm{PSD}(x_{\mathrm{sim}}, \omega_k))|\,, \tag{21}$$

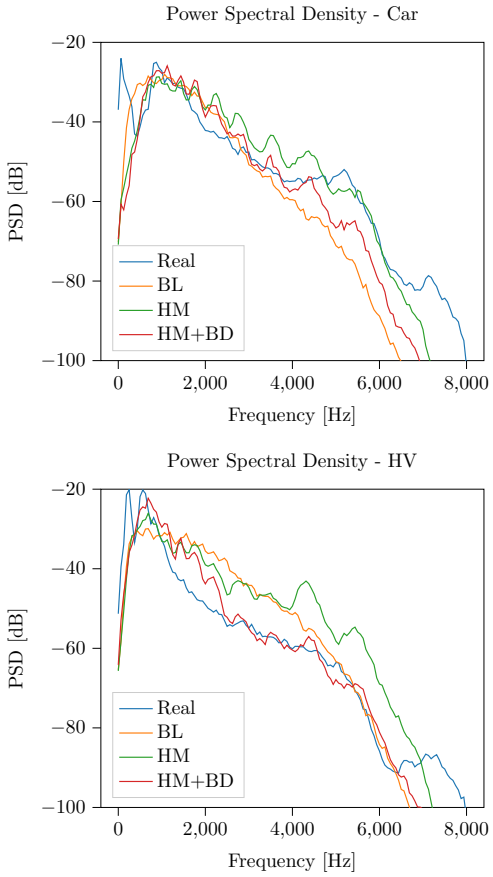

Fig. 7 PSD computed on a 4 s signal temporally centered on the pass-by instant for the recorded event and the simulated events for a car (speed $v = 52$ km/h) and HV ($v = 56$ km/h) using baseline, Harmonoise and combined Harmonoise + Baldan model for signal generation. The engine speed is set to 3000 RPM, and the propagation is the same for all simulations.

Since most of the information is contained in the time interval around the pass-by instant, both metrics are computed on the 4 s-long signal in the following experiments.

To evaluate the perceptual quality of the simulated sounds, we also consider two psychoacoustics sound quality measures from the Zwicker's method [49], namely the *roughness* and the *sharpness*. These sound quality measures are commonly used to evaluate drone noise annoyance [1] and traffic noise annoyance [48]. Roughness, measured in Asper, measures the subjective perception of rapid amplitude and frequency

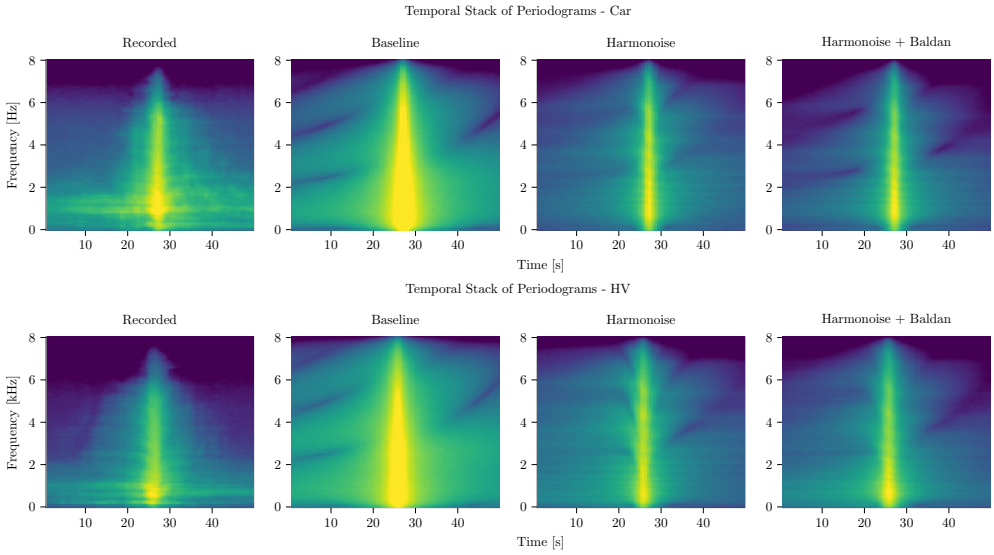

Fig. 8 Temporal stacks of periodograms computed for passing car (speed $v = 52\,\mathrm{km/h}$, engine speed 3000 RPM) and HV (speed $v = 56\,\mathrm{km/h}$, engine speed 3000 RPM): comparison between recorded and simulated events using BL, HM and HM+BD source signal generation models.

variation of sound; a high roughness is an indicator of an aggressive sounding acoustic signal [18]. Sharpness denotes the strength of high-frequency components in the sound and is measured in Acum; it can be interpreted as a measure of the colour of the sound [13]. Both metrics are computed based on auditory models and correlate with human perception. Note that, since they are not error metrics, they are computed on individual audio signals, and the assessment is performed by analysing the agreement between the sound quality measures obtained for the real and for the simulated signals. We used the implementation of metrics provided in [25].

4.1 Experimental Results

We first qualitatively analyze the difference between the PSDs and the periodograms of a single pass-by event for each of the two classes. For the car vehicle, a speed of 52 km/h is used in accordance with the label of the recorded event, and for the Baldan engine model a 4-cylinder engine is simulated, running at a speed of 3000 RPM. Similarly, for

the HV a speed of 56 km/h and a 6-cylinder engine running at 3000 RPM have been used. In Fig. 7 we show the PSD computed on a 4 s-long segment of the audio signals, temporally centered on the pass-by instant, comparing the real event and the three simulations (BL, HM, HM+BD). All models produce a PSD that closely matches the one of the recorded event in the frequency range $f \in [100, 5000]$ Hz, with HM and HM+BD achieving the closest match and showing the capability of the models to accurately simulate a passing vehicle at constant speed. Note that, since the BL source signal is designed to match the spectral features of the recorded signal used for the experiment, a close match of the two PSDs is to be expected. At higher frequencies, some resonance artifacts caused by the microphone enclosure can be observed in the real event. However, their impact to the overall PSD is negligible, being their amplitude more than 30 dB lower than the amplitude registered at 5000 Hz. We also observe that, whereas for car events the HM and HM+BD show a comparable performance, the difference becomes larger when comparing HV events. In HV pass-bys, in fact, the engine component plays a more important role: therefore, the HM+BD simulation, including a more detailed engine texture, is better suited to simulate heavy vehicles.

In Fig. 8 we show temporal stacks of the periodograms of the full 50 s-long signals corresponding to the same events. The periodograms have been computed using a sliding window approach, on 2 s-long segments with an overlap of 500 ms. We observe that the recorded signals are spectrally richer than all of the simulations: this is due both to the presence of low-amplitude, broadband background noise, that has not been included in the simulated signals, and to the scattering and diffusion effects, that are not modelled by the geometrical acoustic propagation model adopted in *Traffic-SoundSim*. The difference is particularly evident in the signal tails, whereas around the pass-by instant a better match can be observed. When compared to the BL, the HM and HM+BD models exhibit a closer match to the source signals, with tonal components (i.e., horizontal lines) corresponding to engine orders and resonances not

appearing in the BL model. An artifact that can be observed in the simulations, particularly accentuated for the BL model, is the presence of deep valleys produced by the destructive interference caused by the ground reflection. This is attenuated in the recorded audio by the scattering effect, thus it is not visible in the real-world data. This illustrates the main limitation of the proposed propagation framework.

The results suggest that the proposed simulation framework, with either the HM or the HM+BD sound source signal, achieves a higher realism compared to the baseline, showing the importance of having an accurate sound source model for the simulation of traffic noise. Moreover, audio demos for these comparisons are available in the additional file set 1 (for the car events) and additional file set 2 (for the HV events). From these files, it is possible to evaluate the potential of the proposed framework and the impact of the choice of the sound source model. In the HM+BD model, the rumbling and rattling sound of the engine sound contribute to the realism of the simulation, at the cost of the manual tuning of the engine parameters.

To show how these results generalize to different pass-by events (i.e., recorded with different cars, having a different speed and different engine parameters), we then compare five 4 s-long pass-bys for each of the two classes and report the mean and standard deviations of the MAE and RMSE scores for the BL, HM and HM+BD models in Tab. 2. The speed of the recorded events is in the range $[52, 69]$ km/h, and, since no information is available on the engine parameters, we randomly selected an engine speed in the range $[2000, 5000]$ RPM. For the BL, a source signal spectrally matching the recorded one is generated for each considered event using LP filtering. Both metrics improve for all vehicle types when a more accurate source signal generation is used, showing the importance of adopting an accurate source model in the simulation of vehicle pass-by events. As an additional remark, in the generation of the propulsion signal, the engine parameters have been manually tuned, and do not necessarily match the real conditions. Although a better tuning could lead to improved

realism, these results suggest that even without an accurate knowledge (or tuning) of the engine parameters, difficult to fulfill in a real-world situation where this information is usually missing, a good match with recorded can be obtained. Finally, the benefit of using the physical engine model instead of the Harmonoise engine characterization is the improved realism of the source signal texture, that includes rattling and rumbling effects, not captured by a broadband noise signal characterization.

To further assess the auditory plausibility of the proposed framework, the sharpness and roughness psychoacoustic metrics are computed for the 5 pass-by events in a 2 s time window centered on the pass-by instant, and reported, for the recorded audio and the three simulations, in Tab. 3. This table shows that the BL achieves the closest match with the recorded audio, that might be explained by the fact that the source signal used for the BL is artificially shaped to closely match the recorded audio. In accordance to what observed when using error metrics, the table shows that HM+BD model comes closer to the recorded audio than the HM, suggesting that a more accurate engine model is beneficial to obtain a higher realism.

The perceptual evaluation of the framework goes beyond the scope of this work; nevertheless, audio demos are available as additional material to show its capabilities. Even though simulations achieve a good match with recordings in terms of the shown metrics and sufficiently good auditory quality as from listening to the sounds, differences can be both heard in the audio files and seen in the PSD plots. The first major reason for that is the absence of any form of background noise in the simulation, that constitutes a difference with real recordings. Secondly, the Harmonoise model adopted for the generation of rolling noise is not capable of simulating the so-called *horn effect*, that impacts on the perceptual quality of the road/tire interaction sound. In order to account for that, more complex physical models of the interaction between the tire and the asphalt surface might be used, at the cost of significantly increasing the computational effort required by the simulator [5]. Lastly, in the propagation module diffusion

		BL	HM	HM + BD
Car	RMSE	8.21 ± 1.70	4.89 ± 1.20	$\mathbf{4.05 \pm 1.10}$
	MAE	6.66 ± 0.89	4.18 ± 1.06	$\mathbf{3.38 \pm 1.09}$
HV	RMSE	8.30 ± 1.12	6.77 ± 1.92	$\mathbf{4.90 \pm 1.03}$
	MAE	7.10 ± 0.78	6.05 ± 1.91	$\mathbf{4.18 \pm 0.96}$

Table 2 Mean and standard deviation of RMSE and MAE metrics computed for the BL, HM and HM+BD models on 5 pass-by events for each vehicle type (car, HV). All metrics are expressed in dB-scale.

and scattering effects are not considered. These factors have an impact on the sound quality of the simulations, specially in the higher frequency range, and explain the auditory and spectral differences observed between the recorded and simulated audio.

In all the discussed simulations, a complex-valued reflection coefficient has been used to model ground reflection, as introduced in Sec. 3.2.1. To illustrate the benefit of using a complex-valued reflection coefficient, as compared to using a real-valued one, we report in Fig. 9 the PSD of two 4 s-long simulations generated using the HM+BD source signal model and the two different reflection models. Since the effect produced by the reflection is independent on the source signal model, we limit the analysis to the HM+DB model, without loss of generality. The metrics corresponding to these tests are reported in Tab. 4, the sound quality measures are reported in Tab. 5. From both the figure and the tables it can be observed that modelling the ground reflection using a complex-valued reflection coefficient leads to a better match with the recorded pass-by, with a MAE improvement of 1.78 dB and a RMSE improvement of 1.35 dB. Moreover, the sound quality measures and the auditory evaluation of the audio demos, available in the additional file set 3, confirm this result. When a real-valued reflection coefficient is used, in fact, the effect of the ground reflection on the phase of the signal is neglected, whereas it is considered when the complex-valued reflection coefficient is used.

We finally analyse the simulation time needed to create auralizations using the proposed method. In Tab. 6 we report the total time required by the proposed method

		Real	BL	HM	HM + BD
Car	Roughness	0.187 ± 0.053	$\mathbf{0.135 \pm 0.073}$	0.083 ± 0.049	0.253 ± 0.012
	Sharpness	1.493 ± 1.133	$\mathbf{1.464 \pm 0.049}$	1.646 ± 0.013	1.608 ± 0.061
HV	Roughness	0.165 ± 0.042	$\mathbf{1.530 \pm 0.028}$	0.127 ± 0.024	0.135 ± 0.030
	Sharpness	1.459 ± 0.054	$\mathbf{1.433 \pm 0.040}$	1.601 ± 0.013	1.562 ± 0.008

Table 3 Mean and standard deviation of Roughness and Sharpness metrics computed for the Real, BL, HM and HM+BD models on 5 pass-by events for each vehicle type (car, HV).

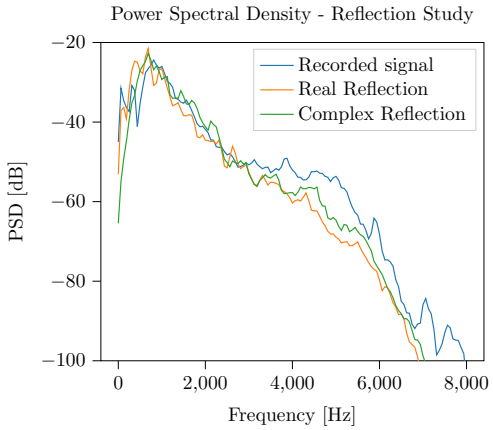

Fig. 9 Comparison between PSD obtained using, respectively, real-valued and complex-valued reflection coefficients to model ground effect in the propagation module. Single HV pass-by event, with speed $v = 61\,\text{km/h}$ and engine speed 3000 RPM.

Metric	Real $R(\vartheta)$	Complex $R(\vartheta)$
RMSE	6.65	**4.87**
MAE	5.17	**3.82**

Table 4 Comparison between recorded pass-by and simulations run with real-valued and complex-valued ground reflection coefficient $R(\vartheta)$ for the same single HV pass-by event (speed $v = 61\,\text{km/h}$, engine speed 3000 RPM) represented in Fig. 9. All metrics are expressed in dB-scale.

to simulate 1 s, 10 s, 30 s and 60 s of audio and break it down to the time needed to perform different parts of the simulation. The simulations run on a 8 core Apple M1 CPU, with 16 GB memory. In the upper part of the table, we investigate the time needed to generate the source signal, to initialize the simulation and to generate the

Metric	Recorded audio	Real $R(\theta)$	Complex $R(\theta)$
Roughness	0.173	0.381	**0.165**
Sharpness	1.524	1.389	**1.461**

Table 5 Comparison of Roughness and Sharpness sound quality metrics computed for the real recorded signal and for simulations run with real-valued and complex-valued ground reflection coefficient $R(\theta)$. The considered event is a HV with speed $v = 61\,\mathrm{km/h}$, engine speed 3000 RPM.

Component	1 s audio	10 s audio	30 s audio	60 s audio
Source signal generation	1.092	1.290	1.733	2.654
Initialization	0.133	3.884	3.902	4.081
Direct path	0.567	5.597	17.010	33.998
Reflected path	2.292	22.044	64.795	127.276
Other (auxiliary)	4.311	5.719	17.370	34.765
Total time	8.395	38.534	104.810	202.774
Air absorption filtering	0.548	5.392	16.437	32.871
Asphalt reflection filtering	1.463	13.872	39.887	77.434
Directivity filtering	0.359	3.547	10.776	21.584

Table 6 Time in seconds required to simulate 1 s, 10 s, 30 s and 60 s of audio, split into different simulation blocks. Upper part of the table: breakdown of total simulation time into source signal generation (HM+BD model), initialization, simulation of acoustic propagation along direct and reflected path and time spent into other auxiliary functions. The sum of these components adds up to the total time. Lower part of the table: time employed by absorption filtering, asphalt reflection filtering and directivity filtering throughout the entire simulation (i.e., direct and reflected paths).

direct and the reflected audio signals. Moreover, we report the additional time required by auxiliary functions, and not included in the listed components. In the lower part of the table, instead, we report the time needed to perform the air absorption, asphalt reflection and directivity filtering, when simulating both the direct and reflected path. Simulating only the direct path results in the fastest method, achieving real-time capability in the current implementation. Increasing the physical accuracy, by adding the reflected path, comes at the cost of a slower simulation. The current Python implementation of *TrafficSoundSim* might be optimized for simulation speed; this improvement is beyond the scope of this paper and is left for future work.

4.2 Comparison with state of the art

In this section, we compare *TrafficSoundSim* with the state-of-the-art auralization framework [40], based on spectral modeling synthesis (SMS). This framework is specifically designed for the simulation of accelerating passenger cars and comes with an extensive set of adjustable parameters to maximize the realism of the auralizations in specific scenarios. Its internal structure consists of an emission module, gathering two submodules designed to generate road/tire noise and propulsion noise, a propagation filtering module and a final rendering module. Although the backbone of the emission module is based on the Harmonoise two-source model of a car vehicle, the source signals are generated using spectral modeling synthesis, and the parameters are obtained from an extensive set of recordings of several car models. The propagation module produces the effects of the propagation delay and spreading of the emitted sound, the Doppler effect, the convective amplification, the ground reflection and air absorption; the implementation of these effects relies on delay lines and digital filtering. The key features of [40] are the possibility to realistically simulate accelerating vehicles and gear changes, as showcased in the published demos[1,2]. Its main drawbacks are that it requires recordings for the source signal generation, it is tailored to simulate passenger cars, hence not supporting different vehicles, and it is not publicly available.

Since [40] does not allow to generate truck simulations, we compare it with *TrafficSoundSim* based on the car pass-by event used for the experiments discussed in Fig. 7 and Fig. 8. To enable this comparison, the authors of [40] generated an auralization with their tool corresponding to the target scenario (described in the beginning of Sec. 4.1, and manually tuned the SMS simulator parameters based on the recorded audio considered in the comparison: the car drives in the $3^{\mathbf{rd}}$ gear, with an engine speed of 2700 RPMs, and the recording microphone is set at a height of 1.2 m. The SMS audio segment used in this evaluation is available in the additional files.

[1] https://www.youtube.com/watch?v=_XbKWmKjdr4
[2] https://www.youtube.com/watch?v=fQPnLlXvG9I

Following the experimental outline presented in Sec. 4.1, we first qualitatively compare in Fig. 10 the PSDs of 4 s-long segments of the recorded and simulated signals, centered around the pass-by instant. The SMS framework closely matches the recorded signal at frequencies lower than 2 kHz, whereas at higher frequencies the HM and HM+BD model achieve a closer match. In Fig. 11 we show temporal stacks of the periodograms of 40 s-long signals centered around the pass-by instant: this figure shows that the SMS framework captures the signal tails better than *TrafficSoundSim*, achieving a closer match with the recorded audio. Finally, we carry out a quantitative evaluation by computing error and psychoacoustics metrics on the 4 s-long segments centered on the pass-by instant, reported in Tab. 7. *TrafficSoundSim* achieves the best RMSE score with the HM source model and the best MAE score with the HM+BD source model, outperforming the SMS framework. In terms of psychoacoustics metrics, the best roughness score is obtained by the HM+BD model, whereas the best sharpness is obtained by the BL model.

The analysis shows that *TrafficSoundSim* achieves competitive results when compared with the state-of-the-art. Although the audio demo and Fig. 11 show that [40] is more effective in modeling longer-term effects of the passing vehicle, the metrics reported in Fig. 7 and the short-term PSD plotted in Fig. 10 suggest that *TrafficSoundSim* achieves higher realism around the pass-by instant. Moreover, as underlined in Tab. 1, we remark that the two tools have different features and limitations that might make each of them the best option for some specific scenarios. Finally, the open-source distribution of *TrafficSoundSim* constitutes a major advantage over other tools not publicly available for usage.

5 Conclusions

In this work we have presented *TrafficSoundSim*, a framework for the acoustic simulation of passing vehicles that relies on the *pyroadacoustics* road acoustics propagation

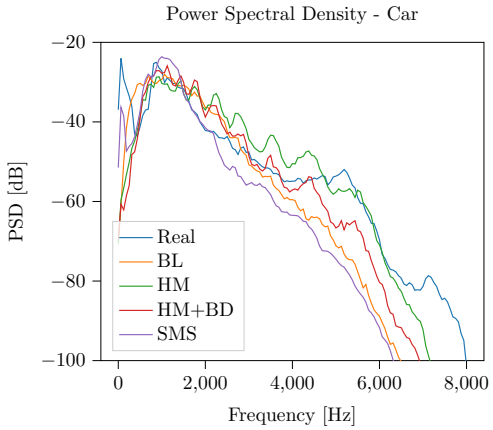

Fig. 10 Comparison between PSD of 4 s-long recorded car pass-by event and simulations using baseline, *TrafficSoundSim* (Harmonoise and Harmonoise + Baldan configurations) and state-of-the-art simulator [40] based on modeling synthesis (SMS). The parameters of the SMS method are manually tuned based on the recorded signal: the car drives in 3rd gear with an engine speed of 2700 RPM. The parameters of *TrafficSoundSim* are the same used in Fig. 7.

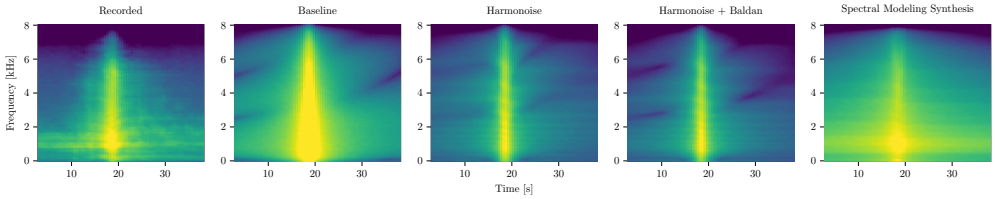

Fig. 11 Comparison between temporal stacks of periodograms computed for a recorded car pass-by and simulations using baseline, *TrafficSoundSim* (HM and HM+BD) and SMS [40]. All parameters are the same used for Fig. 10.

simulator. *TrafficSoundSim* contains two main modules, used for the generation of source signals and for the simulation of acoustic propagation. The source signal generation is based on the Harmonoise engineering method, that characterizes a passing vehicle using two vertically stacked sources containing a mixture of road/tire interaction noise and engine noise. The road/tire interaction is simulated using broadband noise, filtered in one-third octave bands according to the Harmonoise description. The engine component, instead, is simulated using the Baldan model, based on physical modelling, to improve the sound quality over Harmonoise by including tonal components as well as rattling and rumbling sounds that characterize thermal engines.

Metric	Real	BL	HM	HM+BD	SMS
RMSE	-	8.65	**5.24**	5.28	10.99
MAE	-	6.48	4.60	**4.02**	8.47
Roughness	0.118	0.181	0.038	**0.160**	0.193
Sharpness	1.471	**1.401**	1.644	1.544	1.364

Table 7 Comparison of error and psychoacoustics metrics for a recorded car pass-by event and simulations using basline, HM, HM+BD and SMS [40] simulators. All parameters are the same used for Fig. 10.

The propagation module is an extended version of *pyroadacoustics*, that models sound propagation and Doppler effect using variable length delay lines, and ground reflection and air absorption using FIR filters. In particular, we introduced an improved ground reflection model based on the use of a complex-valued reflection coefficient, and implemented directivity for both sound sources, based on the Harmonoise directivity model, and for the virtual microphones, using standard directivity patterns. Lastly, we have shown that an accurate source signal generation can significantly improve the realism of the simulations by comparing simulated and recorded pass-by events for cars and heavy vehicles.

Some improvements of *TrafficSoundSim* can be foreseen for the future. First, a sound reproduction module based on head-related transfer functions can be implemented to build a full auralization framework. Second, although in the current version the only simulated reflection is the one produced by the ground surface, further reflections from buildings and objects on the road can be included by means of additional delay lines. Third, further study on source signal generation is required to characterize electric vehicles: the decoupled structure adopted for the generation of the rolling and propulsion noise in *TrafficSoundSim* facilitates its extension to include these vehicles by introducing a different model of the sound produced by their engine. Lastly, perceptual tests could be performed to further validate the framework.

Declarations

Funding

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 956962 and from the European Research Council under the European Union's Horizon 2020 research and innovation program / ERC Consolidator Grant: SONORA (no. 773268). This paper reflects only the authors' views and the Union is not liable for any use that may be made of the contained information.

Conflict of interest/Competing interests

Author LB was employed by Robert Bosch LLC; author AG was employed by Robert Bosch GmbH. The other authors declare that they have no competing interests.

Authors' contributions

SD, LB AG and TVW designed the simulation framework. LB provided the data used in for the experimental assessment. SD implemented the framework, run the experiments, and drafted the manuscript. All authors read and reviewed the final manuscript.

Acknowledgements

The authors would like to thank Hans-Georg Horst for contributing to the collection of data used in the development of this work, and Reto Pieren for his collaboration in providing the audio demos of his auralization tool.

Availability of data and materials

The source code is available at https://github.com/boschresearch/acoustic-traffic-simulation-counting.

The following software, that implements the Baldan model, has been used in the work. Project name: enginesound v1.6; project home page: https://github.com/DasEtwas/enginesound; operating system: platform independent; programming language: Rust; license: MIT

Audio demos are available as additional files.

List of Abbreviations

SPL	Sound pressure level
FIR	Finite impulse response
HV	Heavy vehicle
RPM	Revolutions per minute
LS	Lower source
HS	Higher source
BL	Baseline model
HM	Harmonoise model
HM+BD	Harmonoise + Baldan model
PSD	Power spectral density
FFT	Fast Fourier transform
RMSE	Root mean squared error
MAE	Mean absolute error
SMS	Spectral modeling synthesis

References

[1] Alkmim, M., J. Cardenuto, E. Tengan, T. Dietzen, T. Van Waterschoot, J. Cuenca, L. De Ryck, and W. Desmet. 2022, 11. Drone noise directivity and psychoacoustic evaluation using a hemispherical microphone array. *The Journal of the Acoustical Society of America 152*(5): 2735–2745. https://doi.org/10.1121/10.0014957 .

[2] Attenborough, K., S.I. Hayek, and J.M. Lawther. 1980, November. Propagation of sound above a porous half-space. *J. Acoust. Soc. Am. 68*(5): 1493–1501. https://doi.org/10.1121/1.385074 .

[3] Baldan, S., S. Delle Monache, and D. Rocchesso. 2017. The Sound Design Toolkit. *SoftwareX* 6: 255–260. https://doi.org/10.1016/j.softx.2017.06.003 .

[4] Baldan, S., H. Lachambre, S. Delle Monache, and P. Boussard 2015. Physically informed car engine sound synthesis for virtual and augmented environments. In *Proc. 2015 IEEE 2nd VR Workshop Sonic Interactions Virtual Environments (SIVE)*, Arles, France, pp. 21–26.

[5] Beckenbauer, T., P. Klein, J. Hamet, and W. Kropp 2008. Tyre/road noise prediction: A comparison between the speron and hyrone models - part 1. In *Proc. 7th European Conf. Noise Control 2008 (EURONOISE 2008)*, Paris, France, pp. 5017–5022.

[6] Braun, M., S. Walsh, J. Horner, and R. Chuter. 2013, November. Noise source characteristics in the ISO 362 vehicle pass-by noise test: Literature review. *Applied Acoustics 74*(11): 1241–1265. https://doi.org/10.1016/j.apacoust.2013.04.005 .

[7] Damiano, S. 2022. pyroadacoustics - Python Road Acoustics Simulation. https://github.com/steDamiano/pyroadacoustics.

[8] Damiano, S., L. Bondi, and S. Ghaffarzadegan. 2024. Trafficsoundsim - traffic sound simulation framework. https://github.com/boschresearch/acoustic-traffic-simulation-counting.

[9] Damiano, S., L. Bondi, S. Ghaffarzadegan, A. Guntoro, and T. van Waterschoot 2024. Can synthetic data boost the training of deep acoustic vehicle counting networks? In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech*

and Signal Processing (ICASSP), pp. 631–635.

[10] Damiano, S. and T. van Waterschoot 2022, September. Pyroadacoustics: a Road Acoustics Simulator Based on Variable Length Delay Lines. In *Proc. 25th Int. Conf. Digital Audio Effects (DAFx20in22)*, Vienna, Austria, pp. 216–223.

[11] DasEtwas. 2021. enginesound. https://github.com/DasEtwas/enginesound.

[12] Delany, M. and E. Bazley. 1970. Acoustical properties of fibrous absorbent materials. *Applied Acoustics 3*(2): 105–116. https://doi.org/https://doi.org/10.1016/0003-682X(70)90031-9 .

[13] DIN 45692 2009. Measurement technique for the simulation of the auditory sensation of sharpness. Technical report, Deutsches Institut für Normung, Berlin, Germany.

[14] Djukanović, S., Y. Patel, J. Matas, and T. Virtanen 2021. Neural network-based acoustic vehicle counting. In *Proc. 29th European Signal Process. Conf. (EUSIPCO)*, Dublin, Ireland, pp. 561–565.

[15] Dreier, C., J. Hahn, J. Heck, J. Llorca-Bofí, and M. Vorländer 2022. Real-time vehicle pass-by noise synthesis for urban sound auralization. In *Proc. 24th Int. Congr. Acoust.*, Gyeongju, South Korea.

[16] Dreier, C., J. Hahn, and M. Vorländer 2022. Inverse modelling of vehicle pass-by noise for auralizations. In *Proc. DAGA 2022: 48th Jahrestagung für Akustik*, Stuttgart, Germany, pp. 1455–1458.

[17] Dreier, C. and M. Vorländer 2022, August. Vehicle pass-by noise auralization in a virtual urban environment. In *Proc. Internoise 2022*, Glasgow, UK.

[18] ECMA-418-2 2020. Psychoacoustic metrics for itt equipment — part 2 (models based on human perception). Technical report, Ecma International, Geneva, Switzerland.

[19] European Environment Agency ed. 2009. *Transport at a crossroads: TERM 2008: indicators tracking transport and environment in the European Union.* Number no. 3/2009 in EEA report. Luxembourg: Office for Official Publications of the European Communities.

[20] Forssén, J., A. Hoffmann, and W. Kropp. 2018, March. Auralization model for the perceptual evaluation of tyre–road noise. *Applied Acoustics* 132: 232–240. https://doi.org/10.1016/j.apacoust.2017.11.023 .

[21] Forssén, J., T. Kaczmarek, J. Alvarsson, P. Lundén, and M.E. Nilsson 2009. Auralization of traffic noise within the LISTEN project—Preliminary results for passenger car pass-by. In *Proc. Euronoise 2009*, Edinburgh, UK.

[22] Fu, Y. 2021, January. *Auralisation of Traffic Flow using Procedural Audio Methods.* PhD Thesis, University of York, York, UK.

[23] Fu, Y. and D. Murphy 2018. A comparative overeview of traffic flow modelling approaches for auralisation. In *Proc. Euronoise 2018*, Crete, Greece, pp. 273–278.

[24] Fu, Y., D. Murphy, and A. Southern 2019. Traffic Flow Auralisation based on Single Vehicle Pass-by noise Synthesis. In *Proc. 23rd Int. Congr. Acoust.*, Aachen, Germany, pp. 1705–1712.

[25] Green Forge Coop. 2024. Mosqito (version 1.2.1). https://doi.org/10.5281/zenodo.11026796.

[26] Heutschi, K., R. Pieren, M. Müller, M. Manyoky, U.W. Hayek, and K. Eggenschwiler. 2014, January. Auralization of Wind Turbine Noise: Propagation Filtering and Vegetation Noise Synthesis. *Acta Acustica united with Acustica 100*(1): 13–24. https://doi.org/10.3813/AAA.918682 .

[27] Hoffmann, A. 2016. *Auralization, perception and detection of tyre-road noise.* Doktorsavhandlingar vid Chalmers tekniska högskola, Chalmers University of Technology, Gothenburg, Sweden.

[28] Hoffmann, A. and W. Kropp. 2019, February. Auralization of simulated tyre noise: Psychoacoustic validation of a combined model. *Applied Acoustics* 145: 220–227. https://doi.org/10.1016/j.apacoust.2018.08.026 .

[29] ISO 9613-1 1993. Acoustics, Attenuation of sound during propagation outdoors — Part 1: Calculation of the absorption of sound by the atmosphere. Geneva, Switzerland. International Organization for Standardization.

[30] Jonasson, H., U. Sandberg, G.v. Blokland, J. Ejsmont, G. Watts, and M. Luminari 2004. Source modelling of road vehicles. Technical report, Statens Provningsanstalt. https://www.diva-portal.org/smash/get/diva2:674007/FULLTEXT01.pdf.

[31] Jonasson, H.G. 2007. Acoustical source modelling of road vehicles. *Acta Acustica united with Acustica 93*(2): 173–184 .

[32] Jonasson, H.G. and S. Storeheier 2001. Nord 2000. New Nordic prediction method for road traffic noise. Technical Report SP Report 2001:10, SP Swedish National Testing and Research Institute. https://www.diva-portal.org/smash/get/diva2:1190134/FULLTEXT01.pdf.

[33] Kaczmarek, T. 2007. Road-vehicle simulation for psychoacoustic studies. In *Proc. Int. Congr. Acoust.*, Madrid, Spain.

[34] Kuijpers, A. and G. Van Blokland 2001. Tyre/road noise models in the last two decades: a critical evaluation. In *Proc. 2001 Inter-Noise and Noise-Con Congress and Conf.*, Volume 2001, The Hague, Netherlands, pp. 2494–2499.

[35] Kuttruff, H. 2007. *Acoustics: an introduction*. London: Taylor & Francis.

[36] Maillard, J. and J. Jagla 2012. Auralization of non-stationary traffic noise using sample based synthesis - comparison with pass-by recordings. In *Proc. 2012 Inter-Noise and Noise-Con Congress and Conf.*, Volume 2012, New York City, USA, pp. 5978–5989.

[37] Makhoul, J. 1975. Linear prediction: A tutorial review. *Proceedings of the IEEE 63*(4): 561–580. https://doi.org/10.1109/PROC.1975.9792 .

[38] Marchegiani, L. and X. Fafoutis. 2022. How Well Can Driverless Vehicles Hear? A Gentle Introduction to Auditory Perception for Autonomous and Smart Vehicles. *IEEE Intell. Transp. Syst. Mag.*: 92–105. https://doi.org/10.1109/MITS.2021.3049425 .

[39] Nilsson, M.E., J. Forssén, P. Lundén, A. Peplow, and B. Hellstrom 2011. LISTEN: Final Report. Technical report, Chalmers University of Technology. https://publications.lib.chalmers.se/records/fulltext/232736/local_232736.pdf.

[40] Pieren, R., T. Bütler, and K. Heutschi. 2015, December. Auralization of Accelerating Passenger Cars Using Spectral Modeling Synthesis. *Applied Sciences 6*(1): 5. https://doi.org/10.3390/app6010005 .

[41] Ranzo, A., P. Di Mascio, and A. D'Andrea 2020, September. Absorption Coefficients of Asphalt Pavements. In *Proc. 2nd Eurasphalt & Eurobitume Congr.*, Volume II, Barcelona, Spain, pp. 891–902.

[42] Rochat, J.L. and D.R. Read. 2013, 12. Effective flow resistivity of highway pavementsa). *The Journal of the Acoustical Society of America 134*(6): 4710–4716. https://doi.org/10.1121/1.4824964 .

[43] Sarrizin, M., C. Colangeli, K. Janssens, and H. van der Auweraer 2013. Synthesis techniques for wind and tire-road noise. In *Proc. 2013 Inter-Noise and Noise-Con Congress and Conf.*, Volume 247, Innsbruck, Austria, pp. 2303–2312.

[44] Smith, J.O. 2010. *Physical Audio Signal Processing.* online book, 2010 edition.

[45] Smith, J.O. 2011. *Spectral Audio Signal Processing.* online book, 2011 edition.

[46] Vorländer, M. 2020. *Auralization: Fundamentals of Acoustics, Modelling, Simulation, Algorithms and Acoustic Virtual Reality.* RWTHedition. Cham, Switzerland: Springer.

[47] Welch, P. 1967. The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Trans. Audio Electroacoust. 15*(2): 70–73. https://doi.org/10.1109/TAU.1967.1161901 .

[48] Zhang, X. 2022. *Development of pass-by noise prediction models for road vehicles employing machine learning.* Ph. D. thesis, KU Leuven.

[49] Zwicker, H.F.E. 2007. *Psychoacoustics - Facts and Models* (3 ed.). Springer Berlin, Heidelberg.