

# Development of an analysis method for the extraction of sound synthesis parameters from driving noise of motor vehicles

Toningenieur-Projekt

Martin Czuka

Assessor: DI Johannes Luig

Graz, SS/WS 2013



institut für elektronische musik und akustik



## **Abstract**

Building upon an existing IEM sound design tool for vehicles, an analysis method for extracting sound synthesis parameters from a recorded vehicle driving noise has been developed in this project. Using these parameters, the user is enabled to design an engine speed- and torque-dependent engine sound for the interior acoustic of a vehicle. This acoustic feedback provides the driver with additional information regarding his / her driving behaviour, in order to prevent higher power consumption and, respectively, an increased risk of an accident.

In the second part of the project, an easy-to-use graphical user interface has been developed for the sound design tool using the programming language C#.

## **Zusammenfassung**

Die vorliegende Projektarbeit baut auf einem am IEM erstellten Sounddesign-Tool für Kraftfahrzeuge auf, wobei im ersten Teil eine Analysemethode entwickelt wurde, die aufgenommene Fahrgeräusche eines Kraftfahrzeuges in entsprechende Klangsynthese-Parameter übersetzt. Mit Hilfe dieser Parameter ist es dem / der AnwenderIn möglich, ein drehzahl- und drehmomentabhängiges Motorgeräusch für die Innenraumakustik eines Fahrzeuges zu designen. Das zusätzliche akustische Feedback soll dem Lenker / der Lenkerin mehr Informationen über das eigene Fahrverhalten geben und damit vorbeugend gegen einen erhöhten Energieverbrauch beziehungsweise ein erhöhtes Unfallrisiko wirken.

Zur komfortablen Bedienung des Sounddesign-Tools wurde im zweiten Teil der Projektarbeit eine graphische Benutzeroberfläche in der Programmiersprache C# implementiert.

## Contents

|          |                                                       |           |
|----------|-------------------------------------------------------|-----------|
| <b>1</b> | <b>Introduction</b>                                   | <b>4</b>  |
| <b>2</b> | <b>The sound synthesis model</b>                      | <b>6</b>  |
| 2.1      | Engine orders . . . . .                               | 6         |
| 2.2      | Synthesis of engine orders . . . . .                  | 7         |
| 2.3      | Determination of the synthesis parameters . . . . .   | 9         |
| <b>3</b> | <b>The analysis method</b>                            | <b>13</b> |
| 3.1      | Engine speed estimation . . . . .                     | 14        |
| 3.1.1    | Basic methods of pitch detection . . . . .            | 14        |
| 3.1.2    | Advanced pitch detection and pitch tracking . . . . . | 18        |
| 3.1.3    | Post-processing . . . . .                             | 25        |
| 3.1.4    | Case studies . . . . .                                | 28        |
| 3.2      | Sound synthesis parameter extraction . . . . .        | 33        |
| 3.2.1    | Engine order extraction . . . . .                     | 33        |
| 3.2.2    | Calculation of C0 . . . . .                           | 43        |
| 3.2.3    | Calculation of C1 . . . . .                           | 45        |
| 3.2.4    | Calculation of C2 . . . . .                           | 45        |
| 3.2.5    | Case studies . . . . .                                | 46        |
| <b>4</b> | <b>Engine sound design software</b>                   | <b>50</b> |
| 4.1      | Overview . . . . .                                    | 50        |
| 4.2      | Matrix editor . . . . .                               | 52        |
| 4.3      | Engine order editor . . . . .                         | 53        |
| <b>5</b> | <b>Summary and outlook</b>                            | <b>55</b> |

# 1 Introduction

Sound provides living beings with an auditory sense with information about their surrounding. If a proper propagation medium is available, every motion is accompanied by an acoustic event, which contains information about the intensity and quality of this motion. The human brain is capable of processing this information and distinguishing between different qualities and intensities of sound.

In road traffic, the sound emitted by a vehicle is perceived by other traffic participants and contains information such as the vehicle speed as well as its current acceleration. The sound emitted in the interior of a vehicle on the other hand gives the driver of the vehicle information about his or her driving behaviour and sheds light on the current power consumption. Besides the tyre-road interaction as a source of noise and vibrations, the drive system of the vehicle is one of the main contributors to the overall interior acoustic of a vehicle. The rotational speed-dependent and torque-dependent sound emitted by the engine of the drive system shapes the specific interior acoustic of a vehicle and provides a well-recognizable acoustic feedback.

The Institute of Electronic Music and Acoustics (IEM) at the University of Music and performing Arts in Graz has developed a sound synthesis model for creating an engine sound for a vehicle. It enables a user to reproduce and alter existing engine sounds, or to create entirely new engine sounds for conventional and electric vehicles. For an user-friendly engine sound design process, a MATLAB-based graphical user interface with the synthesis model implemented in Pure Data has been created. However, the implementation of the analysis procedure for recreating existing engine sounds in this software has been implemented only rudimentarily. This is the point where the present Toningenieur-Projekt comes in.

Building upon an existing sound design tool for engine sounds of vehicles, an analysis method for extracting sound synthesis parameters from a recorded vehicle driving noise has been developed in this project. Using these parameters, the user is enabled to design an engine speed- and torque-dependent engine sound for the interior acoustic of a vehicle. This acoustic feedback provides the driver with additional information regarding his or her driving behaviour, in order to prevent higher power consumption and, respectively, an increased risk of an accident. In the second part of the project, an easy-to-use graphical user interface has been developed for the sound design tool using the programming language C#.

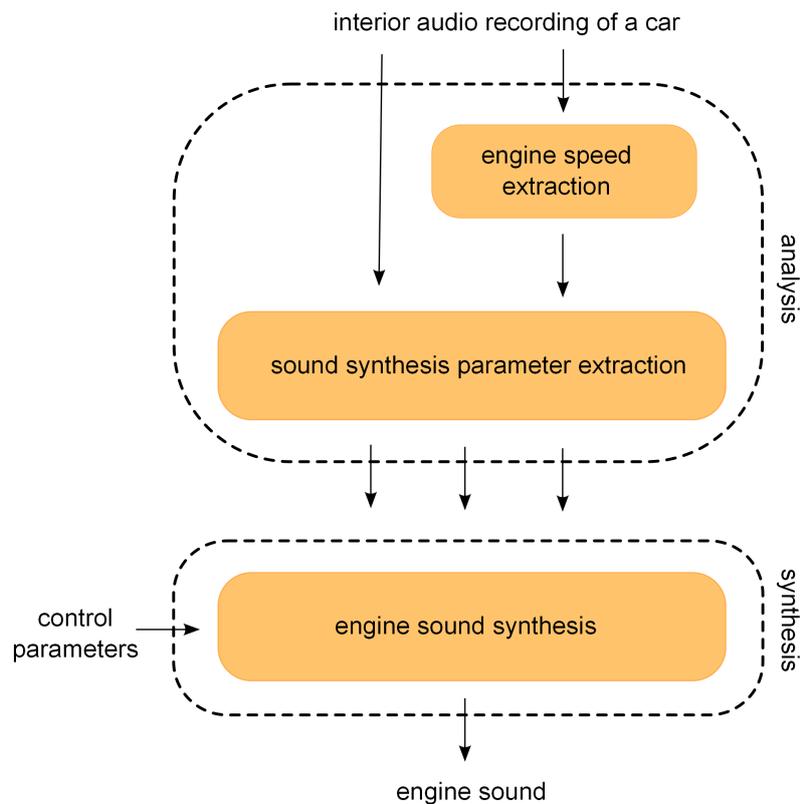
This project report represents a summary of the work performed within this project with its final results. The report is divided into several sections with the following content:

- In section 2, the engine sound synthesis model will be described, which forms the scope for the analysis method.
- The analysis method for analysing existing engine sounds and extracting all necessary sound synthesis parameters for the model is described in section 3. In the first part of this section, the developed engine speed estimation procedure as part of the analysis method is explained in detail. The second part deals with the actual

sound synthesis parameter determination, which is based on Vold-Kalman filters.

- In section 4 of this report, the written engine sound design software is presented and explained.

In order to get a better overview of the whole analysis-synthesis procedure, which is used to analyse an existing engine sound to determine synthesis parameters for the engine sound model, the bloc diagram in figure 1 visualizes the analysis and synthesis part of the system:



**Figure 1:** Block diagram of the analysis-synthesis system for the engine sound model. The analysis part consists of an engine speed estimation from an interior audio recording of a vehicle, which serves as input for the sound synthesis parameter extraction together with the engine speed estimation. The determined sound synthesis parameters are then sent into the synthesis model, which (re)creates an actual engine sound.

## 2 The sound synthesis model

In this chapter, the main parts of the sound synthesis model for creating the engine sound of a motor vehicle will be described. The theory behind the model is of vital importance, since it defines the synthesis parameters and furthermore the requirements for the analysis method for extracting lookup tables of these parameters from the driving noise of a car.

The model, as described in this work, had been developed at the Institute of Electronic Music and Acoustics (IEM) till the end of 2012. A brief description of the synthesis model from a software user's point of view can be found in [1].

### 2.1 Engine orders

In order to describe the core principle behind the model, let us imagine the spectrogram of an interior audio recording of a sports car during an engine run-up<sup>1</sup> at the driver's seat, as depicted in figure 2. With increasing engine speed over time, the fundamental frequency as well as higher partial tones, which are called engine orders in motor engineering, change in terms of frequency and magnitude. The frequency  $f_{o,1}$  of the first order and multiple integers ( $f_{o,k}$ ) of it are defined as:

$$f_{o,k} = k \frac{f_{rot}}{60} \quad (1)$$

Besides these so called full engine orders, the frequency  $f_{o,k}$  of half engine orders can be obtained with equation 1 and  $k = \{0.5, 1.5, 2.5, \dots\}$ . The variable  $f_{rot}$  represents the rotational speed (or engine speed) of the engine in revolutions per minute.

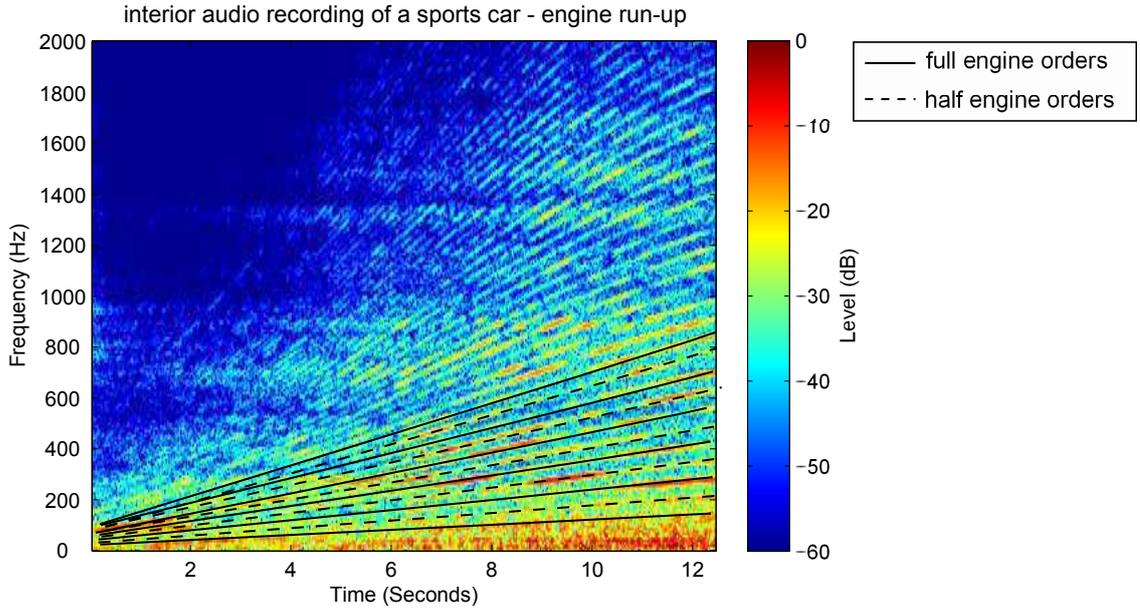
The kind of engine and its number of cylinders determines the general presence of certain engine orders [2], whereas one prominent order originates from the normal firing frequency, which equals to *number of cylinders*/2. The V8 engine of the sports car example in figure 2 points out, that engine orders of 15 and higher can be measured.

With this example in mind, one can define the following basic requirements for the sound synthesis model:

1. The model shall be capable of generating a high number of engine orders (partial tones), which are superimposed by using the principle of additive synthesis.
2. Each engine order can vary in terms of magnitude and frequency with respect to the engine speed and the torque.
3. Since the model should be applicable in real-life driving situations, the synthesis of the engine sound has to be done in real-time, whereas the resulting sound is dependent on the actual engine speed and torque of the engine.

---

1. starting from the smallest possible engine speed and constantly increasing to the highest possible engine speed with full torque



**Figure 2:** Spectrogram of an engine run-up of a sports car at the driver's seat. The rising full engine orders (1<sup>st</sup> to 6<sup>th</sup> are marked with black solid lines) as well as the half engine orders (2<sup>nd</sup> to 6<sup>th</sup> are marked with black dashed lines) are clearly distinguishable.

## 2.2 Synthesis of engine orders

A core task can be derived from the requirements defined in section 2.1 - the synthesis of a single engine order. Based on the example in figure 2, it can be assumed that an engine order is a narrowband signal with its spectral peak at the angular frequency  $\omega_C$ . The basic equation for a single order can be defined as:

$$x(t) = \hat{x}_C \cos(\omega_C t) \quad (2)$$

Since this sinusoidal waveform sounds very unnatural, the signal's peak amplitude  $\hat{x}_C$  as well as its momentary phase, determined by  $\omega_C t$ , will be modulated. Introducing the amplitude modulation leads to:

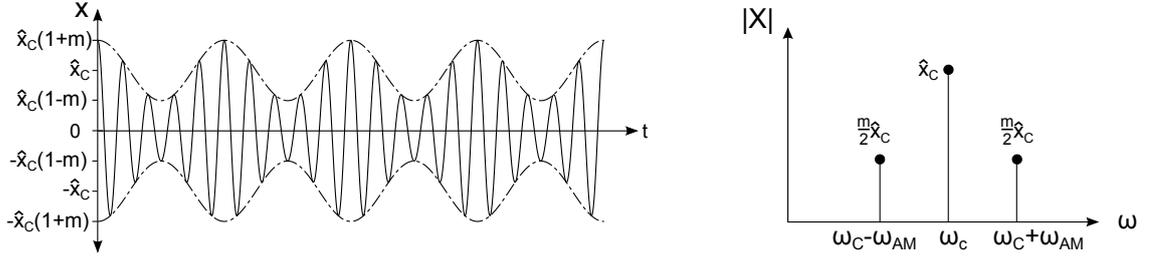
$$x(t) = [\hat{x}_C + \hat{x}_{AM} \cos(\omega_{AM} t)] \cos(\omega_C t) \quad (3)$$

The parameter  $\hat{x}_{AM}$  stands for the peak amplitude of the modulator,  $\omega_{AM}$  describes the amplitude modulation angular frequency. Adding the phase modulation to equation 3, the final formula for an engine order is:

$$x(t) = [\hat{x}_C + \hat{x}_{AM} \cos(\omega_{AM} t)] \cos[(\omega_C t) + \Delta\phi \cos(\omega_{PM} t)] \quad (4)$$

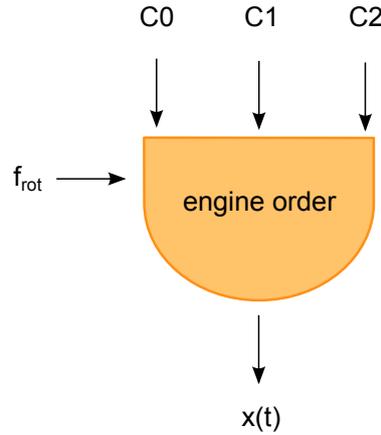
$\Delta\phi$  represents the phase deviation and the parameter  $\omega_{PM}$  stands for the phase modulation angular frequency. Figure 3 shows a graphical summary of a single sinusoidal with

a modulated amplitude.



**Figure 3:** Temporal representation (left) and magnitude spectrum (right) of a sinusoidal carrier amplitude modulated by a sinusoidal signal.  $m$  represents the ratio between the peak amplitude of the modulator and the carrier ( $\frac{\hat{x}_{AM}}{\hat{x}_C}$ ).

From equation 4,  $\hat{x}_C$ ,  $\frac{\hat{x}_{AM}}{\hat{x}_C}$  and  $\Delta\phi$  are declared as the synthesis parameters  $C0$ ,  $C1$  and  $C2$  for the engine sound synthesis model. In combination with the angular frequency  $\omega_C$ , which is a function of the control parameter engine speed  $f_{rot}$ , the signal  $x(t)$  for a single engine order can be synthesized with the following oscillator:



**Figure 4:** Oscillator of a single engine order - the synthesis parameter  $C0$  represents the peak amplitude of the carrier,  $C1$  the amplitude modulation depth and  $C2$  stands for the phase deviation. The angular frequency  $\omega_C$  is controlled by the engine speed  $f_{rot}$ .

By adding a random signal  $N(t)$  to the synthesis parameters for the amplitude and phase modulation, the resulting equation for the waveform created by the oscillator can be derived from equation 4 and is written as:

$$x(t) = [C0 + (C1 + N(t))C0\cos(\omega_{AM}t)]\cos\left[\left(\frac{2\pi f_{rot}}{60}t\right) + (C2 + N(t))\cos(\omega_{PM}t)\right] \quad (5)$$

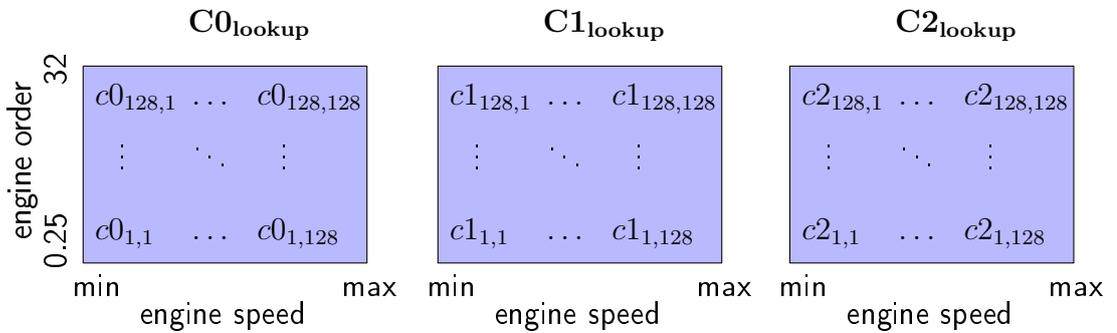
For the synthesis of a complete engine sound, the model for a single engine order has to be applied multiple times. The version of the model used within the project contains 128 oscillators in total. The oscillator  $i$  is used to synthesize a specific engine order with the number  $k$  according to the following relationship:

$$k = 0.25i \quad (6)$$

Furthermore, the synthesis parameters for the oscillator  $i$  are defined as  $C0_i$ ,  $C1_i$  and  $C2_i$ . The determination of those parameters is done via interpolation between lookup tables, which are described in the next section.

### 2.3 Determination of the synthesis parameters

The first part of this section deals with the derivation of the lookup tables, which serve as a source for obtaining the actual values of  $C0_i$ ,  $C1_i$  and  $C2_i$  for the oscillators. Based on the total number of oscillators, the column vectors  $\mathbf{0}_{\text{lookup}}$ ,  $\mathbf{c1}_{\text{lookup}}$  and  $\mathbf{c2}_{\text{lookup}}$  with 128 elements can be defined. In addition, the three synthesis parameters are dependent on the actual engine speed  $f_{rot}$ . For this reason, the interval between the minimum and maximum possible engine speed of the engine is divided into 128 equally spaced bins. With this in mind, the vectors  $\mathbf{c0}_{\text{lookup}}$ ,  $\mathbf{c1}_{\text{lookup}}$  and  $\mathbf{c2}_{\text{lookup}}$  can be extended to three  $128 \times 128$  matrices  $\mathbf{C0}_{\text{lookup}}$ ,  $\mathbf{C1}_{\text{lookup}}$  and  $\mathbf{C2}_{\text{lookup}}$ , which are depicted in figure 5:



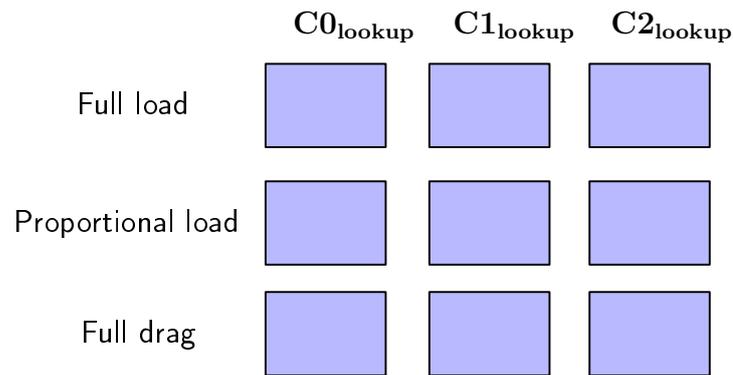
**Figure 5:** Synthesis parameter matrices  $\mathbf{C0}_{\text{lookup}}$ ,  $\mathbf{C1}_{\text{lookup}}$  and  $\mathbf{C2}_{\text{lookup}}$  - the row index  $i$  corresponds to the engine order with the number  $k$  equal to  $0.25i$ . The column index indicates the synthesis parameter at a specific engine speed bin.

In the second part of this section, the actual torque  $M$  of the engine will be taken into account. Therefore, the three torque states "full load", "proportional load" and "full drag" have been defined. Table 1 provides the torque value of those states as well as an example when they may occur:

**Table 1:** Definition of the three torque states

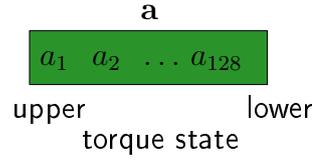
| State name        | Torque $M$ | Example                                                  |
|-------------------|------------|----------------------------------------------------------|
| Full load         | $\max(M)$  | Engine run-up: maximum acceleration starting from 0 km/h |
| Proportional load | $M = 0$    | Idle speed: ungear while driving at constant speed       |
| Full drag         | $\min(M)$  | Switching to a low gear while driving at a high speed    |

While  $\max(M)$  is greater than 0, the minimum of  $M$  is assumed to be smaller than 0. For covering the three different states with different lookup tables, a set of three matrices as in figure 5 is used for every state. The resulting nine matrices are shown in figure 6:



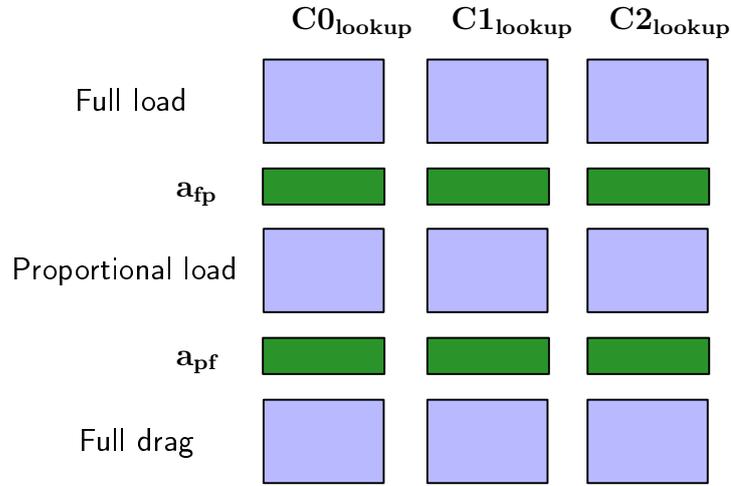
**Figure 6:** Separate set of 3 matrices for the torque states "full load", "proportional load" and "full drag".

Every set of matrices represents a specific torque state. If now a torque value between two states occurs, the actual synthesis parameters  $C0$ ,  $C1$  and  $C2$  for the oscillator  $i$  are determined by interpolating between a value of the row  $i$  at a specific engine speed bin (determined by the actual engine speed) of the lookup matrices of the first torque state and the second torque state. Furthermore, the interpolation between two states can be controlled by the so-called weighting factor  $a$ . Since there is an individual weighting for every transition between two states, a total of 6 factors exists. By dividing the range between two torque states in 128 equally spaced bins, the weighting factors get extended to vectors with the following shape:



**Figure 7:** Shape of a weighting factor vector between two torque states

Combined with all 9 matrices, the synthesis model can be visualized as in figure 8:



**Figure 8:** Separate set of 3 matrices (violet) for the torque states "full load", "proportional load" and "full drag" as well as the 6 different weighting vectors (green) between two torque states (index  $fp$  - "full load" and "proportional load", index  $pf$  - "proportional load" and "full drag").

For a better understanding of the whole determination procedure of  $C0$ ,  $C1$  and  $C2$  for the oscillators, let us assume the following example: the actual torque  $M$  equals to  $\frac{\max(M)}{2}$  and the engine speed  $f_{rot}$  is  $\frac{\max(f_{rot})}{4}$ . Table 1 indicates that  $M$  lies in the middle between the states "full load" and "proportional load". The values for  $C0$ ,  $C1$ , and  $C2$  for the oscillator with the number  $i$  are then determined with:

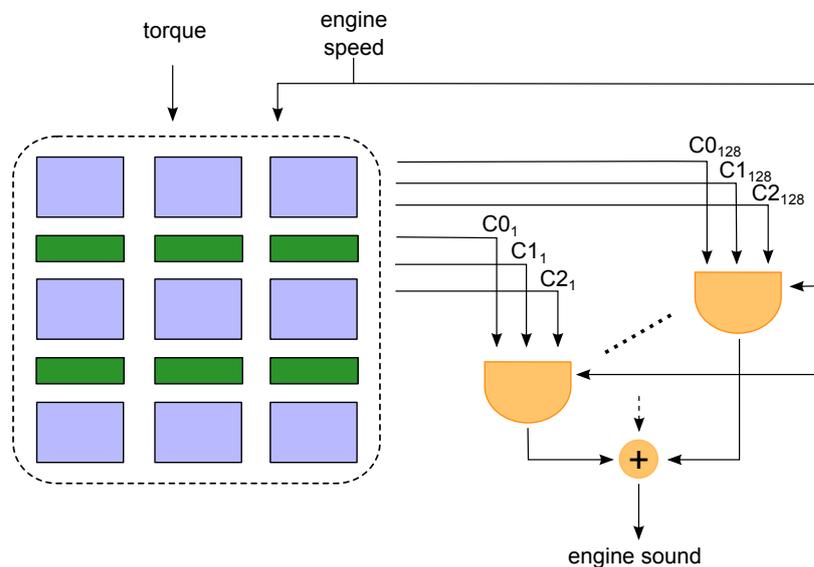
$$\begin{aligned}
 C0_i &= a_{fp,C0,64}C0_{FullLoad,i,32} + (1 - a_{fp,C0,64})C0_{ProportionalLoad,i,32} \\
 C1_i &= a_{fp,C1,64}C1_{FullLoad,i,32} + (1 - a_{fp,C1,64})C1_{ProportionalLoad,i,32} \\
 C2_i &= a_{fp,C2,64}C2_{FullLoad,i,32} + (1 - a_{fp,C2,64})C2_{ProportionalLoad,i,32}
 \end{aligned} \tag{7}$$

Since the control parameters  $f_{rot}$  and  $M$  are both time-dependent, the actual synthesis parameters for the oscillators will be updated accordingly.

With this last step, the sound synthesis procedure is fully described and can be summarized with the block diagram in figure 9. Furthermore, the synthesis model can be characterized by the following 3 aspects:

1. The synthesis model consists of 128 oscillators. The output of each oscillator is determined by the synthesis parameters called  $C0$ ,  $C1$  and  $C2$  and the engine speed  $f_{rot}$ . The 3 synthesis parameters stand for the peak amplitude of the oscillator, the amplitude modulation depth and the phase deviation of the phase modulation. The resulting engine sound equals to the superposition of the output signal of all oscillators.
2.  $C0$ ,  $C1$  and  $C2$  depend on the actual engine speed  $f_{rot}$  as well as the actual torque  $M$ , which are the control parameters.
3. The determination of the actual synthesis parameters  $C0$ ,  $C1$  and  $C2$  is done by interpolating between values of two different lookup matrices, whereas each of them represents a specific torque state. In addition, the interpolation is influenced by a weighting factor  $a$ , which depends on the actual torque  $M$ .

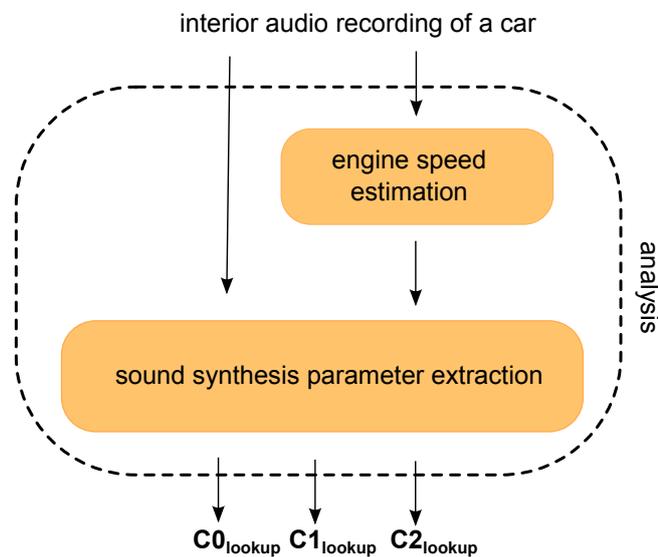
The last mentioned point shows that the determination of the synthesis parameters requires lookup matrices and weighting vectors, as depicted in figure 8. Only the content of these matrices and vectors describes a certain engine sound for the interior of a vehicle. In order to obtain a holistic description of an engine sound, a proper analysis method is needed, which will be described in the next section.



**Figure 9:** Visualization of the sound synthesis model, which consists of 9 lookup matrices (violet) and 6 weighting vectors (green). The engine speed and the torque serve as control parameters and determine the exact selection of the actual synthesis parameters from the lookup matrices. The chosen parameters  $C0$ ,  $C1$  and  $C2$  as well as the engine speed are used as input for a single oscillator which synthesizes one engine order. The final engine sound is obtained by superimposing the output of all 128 oscillators.

### 3 The analysis method

The present section deals with the analysis method, whose final result is the characterisation of an engine sound with lookup matrices as described in chapter 2. Therefore, meaningful values for the sound synthesis parameters  $C_0$ ,  $C_1$  and  $C_2$  have to be extracted from an interior audio recording of a car. According to the structure of the lookup matrices, the synthesis parameters must be determined for the widest possible range of the engine speed  $f_{rot}$ . Since an audio recording obviously does not provide any direct information about the actual engine speed, one important step before the main analysis procedure is the extraction of  $f_{rot}$  from the audio recording. The engine speed and the audio recording itself then serve as input for the main analysis procedure, which is depicted as a part of the block diagram in figure 10:



**Figure 10:** Block diagram of the analysis procedure. The interior audio recording is used to estimate the engine speed as a function of time. The result of this estimation and the audio recording itself serve as input for the sound synthesis parameter extraction.

1. **Engine speed estimation:** the audio recording will be analysed in order to find spectral information of engine orders, which are used to estimate the engine speed as a function of time for the recording.
2. **Sound synthesis parameter extraction:** a special type of filter called Vold-Kalman filter is used to extract time-domain representations of the single engine orders. The estimated engine speed provides the necessary information to track the engine orders within the signal over time. Every single extracted engine order is then used to calculate the sound synthesis parameters  $C_0$ ,  $C_1$  and  $C_2$ , which represent the content of the lookup matrices  $C_{0\_lookup}$ ,  $C_{1\_lookup}$  and  $C_{2\_lookup}$ .

Since the characteristics of an interior audio recording of a car depend on numerous factors like engine speed or torque, the following aspects were defined as requirements

within this project:

1. The torque  $M$  of the engine shall be close to one of the torque states defined in table 1: "full load", "proportional load" or "full drag".
2. The audio recording shall provide acoustic information about the engine for a wide range of the engine speed  $f_{rot}$ . A continuously increasing or decreasing engine speed is desirable but not mandatory.
3. The audio recording shall contain as little noise as possible. In context of this analysis, sound which does not originate from the engine itself or vibrating parts excited by the engine, is declared as noise.

An example, which fulfils these requirements, has been described in section 2.1 and depicted in figure 2.

### 3.1 Engine speed estimation

The estimation of the engine speed from an audio recording of a vehicle can be interpreted as a pitch detection and furthermore a pitch tracking problem with the fundamental frequency  $f_0 = \frac{f_{rot}}{60}$ , which is equivalent to the frequency of the 1<sup>st</sup> engine order  $f_{o,1}$ . Since pitch tracking is used in the field of speech recognition, several algorithms like [3] and [4] exist. All of these pitch detection and tracking methods consist of a complex algorithm, which contains several pre- and post-processing steps. However, the majority of those are based on basic methods for pitch detection, which are described in the next section.

#### 3.1.1 Basic methods of pitch detection

Most advanced pitch detection and tracking algorithms are based on one of three basic approaches, which are summarized in [5] and [6], and explained in this section.

An important method of pitch detection uses the **autocorrelation function**. Expressed as a discrete function, it can be described as the sum of multiplications of a time window with the length  $N$  with its shifted version:

$$r(l) = \sum_{n=0}^{N-1} x(n)x(n-l) \quad (8)$$

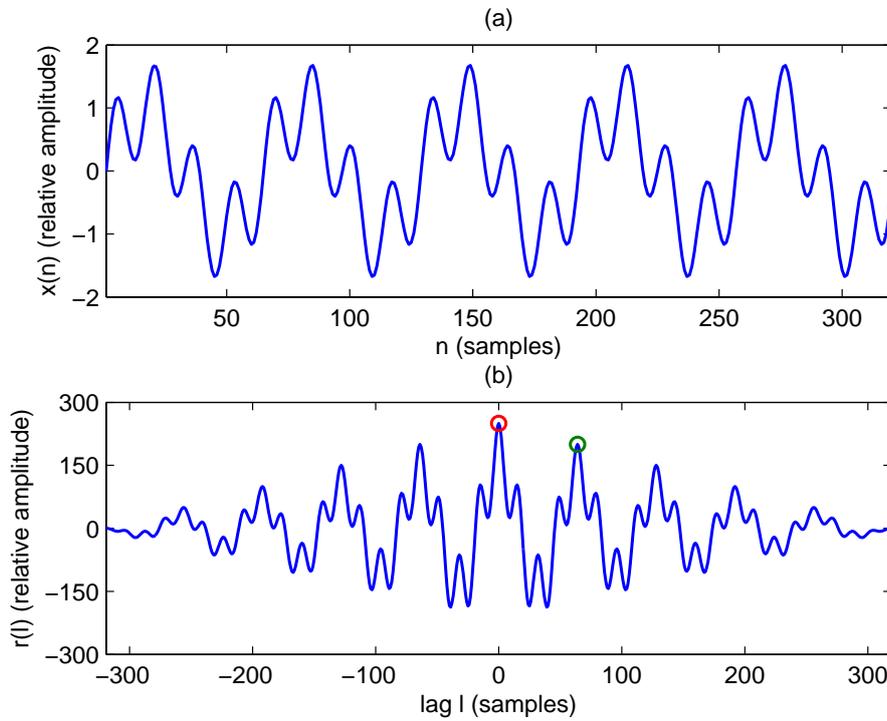
The parameter  $l$  stands for the time lag and leads to the maximum of the autocorrelation function at  $l = 0$ . By defining a minimum lag  $l_{min}$  for the highest detectable frequency, the maximum peak of the autocorrelation function with a lag greater than  $l_{min}$  corresponds to the fundamental period  $t_0$ :

$$t_0 = \arg(\max(r(l))) \quad \text{with } l > l_{min} \quad (9)$$

A simple example for determining  $t_0$  is depicted in figure 11. The reciprocal of  $t_0$  leads to the "discrete" fundamental frequency  $\tilde{f}_0$ , which differs from the exact frequency  $f_0$ , since equation 9 leads to discrete values. This error can be characterized by the frequency error factor  $\alpha(f_0)$ , given by the ratio between the "discrete" and the exact fundamental frequency:

$$\alpha(f_0) = \frac{f_0}{\tilde{f}_0} = 1 + 0.5 \frac{f_0}{f_s} \quad (10)$$

For a given sampling frequency  $f_s$ , the frequency error factor increases with an increasing fundamental frequency  $f_0$ .



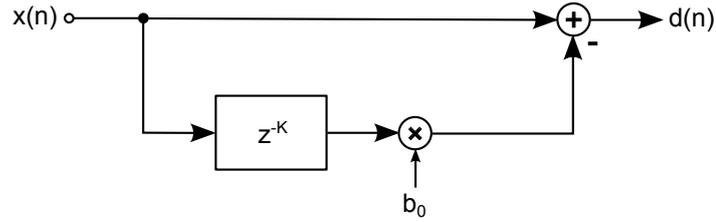
**Figure 11:** Temporal representation (a) of a signal consisting of a sinusoidal with the frequency  $\frac{f_s}{64}$  and a sinusoidal with  $f = \frac{f_s}{16}$ . The corresponding ACF (b) shows the obvious maximum at a lag  $l = 0$  (red circle) and the 2<sup>nd</sup> maximum (green circle) at  $l = t_0 = 64 > l_{min}$ .

One weakness of the autocorrelation function is its vulnerability to octave errors, because its peaks are periodically repeated. Therefore, additional post-processing is of vital importance to increase the robustness of the fundamental frequency detection. An advanced algorithm based on the autocorrelation function is described in [7].

The **long term prediction** approach represents another method for pitch detection. It uses a delay of  $K$  samples followed by a short FIR prediction filter with the filter coefficients  $b_q$ . The long-term residual signal  $d(n)$  is given by:

$$d(n) = x(n) - \sum_{q=0}^Q b_q x(n - K - q) \quad (11)$$

A usual filter order for this application lies between 1 and 3. The resulting block diagram for a 1<sup>st</sup>-order filter is depicted in figure 12:



**Figure 12:** Long-term prediction with a 1<sup>st</sup>-order FIR filter.

Furthermore, equation 11 can be simplified to:

$$d(n) = x(n) - b_0 x(n - K) \quad (12)$$

The minimisation of  $d^2$  for a block of the length  $N$  can be achieved by calculating  $b_0$  according to:

$$b_0 = \frac{\tilde{r}_{xx}(K)}{r_{xx0}(K)} \quad (13)$$

The exact autocorrelation  $\tilde{r}_{xx}$  considers samples before the block with the length of  $N$  and equals to:

$$\tilde{r}_{xx}(l) = \sum_{n=0}^{N-1} x(n)x(n-l) \quad (14)$$

The divisor in equation 13 depends on  $r_{xx0}$ , which stands for the energy of a delayed block by  $l$  samples:

$$r_{xx0}(l) = \sum_{n=0}^{N-1} x^2(n-l) \quad (15)$$

The energy of  $d$  depending on the lag  $K = l$  can be calculated by using the normalized autocorrelation

$$r_{xx,norm}(l) = \frac{\tilde{r}_{xx}(l)^2}{r_{xx0}(l)} \quad (16)$$

and equals to:

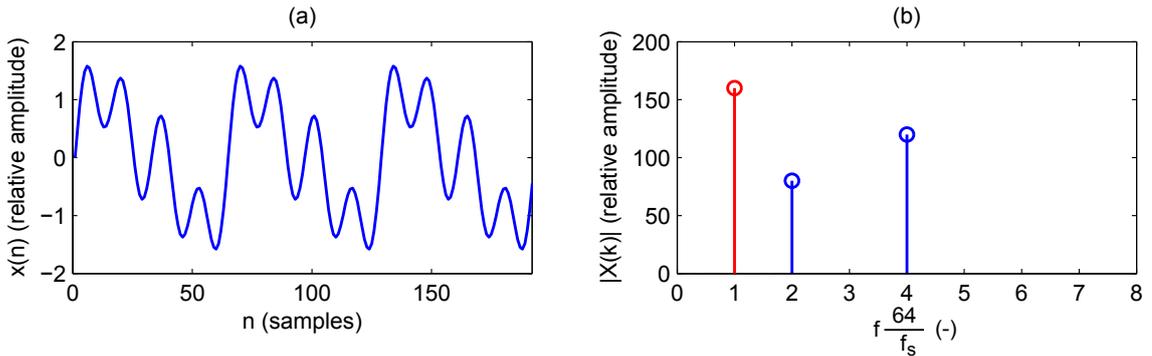
$$d^2 = \sum_{n=0}^{N-1} (x^2(n) - r_{xx, norm}(K)) \quad (17)$$

Since the goal is the minimisation of  $d^2$ , a lag  $l = K$  that maximises the normalized autocorrelation  $r_{xx, norm}(l)$  is needed. The further procedure to determine the final pitch lag candidate is done in 3 steps:

1. Search for local maxima in  $r_{xx, norm}(l)$ .
2. Select maxima of  $r_{xx, norm}(l)$ , where the value of  $\tilde{r}_{xx}(l)$  is positive.
3. Determine the final pitch lag candidates by considering the FIR filter coefficient  $b_0$ . A value close to 1 indicates a voiced sound, whereas a value close to 0 stands for an unvoiced sound. All previous candidates with a  $b_0$  above a certain threshold are final pitch lag candidates.

The final pitch lag is chosen to be the final pitch lag candidate with the lowest lag value.

Besides the two already described time-based approaches, the fundamental pitch can also be determined in the frequency domain via **spectral peak picking**. This approach uses the discrete Fourier transform (DFT) in order to calculate the spectrum of a time frame with the length  $N$ . A peak is simple defined as a local maximum in the magnitude spectrum of the signal. The lowest detected peak is considered as the fundamental frequency  $f_0$ . While the peak detection is a straightforward process for a simple spectrum as in figure 13, complex spectra require advanced peak picking algorithms as described in section 3.1.2.



**Figure 13:** Temporal representation (a) of a signal consisting of a sinusoid with the frequency  $\frac{f_s}{64}$ , a sinusoid with  $f = \frac{f_s}{32}$  and a sinusoid with  $f = \frac{f_s}{16}$ . The corresponding magnitude spectrum (b) shows three peaks, and the peak with the lowest frequency (red) is considered to be the fundamental frequency.

For a reasonable accurate detection of low fundamental frequencies, the time window has to be increased, which results in a poorer temporal resolution. Since the frequency resolution  $\Delta f$  of the DFT is dependent on the frame length, it can be obtained with:

$$\Delta f = \frac{f_s}{N} \quad (18)$$

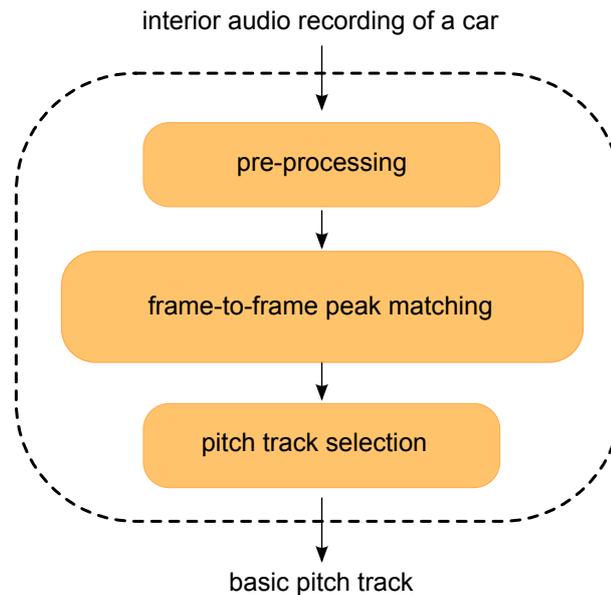
One way to overcome this problem is called zero-padding. By taking a time frame of the length  $N$  and extending it with a series of zeros at the end before computing the DFT, the effective frequency resolution can be increased. Nevertheless, no information is added, since zero-padding represents an interpolation in the frequency domain. It can help to detect a spectral peak, but if two narrowband spectral components lie within the range of one frequency bin before zero-padding is applied, it is still not possible to distinguish them.

Further ways for increasing the frequency resolution of a spectrum are described in [6].

### 3.1.2 Advanced pitch detection and pitch tracking

Based on the fundamental methods of pitch detection as described in the previous section, a suitable advanced algorithm which is capable of detecting and tracking a pitch over time had to be chosen within this project. While the majority of existing pitch detection algorithms focus on the frequency range of speech and musical applications, a 2-cylinder engine at an engine speed of 900 rpm has a fundamental frequency  $f_0$  of 15 Hz. Therefore, an algorithm with the capability of detecting and tracking very low fundamental frequencies is needed.

The finally chosen pitch detection and tracking method operates in the frequency domain. While the detection part is achieved via spectral peak picking as described in section 3.1.1, the pitch tracking is based on **Frame-to-frame peak matching**, which was originally presented as a part of a speech analysis-synthesis technique by McAulay [8]. The peak matching method analyses the spectral peaks time frame by time frame in order to find continuous pitch tracks, which represent single orders of the engine. Afterwards, a designed weighting procedure is applied to determine the basic pitch track. Figure 14 visualizes the different processing steps of the method that was used:



**Figure 14:** Block diagram of the engine speed estimation procedure, not including post-processing.

Before the actual spectral analysis, the following pre-processing is applied on the interior audio recording:

- The upper limit of the spectral range in the audio recording is reduced to 4800 Hz via decimation. With this limit, all necessary engine order information is preserved, even for the sports car example in figure 2, which is considered as the upper limit with engine order information at higher frequencies within this analysis procedure.
- Spectral information below 15 Hz is removed by a 12<sup>th</sup>-order zero-phase high-pass filter because this range is usually dominated by unwanted rumble noise. No A-weighting is applied, since it would also attenuate low engine orders.

The next step includes a conventional short-time Fourier transform (STFT) in order to divide the signal in analysis frames with a length of  $N$  samples for the frame-to-frame peak matching. A longer frame leads to a higher frequency resolution but is also prone to smearing of spectral information, if the spectral information changes within a frame.

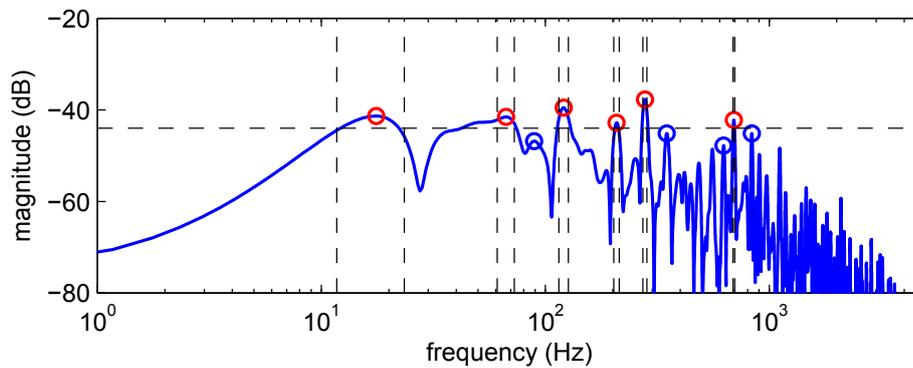
After computing the STFT of the signal, spectral peak picking is applied on each analysis frame. Based on the spectral peak picking described in section 3.1.1, the following additional parameters have to be introduced for peak picking in a more sophisticated spectrum:

**Table 2:** Peak picking parameters and their description

| Parameter             | Description                                                                                    |
|-----------------------|------------------------------------------------------------------------------------------------|
| Minimum peak height   | Minimum value for a local maximum to be considered as a peak.                                  |
| Minimum peak distance | A lower peak within the (horizontal) minimum peak distance of a higher peak will be discarded. |
| Number of peaks       | Only a certain number of the highest local maxima is declared as peaks.                        |

An example for advanced spectral peak picking is shown in figure 15. The "correct" setting of these parameters is influenced by the spectral characteristics of the signal and the individual parameters are strongly dependent on each other, so there is no general rule how to set them in terms of quantity.

The next step includes finding continuous pitch tracks via frame-to-frame peak matching [8] over all analysis frames. By applying the previously described peak picking method with a constant number of peaks, the peaks will vary in terms of frequency (position) and magnitude for every frame. In order to establish a connection between peaks of multiple frames, the concept of "birth" and "death" of pitch tracks is introduced.



**Figure 15:** Spectral peak picking applied on a magnitude spectrum. 4 out of 10 peaks (circles) are discarded (blue circles), since they lie below the minimum peak height (-44 dB, horizontal dashed line). No lower peaks are present within the minimum peak distances ( $\pm 6$  Hz, vertical dashed lines) of the peaks (red circles), therefore no additional peak will be discarded.

Beginning with the analysis frame  $j$ , a certain number of peaks is picked at the frequencies  $f_0^j, f_1^j, \dots, f_{N-1}^j$ . With this action,  $N$  pitch tracks are "born" and taken into account for the next frame. After detecting  $M$  peaks in frame  $j + 1$ , a matching distance will

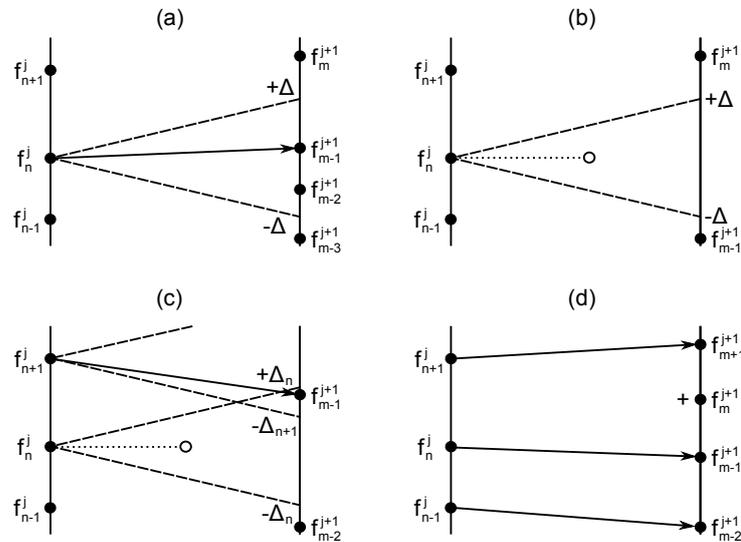
be calculated between every single peak and the last peak of every pitch track that is "alive". This calculation can be written as:

$$\Delta_{nm} = |f_n^j - f_m^{j+1}| \text{ with } n = 0 \dots N - 1, m = 0 \dots M - 1 \quad (19)$$

In addition, every distance  $\Delta_{nm}$  is compared with the matching interval  $\Delta$ :

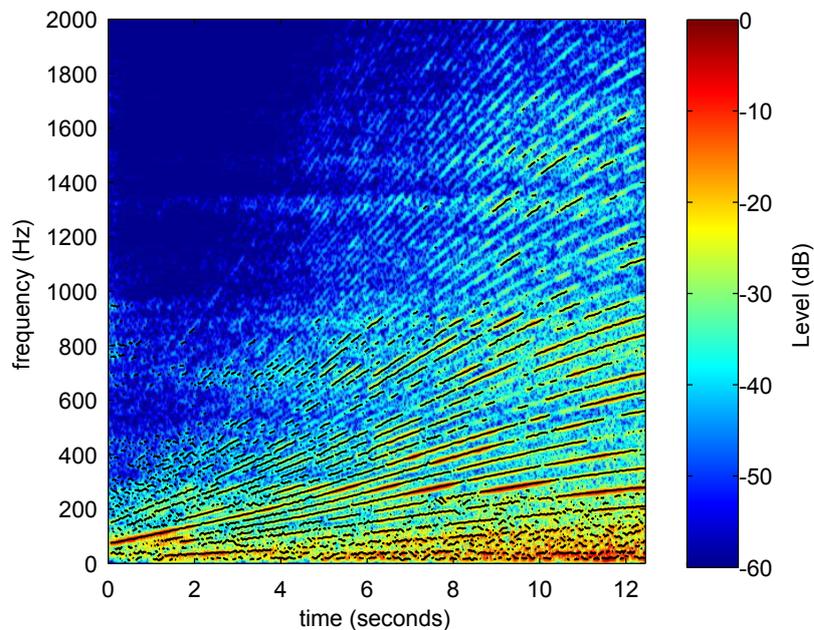
$$\Delta_{nm} \leq \Delta \quad (20)$$

If a "living" pitch track with its last frequency  $f_n^j$  has no matching distance  $\Delta_{nm}$  with  $m = 0 \dots M - 1$  within the matching interval  $\Delta$ , it is declared as "dead". If one or more matching distances are equal or smaller than the matching interval, the frequency  $f_m^{j+1}$  with the smallest distance is added to the pitch track and extends the track. By executing this procedure for every pitch track, it is possible that several tracks have chosen the same frequency  $f_m^{j+1}$ . In this case, the pitch track with the smallest distance  $\Delta_{nm}$  is extended by the frequency. The other pitch tracks chose their next favoured frequency or are declared as "dead", if there is no remaining frequency in the interval  $\Delta$ . In addition, for every frequency  $f_m^{j+1}$  which has not been added to an existing pitch track, a new pitch track is "born". Figure 16 visualizes these different scenarios of the frame-to-frame peak matching procedure.



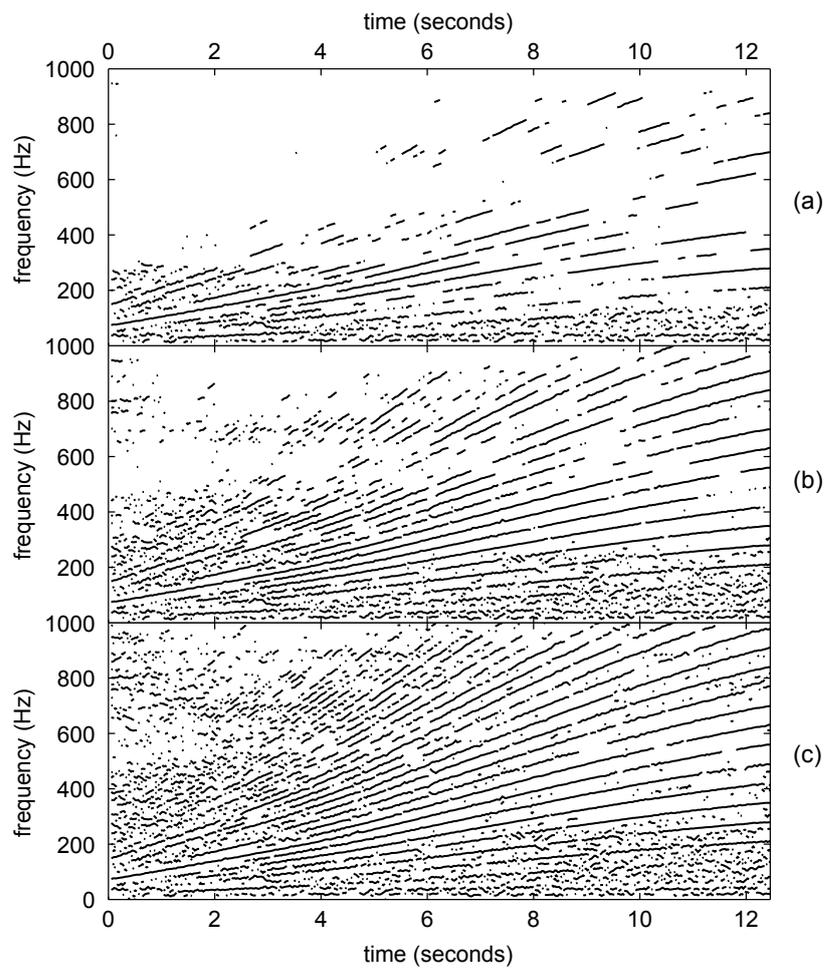
**Figure 16:** Different scenarios for frame-to-frame peak matching. If the distance between the pitch track with the frequency  $f_n^j$  and  $f_{m-1}^{j+1}$  is smaller than any other possible distance and lies within the matching interval (outlined by dashed lines), the pitch track is extended by  $f_{m-1}^{j+1}$  (a). The pitch track is declared as "dead", if there are no frequencies within the matching interval (b). If a frequency is favoured by two existing pitch tracks, the one with the smallest matching distance will be extended. The remaining pitch track is declared as "dead", if no other frequencies are within the matching interval (c). The last scenario (d) shows the "birth" of a new pitch track, if a detected frequency has not been occupied by an existing pitch track.

By analysing the selected peaks in every time frame of the STFT, a high number of pitch tracks with different lengths are created. The example with 20 peaks per analysis frame depicted in figure 17 points out that established pitch tracks can be divided into two groups. The first includes tracks containing meaningful information about an engine order where every track optimally consists of a large number of connected peaks. Unwanted noise on the other hand can lead to very short pitch tracks. Tracks of this second group are characterised through a shape uncorrelated to the actual engine speed. An increase of the peak picking parameter "number of peaks" leads to more pitch tracks with engine speed information as well as to more pitch tracks uncorrelated to the engine speed, as depicted in figure 18. One could now argue that the more number of peaks, the better - but a large number of uncorrelated pitch tracks can cause problems for the post-processing, which is described in section 3.1.3. For the set of interior audio recordings provided for this project, 20 peaks per analysis frame has proven to be a meaningful number.

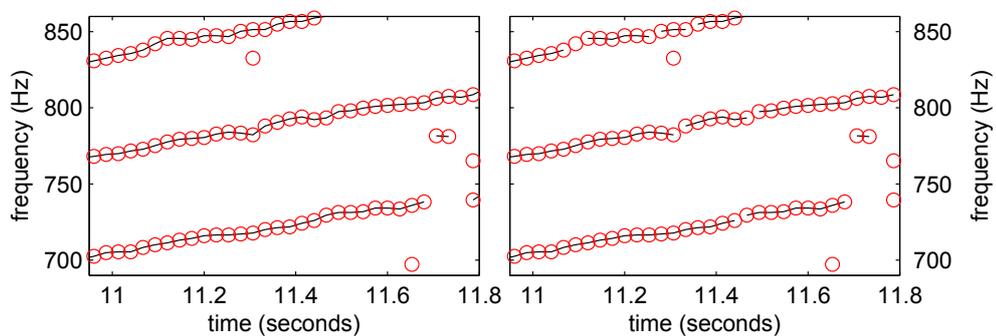


**Figure 17:** Graphic pitch track representation (black lines) after applying the frame-to-frame peak matching procedure on the STFT of an engine run-up - number of peaks = 20.

Another relevant parameter of the algorithm represents the matching interval  $\Delta$  in hertz. Its minimum value can be derived from the length of the DFT used in the STFT including zero-padding. Increasing the matching interval results in a more robust pitch tracking with a higher possibility of adding a "wrong" peak to the pitch track. This is particularly critical, if the recording contains noise and the spectral peaks of the noise get detected by the peak picking procedure. In case of a too low matching interval on the other hand, the tracking of an obvious visible engine order is interrupted, as depicted in figure 19.



**Figure 18:** Pitch tracks after applying the frame-to-frame peak matching procedure with 10 peaks (a), 20 peak (b) and 30 peaks (c) on the STFT of an engine run-up. A higher number of peaks leads to more pitch tracks with information about the engine speed as well as to more short pitch tracks uncorrelated to the engine speed.



**Figure 19:** Close look at pitch tracks (black lines with a red circle for every peak) after applying the frame-to-frame peak matching procedure with 20 peaks and a matching interval of 6 Hz (left) and 3 Hz (right). At 3 Hz, several pitch tracks were not detected as one continuous pitch track.

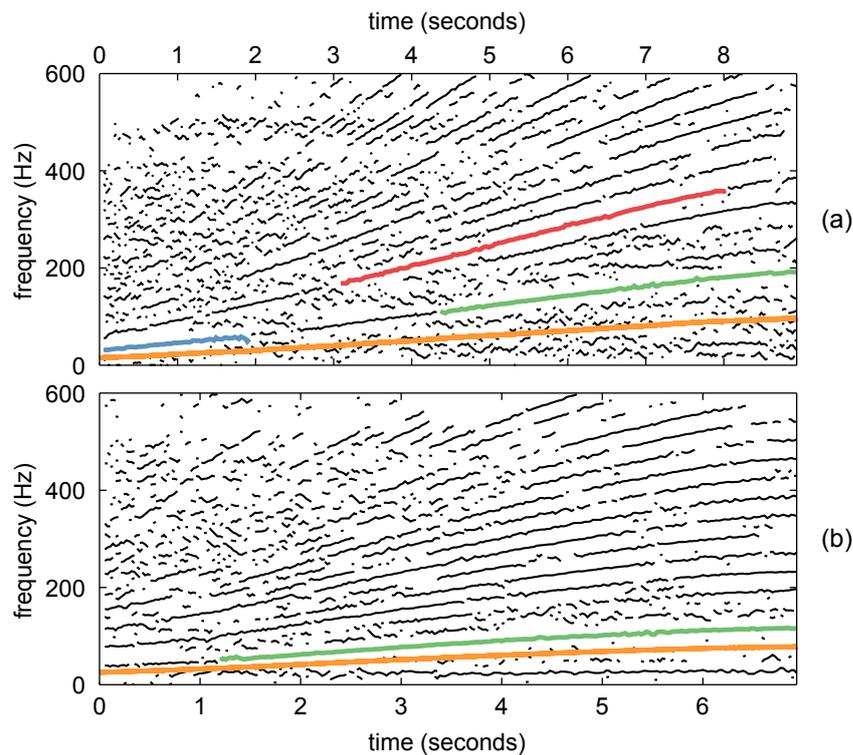
A matching interval of 6 Hz has proven to be a meaningful compromise between adding "wrong" peaks and not detecting obvious peaks to the pitch track. The latter can be overcome by additional post-processing, which is described in the next section.

After applying the frame-to-frame peak matching algorithm, a pitch track has to be chosen as basis for the engine speed estimation. The analysis of a database containing 98 interior audio recordings of engine run-ups from combustion engine vehicles with Vold-Kalman filters has shown that in 97 % of all cases the extracted engine order with the highest energy equals to the *number of cylinders/2*. By finding the frequency of this order as a function of time and with a known number of cylinders, the engine speed can be estimated. For the determination of the basic pitch track, three different selection methods were evaluated:

- a) The longest pitch track will be selected
- b) The pitch track with highest total energy will be selected
- c) The pitch track with the highest product of the total energy and the length will be selected

The application of these 3 criteria on the recordings of the database has shown that even method a) detected a meaningful basic pitch track in 85 % of all recordings. Its main weakness is the selection of very long low-energy pitch tracks of engine orders higher than *number of cylinders/2*. Taking only the energy of a pitch track into account as in method b) leads to a detection rate of 89 %. In the remaining 11 % the selection was influenced by a high-energy low-frequency noise or a high-energy low-frequency pitch track. A combined weighting of track length and energy as in method c) detected a suitable basic pitch track in 96 % of all recordings and was therefore chosen as the final method.

In an optimum case, the selected pitch track extends over the entire record and thus can be used to estimate the engine speed without further processing. In 94 % of the recordings from the database, the basic pitch track covers less than 90 % of the recording, like in the examples in figure 20 (b). In order to derive a pitch track that covers at least 90 % of the recording, which has been defined as a very good result, additional post-processing to extend the pitch track is necessary.

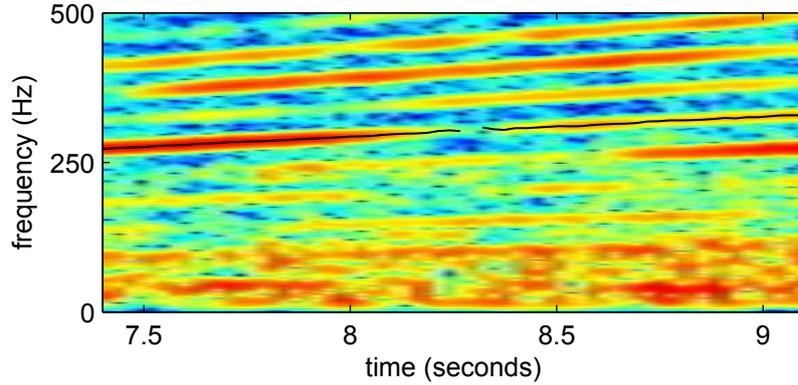


**Figure 20:** (a) Pitch tracks of a vehicle with a 4-cylinder gasoline engine after applying all 3 basic pitch track selections methods (method a) - red, method b) - blue, method c) - green) and true engine speed (orange) in revolutions per second. While methods b) and c) select a pitch track with information about the desired 2<sup>nd</sup> engine order, method a) selects a long pitch track belonging to a higher order. (b) Pitch tracks of a vehicle with a 3-cylinder diesel engine after applying method c).

### 3.1.3 Post-processing

Based on the selected pitch track with the highest product of total energy and the length as a result of the frame-to-frame peak matching procedure, several post-processing methods have been developed and tested to extend the basic pitch track. For this reason, all tracks which have not been chosen in the first place will be investigated as to whether they can be used to extend the basic pitch track.

The first post-processing procedure that was designed and evaluated has been named **pitch track gluing** and addresses scenarios, where a long pitch track is interrupted by a peak gap, as depicted in figure 21. This gap can be caused by engine-speed-dependent acoustic antiresonances in the interior of the car or just by a too high matching distance between two peaks, which was introduced with equation 19 in the last section. In order to fill this gap, all remaining tracks are investigated in three steps. The first includes the withdrawal of all pitch tracks below a defined minimum length. Afterwards, pitch tracks equal to or above the minimum length are investigated as to whether their starting



**Figure 21:** Pitch track (black line), which has been interrupted by a peak gap with a length of 54 ms.

or their end point is within the so-called (horizontal) gluing distance  $d$  of the starting or end point of the basic pitch track. Furthermore, the start or end point of a potential track within the horizontal distance  $d$  must lie within an extended vertical matching interval  $\Delta_{glue}$  in hertz, which is determined by the chosen matching interval  $\Delta$  and the gluing distance  $d$  in analysis frames with

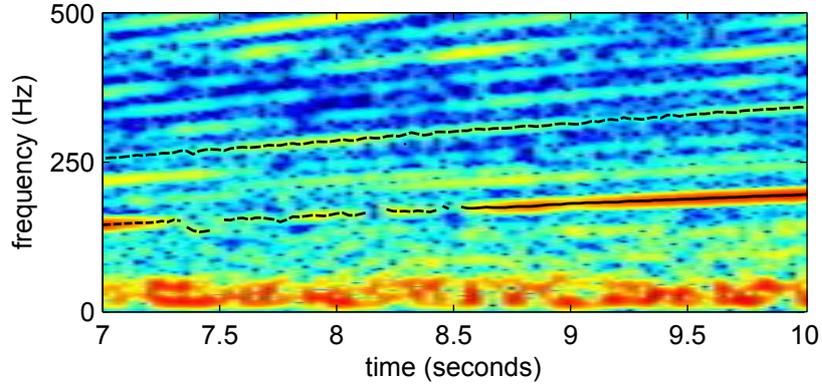
$$\Delta_{glue} = \frac{\Delta}{2}d \quad (21)$$

The divisor of 2 in equation 21 allows higher gluing distances without accidentally taking pitch tracks from higher or lower orders into account. Finally, the longest pitch track candidate which fulfils the horizontal and the vertical condition is chosen to be glued with the basic pitch track. The missing values in the gap are obtained by a simple linear interpolation.

For the chosen length of 27 ms for an analysis frame, a gluing distance of  $d = 10$  has been selected to obtain a total horizontal gluing distance of 270 ms with a vertical matching interval  $\Delta_{glue} = 30$  Hz. Higher gluing distances have been avoided, since a larger temporal gap leaves a higher chance that the unknown engine speed values in the gap have a more complex shape, which would increase the error between the "true" engine speed and the linear interpolation in the gap. Without any post-processing, only 6 % of the recordings of the database contain a pitch track that covers at least 90 % of the recording - after applying one pitch track gluing procedure, 20 % of the recordings contain a pitch track that covers more at least 90 % of the recording. If the gluing is applied more than once, this number can be further increased.

However, depending on the characteristics of a recording, the method of pitch track gluing does not always lead to an extended pitch track. Figure 22 shows an excerpt of a recording, where **pitch track shifting** represents a more efficient way of extending the basic pitch track. This post-processing method tries to find pitch tracks which contain engine speed information in form of a multiple integer of the actual engine speed. For instance, the example in figure 22 shows a continuous pitch track at a higher engine

order above the basic pitch track:



**Figure 22:** Selected basic pitch track (black solid line), which has to be extended. While pitch track gluing from the left would require multiple attempts with only very short pitch tracks (black dashed lines), a very long pitch track (black dashed line above the basic pitch track) provides much more information at twice the frequency compared to the basic pitch track.

In order to use a pitch track of a higher order, a procedure has been designed and implemented, which investigates all remaining pitch tracks of the frame-to-frame peak matching procedure in the following steps:

1. All pitch tracks below the defined minimum length are withdrawn.
2. Every pitch track above the minimum length is investigated as to whether it is horizontally overlapping with the first or last 3 elements of the basic pitch track.
3. The overlapping 3 elements of every pitch track  $f(n)$  are normalized to the corresponding overlapping values of the estimated fundamental pitch track<sup>2</sup> part  $f_{0,est,basic}(n)$  from the basic pitch track and subtracted from a rounded version of the same normalization.

$$\left| \text{round} \left( \frac{f(n)}{f_{0,est,basic}(n)} \right) - \frac{f(n)}{f_{0,est,basic}(n)} \right| \leq 0.3 \quad (22)$$

If the absolute difference between the two normalizations is smaller than 0.3 for all 3 overlapping elements, then the pitch track is considered as a potential candidate for pitch track shifting

4. Finally, the pitch track candidate with the highest length will be chosen for shifting.

---

2. the calculation of this estimation is described in the end of this section

Usually the rounded normalization in equation 22 has the same value for all 3 overlapping elements and indicates the engine order number represented by the chosen pitch track candidate. In order to shift the pitch track to the basic pitch track, every non-overlapping element of  $f(n)$  is shifted with:

$$f_{shift}(n) = f(n) \left( \text{round} \left( \frac{f}{f_{0,est,basic}} \right) \right)^{-1} \frac{\text{number of cylinders}}{2} \quad (23)$$

After applying one pitch track shifting procedure, 11 % of the recordings of the database contain a pitch track that covers at least 90 % of the recording. Since the shifted pitch tracks are often shorter than the one in figure 22, multiple appliance of the procedure is highly recommended. After applying it 3 times, 29 % of the recordings of the database contain a pitch track that covers at least 90 % of the recording.

With these two presented post-processing methods, it is possible to extend the chosen basic pitch track from the frame-to-frame peak matching procedure to obtain a pitch track  $f(n)$  that covers the majority of the audio recording. The last step to the final engine speed estimation consists of simply taking the number of cylinders of the engine into account and converting the result into revolutions per minute

$$f_{rot,est} = \frac{2f(n)}{\text{no of cylinders}} 60 \quad (24)$$

### 3.1.4 Case studies

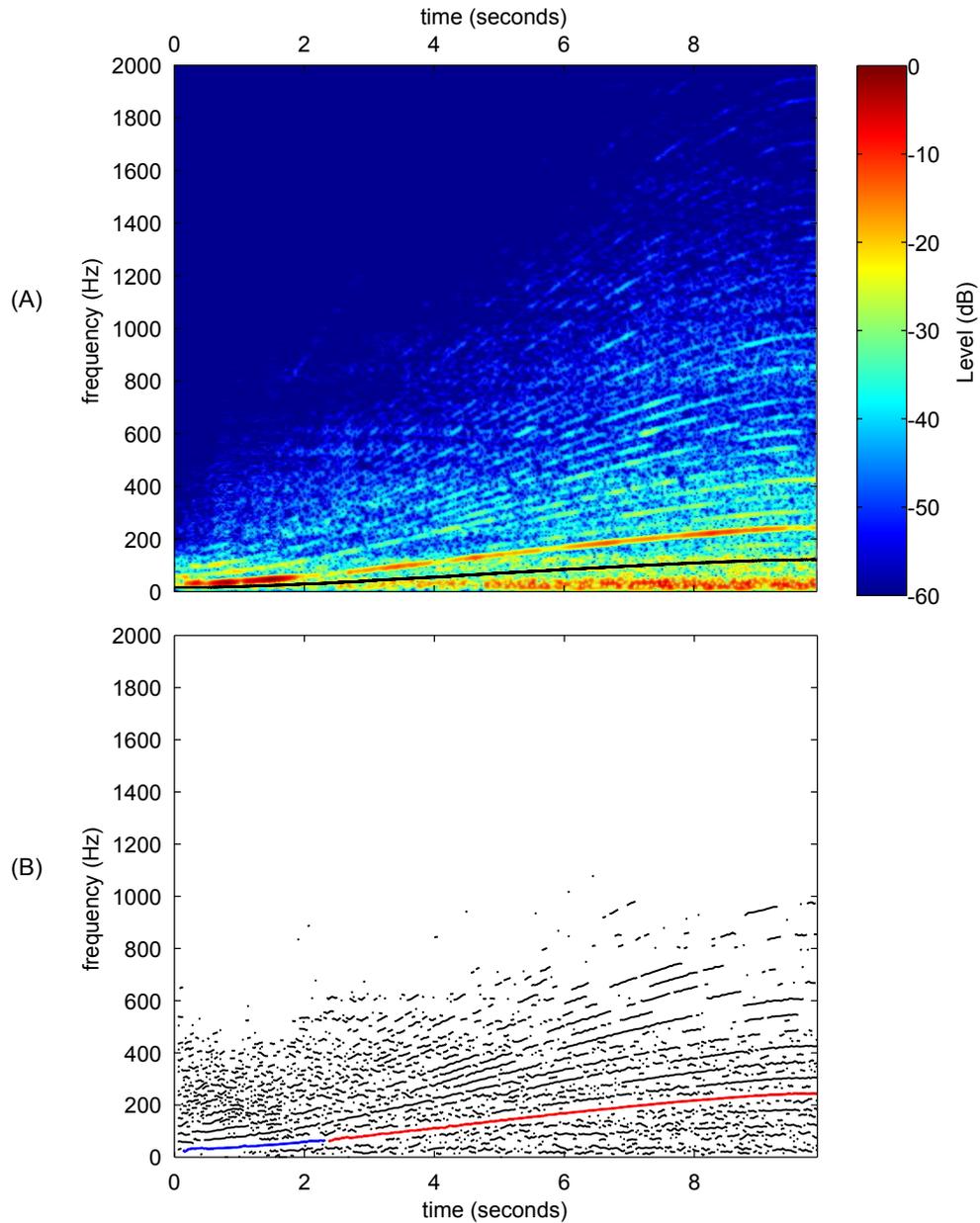
In this section, two exemplary engine speed estimations from audio recordings of a vehicle interior during an engine run-up will be presented. Subsequently, the estimated results will be compared with the directly measured engine speed, which is assumed to be the ground truth.

In the first example, the engine speed  $f_{rot,est}$  as a function of time of a 4-cylinder gasoline engine of a compact car will be estimated. As expected, the corresponding spectrogram of the audio recording in figure 23 (A) shows a very prominent 2<sup>nd</sup> engine order at frequencies twice as high as the fundamental frequency of the engine. The chosen parameters for the applied STFT and for the subsequent frame-to-frame peak matching procedure (explained in table 2) are listed in table 3.

Without additional post-processing, the basic pitch track provides engine speed information for 75 % of the recording. After applying the pitch track gluing procedure once, the extended basic pitch track contains information for a remarkable 98 % of the recording. The final engine speed estimation  $f_{rot,est}$  shows only little deviations from the measured engine speed  $f_{rot}$ , as depicted in figure 24. In order to rate the overall performance of the engine speed estimation algorithm, the mean quadratic deviation  $d^2$  of  $f_{rot,est}$  from the measured engine speed  $f_{rot}$  in Hz has been defined as

$$d^2 = \frac{1}{N} \sum_{n_1=j}^{n_2=k} \left( \frac{f_{rot}(n) - f_{rot,est}(n)}{60} \right)^2 \quad (25)$$

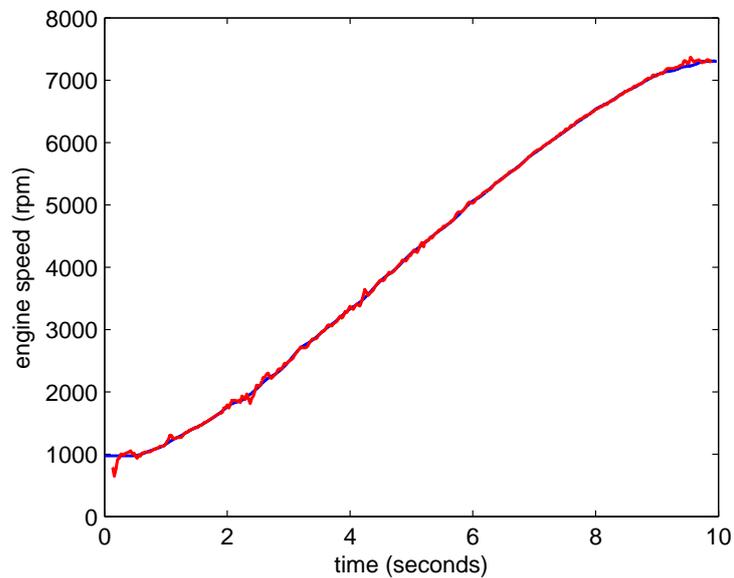
with the indexes  $j$  and  $k$  as the interval, where an engine speed estimation is available. For the presented example,  $d^2$  equals to 0.33 Hz or 20 rpm.



**Figure 23:** (A): Spectrogram of an engine run-up of a 4-cylinder gasoline engine vehicle and the corresponding measured engine speed (black solid line). (B): Pitch tracks (black lines) after applying the frame-to-frame peak matching procedure with the selected basic pitch track (red line) and the chosen pitch track candidate (blue line) for pitch track gluing.

**Table 3:** Chosen engine speed estimation parameters

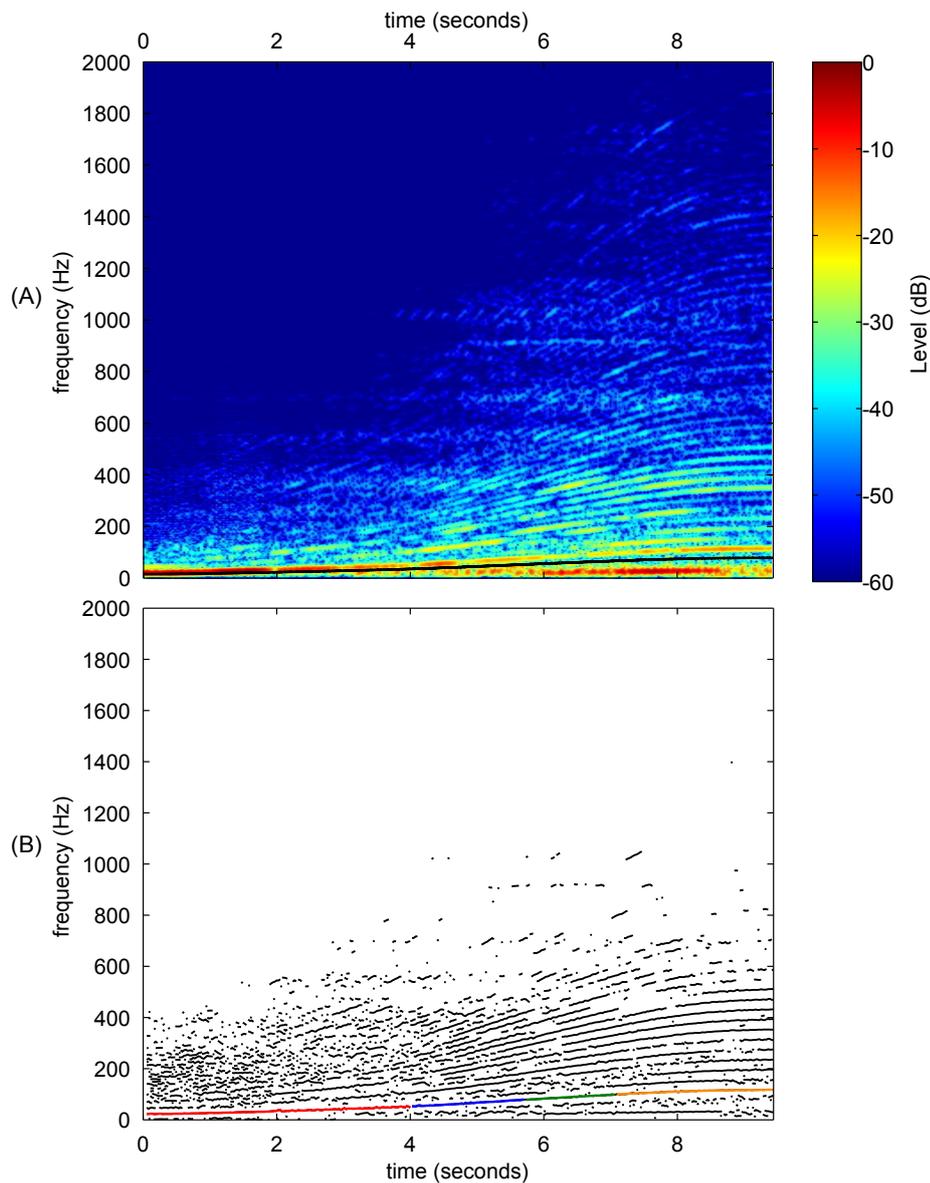
| Parameter                                  | Value      |
|--------------------------------------------|------------|
| Analysis frame length                      | 27 ms      |
| Minimum peak height                        | $10^{-10}$ |
| Number of peaks                            | 20         |
| Matching interval $\Delta$                 | 6 Hz       |
| Gluing distance $d$                        | 10         |
| Extended matching interval $\Delta_{glue}$ | 30 Hz      |

**Figure 24:** Comparison between measured engine speed (blue)  $f_{rot}$  and estimated engine speed (red)  $f_{rot,est}$ .

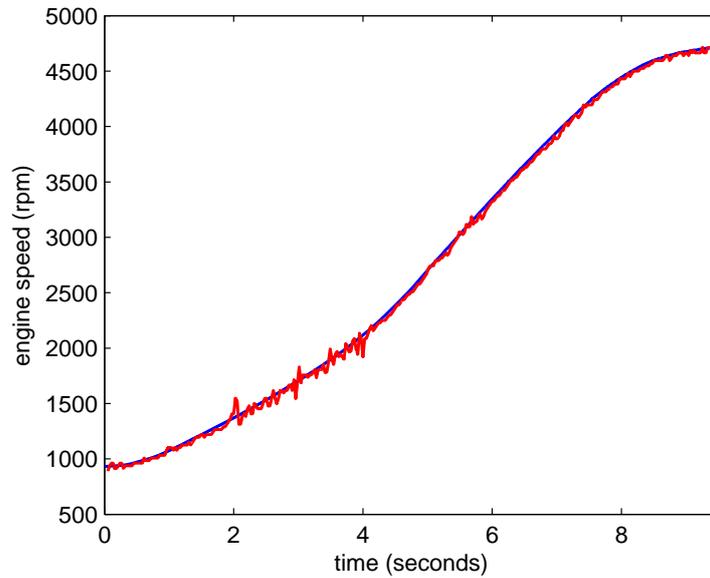
In the second example, the engine speed  $f_{rot,est}$  as a function of time of a 3-cylinder diesel engine of a compact car will be estimated. As expected, the corresponding spectrogram of the audio recording in figure 25 (A) shows a very prominent 1.5<sup>th</sup> engine order at frequencies 1.5 times as high as the fundamental frequency of the engine, but generally surrounded by a certain amount of noise. The chosen parameters for the applied STFT and for the subsequent frame-to-frame peak matching procedure are the same as in the first case example and listed in table 3.

Without additional post-processing, the basic pitch track provides engine speed information for 42 % of the recording. After applying the pitch track shifting procedure 3 times, the extended basic pitch track contains information for a remarkable 99 % of the recording. For the first shifting procedure, a pitch track with information about the 3<sup>rd</sup> engine order has been selected, while for the second procedure, information from the 6<sup>th</sup>

order of a pitch track has been used to extend the basic track. For the third extension, a track with information about the 4.5<sup>th</sup> order has been used. The final engine speed estimation  $f_{rot,est}$  shows higher deviations from the measured engine speed  $f_{rot}$ , as figure 26 points out. Nevertheless, the computed mean quadratic deviation  $d^2$  equals to 0.32 Hz or 19 rpm, which is considered as a good result.



**Figure 25:** (A): Spectrogram of an engine run-up of a 3-cylinder diesel engine vehicle and the corresponding measured engine speed (black solid line). (B): Pitch tracks (black lines) after applying the frame-to-frame peak matching procedure with the selected basic pitch track (red line) and three shifted pitch tracks (blue, green and orange line).

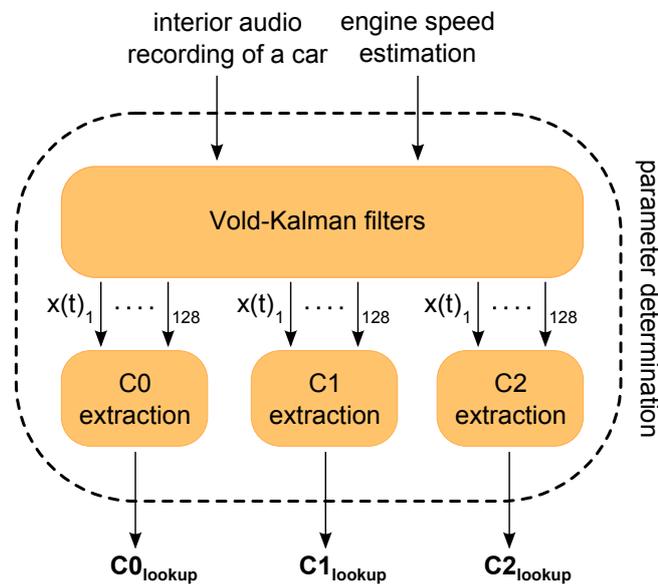


**Figure 26:** Comparison between measured engine speed (blue)  $f_{rot}$  and estimated engine speed (red)  $f_{rot,est}$ .

Besides these two case studies, a general validation of the accuracy of the engine speed estimation has been carried out. For this, the estimation algorithm without post-processing has been applied on all 98 interior audio recordings of the database. Whenever a suitable basic pitch track could be detected, the engine speed as a function of time has been estimated in the region of that pitch track and based on that pitch track. The calculated mean quadratic error  $d^2$  from that engine speed estimations never exceeded 3.4 Hz for any recording in the database. Furthermore, the average mean quadratic error  $d^2$  from these engine speed estimations equals to 0.46 Hz or 28 rpm.

## 3.2 Sound synthesis parameter extraction

The second and main part of the analysis method contains the actual determination of sound synthesis parameters for the engine sound synthesis model, which has been described in section 2. Specifically, the aim of this part of the analysis is the calculation of values for the lookup matrices  $C0_{\text{lookup}}$ ,  $C1_{\text{lookup}}$  and  $C2_{\text{lookup}}$  from the interior audio recording of a vehicle. Therefore, the torque state of the engine must be close to one of the three states, which have been described in table 1. As a second requirement, the engine speed as a function of time has to be available in order to apply a so-called Vold-Kalman filter for every single engine order. The output of these filters is a signal in the time domain, which will then be used to extract the elements for the lookup matrices.



**Figure 27:** Block diagram of the sound synthesis parameter determination procedure. The original interior audio recording as well as an estimation of the engine speed over time serve as input for the Vold-Kalman filters. Each of the 128 extracted engine order signals is then used to determine the actual parameters for the lookup matrices  $C0_{\text{lookup}}$ ,  $C1_{\text{lookup}}$  and  $C2_{\text{lookup}}$ .

### 3.2.1 Engine order extraction

According to the definition of the sound synthesis model, every single row of the lookup matrices  $C0_{\text{lookup}}$ ,  $C1_{\text{lookup}}$  and  $C2_{\text{lookup}}$  contains all necessary information to synthesize a single engine order for one of the three torque states over a specified engine speed range. In order to extract the information for an engine order, an approach that separates the engine orders by filtering has been chosen. The so-called Vold-Kalman filter was first introduced by H. Vold and Leuridan in 1993 [9] and is specialized in tracking non-stationary sinusoidal components in a noisy signal. Both generations of the filter as

described in [10] perform the filtration process by solving an equation system with the least squares technique. Based on a non real-time realization of the second generation by C. Feldbauer and R. Höldrich in [11], the signal to be analysed can be described as a sum of  $K$  sinusoids plus noise:

$$sig(n) = \sum_k x_k(n)e^{i\varphi_k(n)} + noise(n) \quad (26)$$

Each of the sinusoids consist of a complex time-dependent modulator  $x_k(n)$  which modulates the carrier  $e^{i\varphi_k(n)}$ . The phase parameter  $\varphi_k(n)$  is connected to the actual engine speed and leads to a real-valued modulator, if the sequence  $\varphi_k(n)$  is known. Since the analysis method only estimates the engine speed as described in section 3.1, the modulator  $x_k(n)$  becomes a complex function. With respect to an engine order  $k$ ,  $x_k(n)$  describes its peak amplitude while the exponential term  $e^{i\varphi_k(n)}$  controls its frequency over time.

By performing a Vold-Kalman filtering with  $K$  filters on a signal as described in equation 26, the time series  $x_1(n), x_2(n), \dots, x_K(n)$  are extracted. In combination with their known corresponding carriers  $e^{i\varphi_1(n)}, e^{i\varphi_2(n)}, \dots, e^{i\varphi_K(n)}$  they provide a temporal representation of  $K$  engine orders. For extracting the modulators, two equations are needed - the data equation and the structural equation.

If  $K$  engine orders shall be extracted simultaneously, the data equation equals to

$$sig(n) - \sum_k x_k(n)e^{i\varphi_k(n)} = e_d(n) \quad (27)$$

with  $e_d(n)$  as an error sequence, which should be minimal. The simultaneous extraction of the orders enables tracking of crossing engine orders, but leads to a very large system of equations. Since it is assumed that the frequencies of all orders of the engine are directly related to the actual engine speed and furthermore, none of the spectrograms of the recordings in the database revealed any signs of crossing orders, the engine orders can be extracted sequentially. This simplifies equation 27 to:

$$sig(n) - x_k(n)e^{i\varphi_k(n)} = e_{d,k}(n) \quad (28)$$

The structural equation performs a smoothing of the modulator  $x_k(n)$  with a low-pass filter expressed as a difference operator  $\nabla^{p+1}$  with the order  $p$ , and has the following form:

$$\nabla^{p+1}x_k(n) = e_{s,k}(n) \quad (29)$$

For a smooth modulator, the error sequence  $e_{s,k}(n)$  has to be minimized. By introducing the filter coefficients of  $\nabla^{p+1}$ , the structural equation becomes a simple difference equation. For a first order filter, it leads to:

$$x_k(n-1) - 2x_k(n) + x_k(n+1) = e_{s,k}(n) \quad (30)$$

Since the analysis method is not intended to be applicable in real-time, equation 30 uses a non-causal filter to avoid a phase bias. For combining the structural and the data equation, it is preferable to write the latter in vector notation:

$$\mathbf{sig} - \mathbf{C}_k \mathbf{x}_k = \mathbf{e}_{d,k} \quad (31)$$

The diagonal matrix  $\mathbf{C}_k$  contains the complex carrier sequence  $e^{i\varphi_k(n)}$  at the main diagonal and leads to an element-wise multiplication of the carrier with  $\mathbf{x}_k$ . For the structural equation, the matrix  $\mathbf{Str}_p$  is introduced with the coefficients of the low-pass filter and has the following appearance for  $p = 1$

$$\mathbf{Str}_p = \begin{bmatrix} -2 & 1 & 0 & \dots & 0 \\ 1 & -2 & 1 & & 0 \\ 0 & 1 & -2 & \ddots & 0 \\ \vdots & & \ddots & \ddots & 1 \\ 0 & & 0 & 1 & -2 \end{bmatrix} \quad (32)$$

in

$$\mathbf{Str}_p \mathbf{x}_k = \mathbf{e}_{s,k} \quad (33)$$

In addition, the error  $e_{s,k}$  is multiplied with the weighting factor  $R_k$ , which influences the bandwidth and the transition time of the filter. A large weighting factor yields to a very smooth modulator which is equivalent to a small bandwidth. Furthermore,  $R_k$  can be a scalar for a constant filter bandwidth or a diagonal matrix with a time-variant weighting factor for a time-dependent bandwidth. In order to obtain an equation system that contains the structural and the data equation, the error vectors  $\mathbf{e}_{s,k}$  and  $\mathbf{e}_{d,k}$  are merged to  $\mathbf{res} = [R_k \mathbf{e}_{s,k}^T, \mathbf{e}_{d,k}^T]^T$  and minimized according to:

$$\min_x \|\mathbf{res}\|^2 = \min_x \mathbf{res}^H \mathbf{res} \quad (34)$$

This minimization goal is equivalent to finding the optimal least squares solution of the following equation system

$$\mathbf{A} \mathbf{x}_k \approx \mathbf{b} \quad (35)$$

consisting of  $2N$  equations and  $N$  unknowns, where  $N$  is the total length of the signal  $\mathbf{sig}$ . Every unknown represents one sample of the modulator  $x$ . The Matrix  $\mathbf{A}$  combines the sparse band matrix  $\mathbf{Str}_p$  of equation 33 and the diagonal matrix  $\mathbf{C}_k$  of equation 31:

$$\mathbf{A} = \begin{bmatrix} R_k \mathbf{Str}_p \\ \mathbf{C}_k \end{bmatrix} \quad (36)$$

The right hand side vector  $\mathbf{b}$  of the equation system has the following form:

$$\mathbf{b} = \begin{bmatrix} \mathbf{0} \\ sig \end{bmatrix} \quad (37)$$

In order to solve the equation system for  $\mathbf{x}_k$ , the normal equation

$$\mathbf{A}^H \mathbf{A} \mathbf{x}_k = \mathbf{A}^H \mathbf{b} \quad (38)$$

is used and leads to

$$\mathbf{B}_k \mathbf{x}_k = \overline{\mathbf{C}}_k sig \quad (39)$$

where  $\mathbf{A}^H$  stands for the conjugate transposed of  $\mathbf{A}$  and  $\overline{\mathbf{C}}_k$  for the conjugated of the diagonal matrix  $\mathbf{C}_k$ . The matrix  $\mathbf{B}_k$  is given by:

$$\mathbf{B}_k = \mathbf{Str}_p^T R_k^2 \mathbf{Str}_p \mathbf{I} \quad (40)$$

For extracting  $K$  orders independently, the equation system in (39) has to be solved for  $k = 1, \dots, K$ . Since the length  $N$  of the signal is usually very big and the system is ill-conditioned, a proper approach is needed. In [11], the preconditioned conjugate gradient method has been presented and used to determine the modulation sequence  $\mathbf{x}_k$ , which belongs to the  $k^{\text{th}}$  engine order.

Since describing and implementing a suitable method for solving the equation system in (39) was beyond the scope of this work, an existing implementation of the Vold-Kalman filter in MATLAB by M. van der Seijs [12] has been used. The following description of the filter parameters is based on this implementation and thus also serves as a validation.

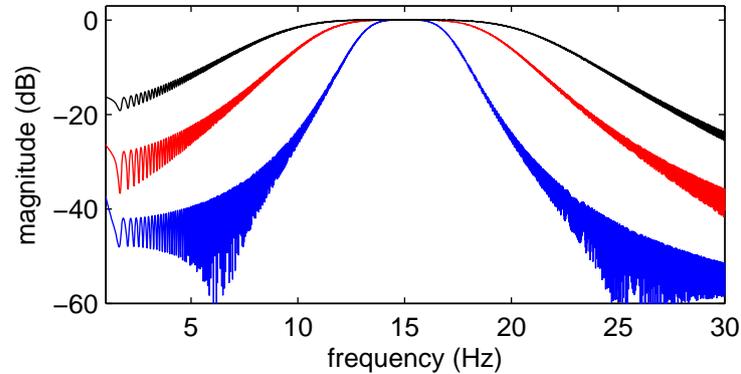
The appliance of the Vold-Kalman filter on a signal contains solving a system of linear equations. Two different parameters in this system control the -3dB-bandwidth  $\Delta f$  of the filter - the earlier mentioned weighting factor  $R$  and the filter order  $p$ . According to [11] the relationship between  $\Delta f$  and the weighting factor can be expressed as

$$\Delta f = 1.58 R^{-\frac{1}{2}} \frac{f_s}{2\pi}, \text{ for } p = 1 \quad (41)$$

$$\Delta f = 1.70 R^{-\frac{1}{3}} \frac{f_s}{2\pi}, \text{ for } p = 2 \quad (42)$$

where  $f_s$  represents the sampling frequency. In the chosen implementation,  $\Delta f$  can be set directly due to the internal adjustment of  $R$  with respect to the selected filter order. The measured frequency responses of a 1<sup>st</sup>-order filter in figure 28 shows that very small bandwidths are achievable with Vold-Kalman filters, even at very low frequencies. This ensures sufficient selectivity at small engine speeds where the distance between engine orders is very small. An extreme example would be the 1<sup>st</sup> order of a 2-cylinder engine

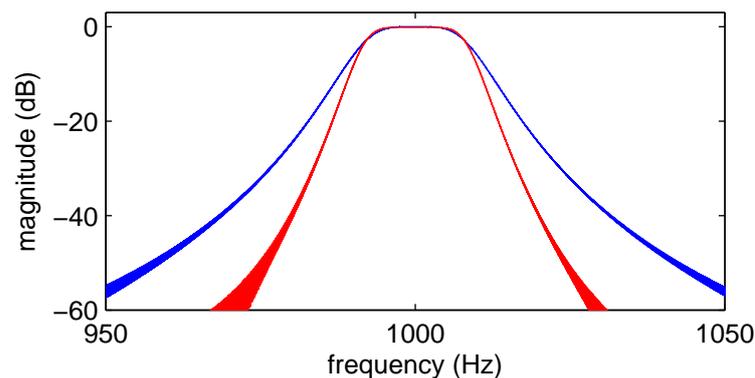
at 900 rpm, which has a spectral maximum at 15 Hz. The next maximum of the 1.5<sup>th</sup> order would be at 22.5 Hz, which corresponds to a distance of just 7.5 Hz.



**Figure 28:** Measured frequency response of a 1<sup>st</sup>-order Vold-Kalman filter with a bandwidth  $\Delta f = 4$  Hz (blue), 8 Hz (red) and 12 Hz (black), and a relative center frequency of 0.0016 in relation to the sampling frequency.

The bandwidth determination of the implementation has been tested from 2 Hz to 64 Hz with  $f_s = 9600$  Hz for 1<sup>st</sup> and 2<sup>nd</sup>-order filter and delivered satisfying results, wherever the system was not too ill-conditioned.

An increase of the filter order  $p$  results in a higher number of coefficients and furthermore in a wider sparse band of the matrix  $\mathbf{Str}_p$ . While the documentation of the filter implementation states, that the damping outside the -3dB-bandwidth is 40 dB/decade per filter order, the verification draws a different picture, as shown figure 29:



**Figure 29:** Measured frequency response of a 1<sup>st</sup> (blue) and 2<sup>nd</sup>-order (red) Vold-Kalman filter with a bandwidth  $\Delta f = 16$  Hz and a relative center frequency of 0.1042 in relation to the sampling frequency.

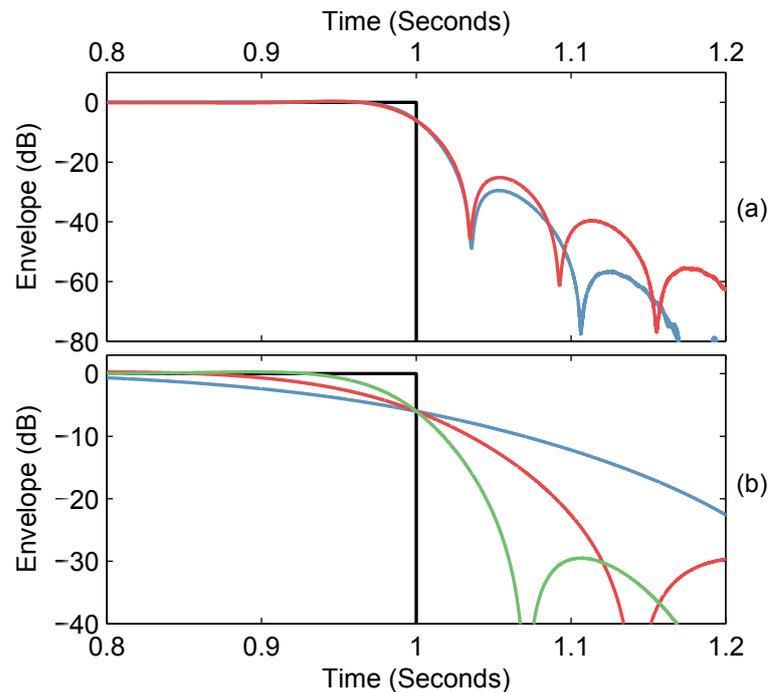
In general it can be said that describing the damping with a constant value per decade or per frequency interval is not possible for a Vold-Kalman filter. For determining the

influence of different parameters on the damping, a series of tests have been performed on the chosen implementation, which lead to the following outcomes:

- As expected, the filter order  $p$  has the most prominent influence and a higher  $p$  leads to a better damping. Furthermore, the ripple outside the pass-band increases with higher orders, as depicted in figure 29.
- A smaller -3dB-bandwidth  $\Delta f$  substantially increases the damping, as visualized in figure 28.

The earlier mentioned sampling frequency of  $f_s = 9600$  Hz has been chosen to reduce the computation effort for the filter with respect to the extraction of high engine orders. With an effective frequency range of 4800 Hz, even a 32<sup>nd</sup> order of a sport engine at 8000 rpm would be covered and could therefore be extracted.

Besides the described frequency characteristics, the temporal behaviour of the Vold-Kalman filter implementation has been validated. A tone burst with a duration of 1 second at a fixed frequency of 100 Hz has been used to measure a filter with a constant center frequency under different bandwidth and filter order settings. The results are presented as the temporal envelope of the burst and depicted in figure 30:



**Figure 30:** Temporal behaviour of a Vold-Kalman filter, measured with a tonal burst. (a) shows the temporal envelope of the burst after applying a 1<sup>st</sup>-order (blue) and a 2<sup>nd</sup>-order (red) filter with a constant center frequency of 100 Hz and  $\Delta f = 16$  Hz. In (b), the envelope of the same burst after applying the same 1<sup>st</sup>-order filter is shown, but with a bandwidth  $\Delta f$  of 2 (blue), 4 (red) and 8 Hz (green). Both sub-figures also include the ideal envelope of the burst as reference (black).

Based on the measurements carried out, it can be said that a higher filter order as well as a smaller bandwidth leads to a worse temporal behaviour. This makes it harder for the filter to track fast amplitude changes of the engine order. The overall results of the temporal analysis of the filter implementation coincide very well with the results in [13], where the following simplified inverse proportional relationship has been established for the upper decay of 25 dB, expressed as the time for the tone burst to decay 8.69 dB:

$$T_{8.69dB} = \frac{0.2}{\Delta f} \quad (43)$$

Since the Vold-Kalman filter implementation from [12] was evaluated successfully, it has been used to individually extract 128 signals, which serve as a basis for the calculation of sound synthesis parameters for the 128 oscillators. The oscillator with the number  $i$  is used to synthesize the engine order  $k$  according to the relation already described in section 2.2:

$$k = 0.25i \quad (44)$$

For a proper extraction of  $K$  engine orders,  $K$  Vold-Kalman filters are needed. Furthermore, for the traction of an order  $k$  by the filter  $k$  the frequency over time of the order  $f_{o,k}(n)$  is required. An estimation of this frequency over time can be derived via the estimated engine speed  $f_{rot,est}(n)$  and the following relationship:

$$f_{o,k}(n) = k \frac{f_{rot,est}}{60}, \text{ with } k = 0.25, 0.5, \dots, 32 \quad (45)$$

The frequency  $f_{o,k}$  as a function of time is then used as the time-varying center frequency  $f_{c,k}(n)$  for the  $k^{\text{th}}$  filter. In addition to the center frequency, the bandwidth  $\Delta f$  and the order of the filters had to be chosen. For this, several characteristics of an engine sound have been considered:

- In general, the spectral bandwidth of one isolated engine order can be assumed as very narrow, but not as a single peak at only one frequency. Even if an engine could produce an order containing only one single frequency, the natural fluctuations from an adjusted constant engine speed would broaden the spectrum.
- Depending on the instantaneous value of  $f_{rot}$ , the distance between the spectral maximum of two engine orders in hertz can be very small. For an engine at 900 rpm at the time  $n = n_1$ , the maximum of the order  $k = 1$  would have a center frequency  $f_{o,1}(n = n_1) = 15$  Hz, whereas the next neighbouring engine orders maximum would equal to  $f_{o,1.25}(n = n_1) = 18.75$  Hz.

The filter order as well as the bandwidth of the Vold-Kalman filter determine its behaviour in the frequency domain in terms of selectivity and damping. In order to minimize the overlapping between two filters at very low frequencies when  $K$  engine orders shall be extracted from the recording, a high filter order  $p$  combined with a small bandwidth  $\Delta f$  would be preferable. However, in practice this requirement cannot be fulfilled, since

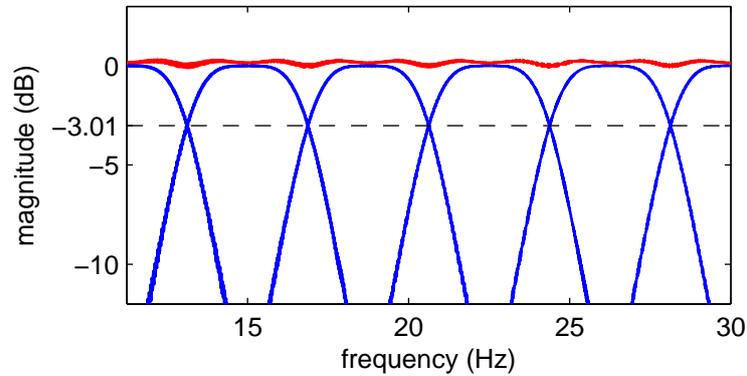
adjusting both parameters in this way leads to a too ill-conditioned equation system and results in a drastic deviation of the actual filter characteristics from the desired ones.

In addition, a lower bandwidth as well as a higher filter order worsens the temporal behaviour and thus the filter's ability to track temporal changes of the spectral magnitude without a substantial delay.

With this considerations in mind, an engine-speed-dependent bandwidth has been chosen, which can be determined with:

$$\Delta f(f_{rot,est}) = \begin{cases} \frac{f_{rot,est}}{60} \frac{1}{4} & \text{for } \frac{f_{rot,est}}{60} \frac{1}{4} < 4Hz \\ 4Hz & \text{for } \frac{f_{rot,est}}{60} \frac{1}{4} \geq 4Hz \end{cases} \quad (46)$$

In general, 4 Hz are considered as a suitable bandwidth to cover the narrow spectral range of an engine order. However, if the distance between the  $f_{o,k}(n)$  and  $f_{o,k+0.25}(n)$  reaches a point, where the upper -3dB-cutoff frequency of the filter  $k$  would be higher than the lower -3dB-cutoff frequency of the filter  $k + 0.25$ , a lower bandwidth  $\Delta f$  is chosen so that these two cutoff frequencies are at the same point. From the perspective of the frequency domain for a filter order  $p = 1$ , this ensures that the total energy of all extracted engine orders does not exceed the energy of the original recording. The resulting transfer function for this case is depicted in figure 31. A small rippling in the overall frequency response is unavoidable due to the shape of a 1<sup>st</sup>-order filter. It should also be noted that this frequency response is only valid for a stationary signal.



**Figure 31:** Individual frequency responses of multiple 1<sup>st</sup>-order Vold-Kalman filters with a -3dB-bandwidth of 3.75 Hz (blue) and overall frequency response of the filters (red). The higher -3dB-cut-off frequency of the filter  $k$  equals to the lower -3dB-cut-off frequency of the filter  $k + 0.25$ .

Possible filter order candidates result from the chosen bandwidth range in relation to the sampling frequency and the condition of the equation system to solve. For a bandwidth  $\Delta f = 4$  Hz down to 1 Hz and  $f_s = 9600$  Hz, the highest achievable filter order with the chosen filter implementation equals to  $p = 1$ , which has been finally selected.

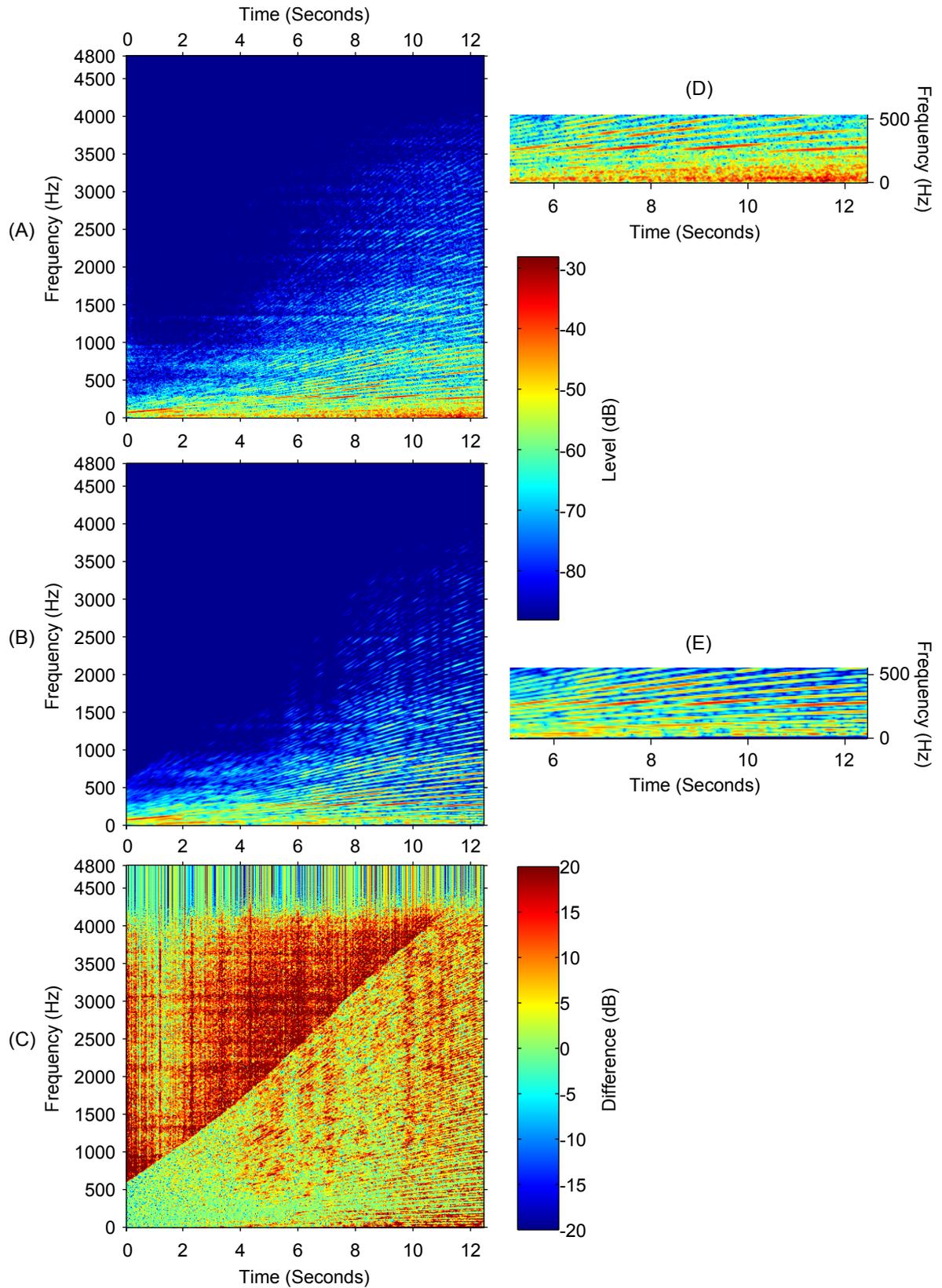
With a chosen bandwidth  $\Delta f$  and a filter order  $p$  for the Vold-Kalman filters, 128 engine orders have been extracted from all recordings of the database. An exemplary

extraction of an 8-cylinder engine is depicted in figure 32 and shows the spectrogram of the original interior audio recording in (A). Sub-figure (B) represents a superposition of the spectral magnitudes of all extracted signals with the Vold-Kalman filters. Based on equation 46, the bandwidth for this example is time-dependent and is not greater than  $\Delta f = 4$  Hz. In sub-figure (C), the calculated spectral magnitude difference between the original recording and the overall extracted signal has been visualized. For this example, the following observations can be made:

- The overall appearance of the spectrogram of the extracted engine orders in sub-figure (B) looks similar to the original spectrogram in sub-figure (A). Even frequency-dependent resonances at approximately 250 Hz are preserved, as depicted in (D).
- The magnified section of the spectrogram of the original recording in (D) shows a substantial amount of low-frequency noise, which partly masks orders below  $k = 2$ . An origin of this noise can be the rumble of a chassis dynamometer or rumble from driving on a street.
- Due to the small bandwidth of the Vold-Kalman filters, noisy components between single engine orders are damped. This effect can be seen in sub-figure (C) or by comparing sub-figure (D) with sub-figure (E).
- The big spectral difference above the 32<sup>nd</sup> engine order in (C) seems to be illogical in the first place since a comparison of (A) with (B) shows no noticeable difference. While every power spectral density value 60 dB below the maximum of the original recording has been set to 60 dB below the maximum (corresponds to -88 dB for this example), the spectrogram in (C) has been computed with the original values.
- A negative difference in (C) indicates that more energy has been extracted than theoretically available in the original recording. Although the overall frequency response of the filters has been tailored to extract not more energy than the original signal contains, the temporal behaviour of the Vold-Kalman filter leads to these negative differences.

Since Vold-Kalman filtering is used on real interior recordings with an unknown progression of the engine orders as well as with an unknown degree of influence of the disturbances like rumble noise, no exact validation of the Vold-Kalman filtering is possible. However, all tested recordings of the database with interior vehicle recordings showed a difference spectrogram close to the one depicted in figure 32.

In order to determine the final sound synthesis parameters  $C_0$ ,  $C_1$  and  $C_2$  for the lookup matrices  $\mathbf{C}_{0\text{lookup}}$ ,  $\mathbf{C}_{1\text{lookup}}$  and  $\mathbf{C}_{2\text{lookup}}$ , a different post-processing is applied for every parameter on every extracted engine order.



**Figure 32:** Spectrogram of the original interior recording of a car with an 8-cylinder engine (A) and a magnified section of it (D), superposition of spectral magnitudes of all 128 extracted engine orders (B) and a magnified section of it (E), spectral magnitude differences (C) between (A) and (B).

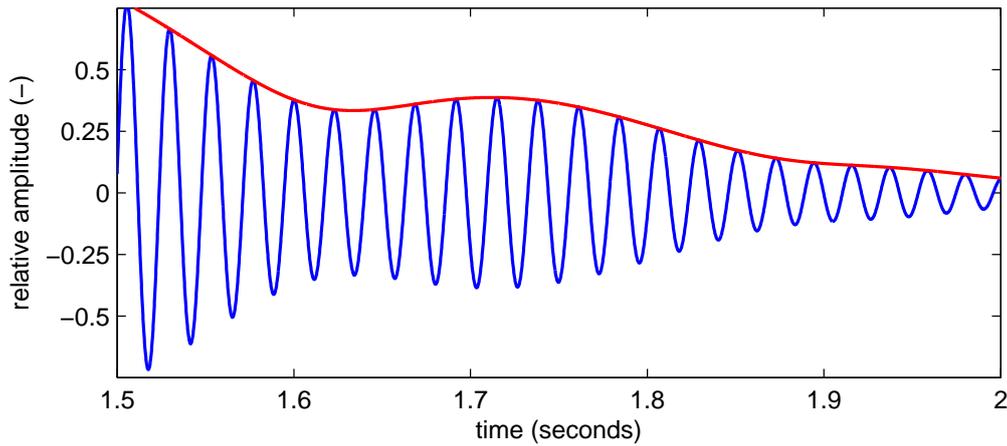
### 3.2.2 Calculation of $C0$

Based on the extracted time-domain signal of an engine order from a Vold-Kalman filter, the sound synthesis parameter  $C0$  will be determined. The parameter  $C0$  controls the peak amplitude of an oscillator without considering a possible amplitude modulation, as described in equation 5. By taking the actual engine speed into account,  $C0$  can be interpreted as the actual intensity of an engine order, synthesized by one of the oscillators of the engine sound model.

As first step for obtaining values for  $C0$ , the envelope  $\mathcal{E}$  of the time-domain signal  $x_k(n)$  of the  $k^{\text{th}}$  extracted engine order is estimated by using the Hilbert transform according to the following equation:

$$\mathcal{E}\{x_k(n)\} = \sqrt{x(n)^2 + \mathcal{H}\{x(n)\}^2} \quad (47)$$

A temporal excerpt of an envelope computed from the 2<sup>nd</sup> order of a 4-cylinder diesel engine can be seen in figure 33.



**Figure 33:** Extracted temporal representation of a 2<sup>nd</sup> order of a 4-cylinder diesel engine (blue) and its corresponding envelope (red), computed with the Hilbert transform.

Since in the sound synthesis model the whole covered engine speed range from  $f_{rot} = 0$  to  $f_{rot} = \max(f_{rot,est})$  is quantized to 128 groups, an actual range  $r$  depending on the analysed engine recording of the group  $l$  has to be calculated with

$$r = \frac{\max(f_{rot,est})}{128 - 1} \quad (48)$$

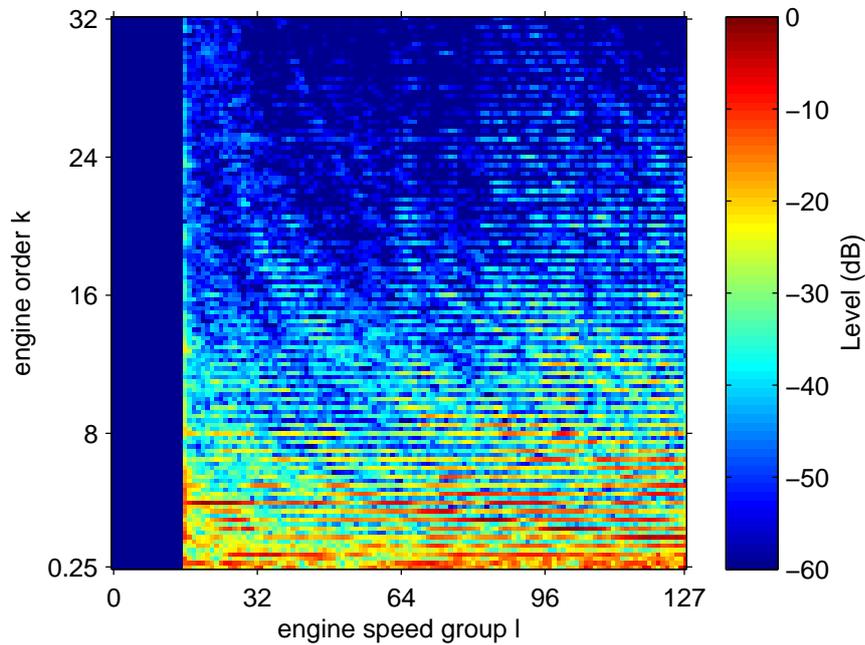
Then the estimated engine speed  $f_{rot,est}(n)$  is used to assign every sample of  $\mathcal{E}\{x_k\}$  to a group  $l$  by the following rule:

$$l(\mathcal{E}\{x_k(n)\}) = \text{round}\left(\frac{f_{rot,est}(n)}{r}\right) \quad (49)$$

In order to determine a value for  $C0$  for the group  $l$ , an average engine speed value is calculated by taking the median,

$$C0_l = \text{median}\left(\mathcal{E}\{x_k(n)\}_{l=0\dots 127}\right) \quad (50)$$

which is less prone to single outliers than the mean value. Finally, the resulting 128 values for  $C0$  of every oscillator can be concentrated to the lookup matrix  $\mathbf{C0}_{\text{lookup}}$ . This matrix can be visualized with different colours for different  $C0$ . Figure 34 shows the computed lookup matrix  $\mathbf{C0}_{\text{lookup}}$  from a run-up of an 8-cylinder engine:



**Figure 34:** Graphic representation of the computed lookup matrix  $\mathbf{C0}_{\text{lookup}}$  from an 8-cylinder engine. All values have been normalized to the element with the highest value.

Several frequency-dependent resonances, as already observed in the spectrogram in figure 32 in a horizontal form, are now visible in form of one-sided paraboles. The vertical blue bar at very low values for  $l$  represents an area, where no envelope samples were available to compute a  $C0$ . Furthermore, about half of the recordings of the database showed unexpected values in the first column right of the blue bar, as in figure 34. The origin of these occurrences could not be determined, but it is believed that the damping of a 1<sup>st</sup>-order Vold-Kalman filter is too low to eliminate the influence of neighbouring engine orders.

The evaluation of the results for  $\mathbf{C0}_{\text{lookup}}$  is part of section 3.2.5, where a complete synthesized engine sound is compared against the original interior recording.

### 3.2.3 Calculation of C1

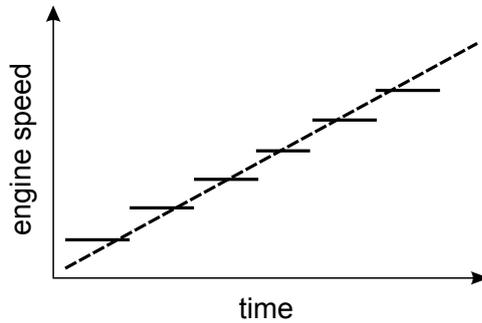
Based on the extracted time-domain signal of an engine order from a Vold-Kalman filter, the sound synthesis parameter  $C1$  will be determined. The parameter  $C1$  controls the intensity of the side-band frequencies of an oscillator, as depicted in figure 3.

In order to compute the amplitude modulation depth  $m = C1$  from an extracted engine order  $k$ , the envelope  $\mathcal{E}\{x_k(n)\}$  is used within the following equation:

$$m = C1 = \frac{\mathcal{E}\{x_k(n)\}_{max} - \mathcal{E}\{x_k(n)\}_{min}}{\mathcal{E}\{x_k(n)\}_{max} + \mathcal{E}\{x_k(n)\}_{min}} \quad (51)$$

As a prerequisite, the exponential carrier  $e^{i\varphi_k(n)}$  of the extracted order has to be at one constant frequency for calculating the modulation depth without disturbing influences. This can only be achieved with a constant increasing phase sequence  $\varphi_k(n)$ . In practice, this would require an engine recording with short time periods of constant engine speed, analogue to a stepped-sine sweep signal.

Since the engine recording database includes only engine run-ups with a constantly changing engine speed, no meaningful calculation of  $C1$  was possible. In figure 35, a typical engine speed progression of a recording from the database is visualized in comparison to an actual required progression:



**Figure 35:** Schematic engine speed progression of a typical audio recording in the database (dashed line) and required engine speed progression (solid lines) in order to calculate a meaningful  $C1$ .

### 3.2.4 Calculation of C2

By the time the author of this report had started to work on a suitable method for calculating the parameter  $C2$ , the sound synthesis model had received its first revision. One new aspect of the updated model was an uncontrollable phase deviation  $\Delta\phi$ , which makes the synthesis parameter  $C2$  unnecessary. Therefore, no method for calculating  $C2$  has been developed.

### 3.2.5 Case studies

In this section, the results for the sound synthesis parameter extraction of  $C0$  will be presented for two exemplary audio recordings of a vehicle interior during an engine run-up. The resulting lookup table  $\mathbf{CO}_{\text{lookup}}$  for the run-up has been used as an engine sound profile for the software Engine Sound Tool to resynthesize the engine sound. In order to compare the original and the synthesized engine sound with each other, the measured engine speed from the recording has been used as control parameter for the synthesis model in the software. A detailed description of the software can be found in section 4. Furthermore, it must be mentioned that the synthesis model has been developed at the Institute of Electronic Music and Acoustics - therefore shaping the synthesized engine sound to the original sound was only possible by calculating the parameters in a different way, but not by changing the synthesis model, since it was already predetermined.

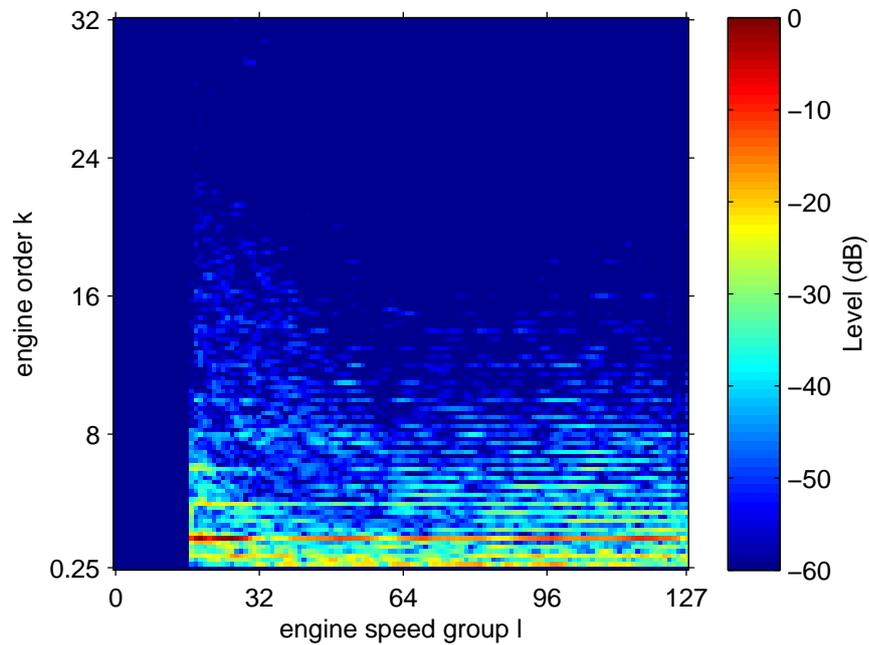
In the first example, the same recording of a 4-cylinder gasoline engine of a compact car as in section 3.1.4 will be analysed. Table 4 shows the settings used for the Vold-Kalman filtering, which have been derived and described in section 3.2.1.

**Table 4:** Chosen Vold-Kalman filter parameters

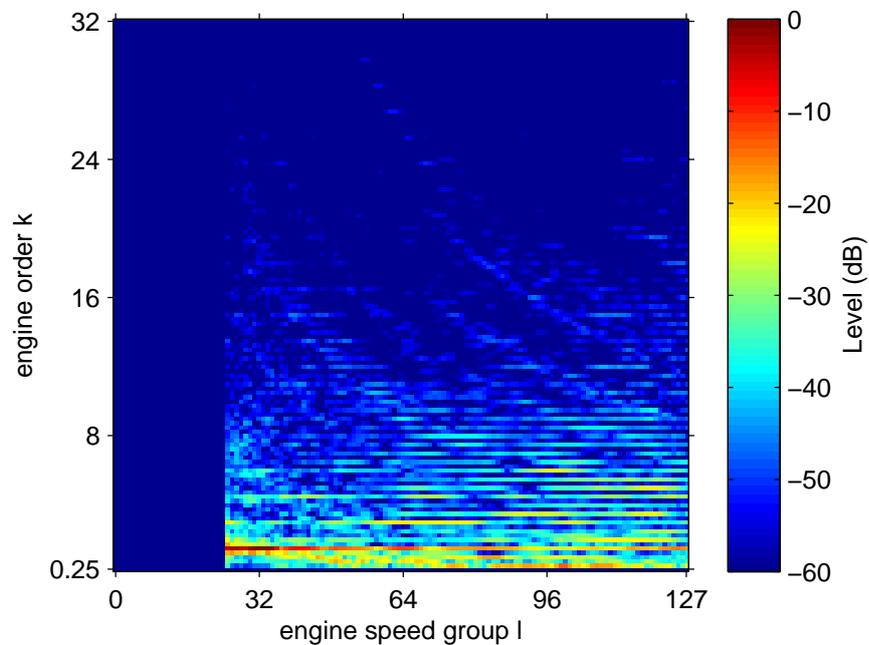
| Parameter                                      | Value   |
|------------------------------------------------|---------|
| filter order $p$                               | 1       |
| maximum<br>-3dB-bandwidth $\Delta f$           | 4 Hz    |
| sampling frequency $f_s$<br>of audio recording | 9600 Hz |

The -3dB-bandwidth of 4 Hz represents only the maximum possible bandwidth - the actual engine-speed-dependent bandwidth  $\Delta f$  has been determined according to equation 46. As control parameter for the sound synthesis model, the measured engine speed as depicted in figure 24 has been used.

As expected, the computed and visualized lookup table  $\mathbf{CO}_{\text{lookup}}$  in figure 36 shows very high values for the engine order  $k = 2$ , which is also clearly visible in the spectrogram of the original recording in figure 38 (A). Since the original recording contains rumble noise at very low frequencies, all values for the oscillator with the engine order  $k = 0.25$  have been set to zero. The resulting effect is visible in the spectrogram of the synthesized engine sound at low frequencies in the region of high engine speeds in figure 38 (B). Besides this intended difference, the synthesized spectrogram shows a very similar shape as the spectrogram of the original recording in figure 38 (A). Nevertheless, one unintended effect results from the high engine order density of  $k = 0.25i$ . The extracted signal of certain "quarter engine order" Vold-Kalman filters contains mostly noise, since the engine sound does not contain any quarter engine orders in these frequency regions. This extracted noise is unintentionally used to compute a  $C0$ , which serves as an input parameter of an oscillator, which outputs a periodic signal.

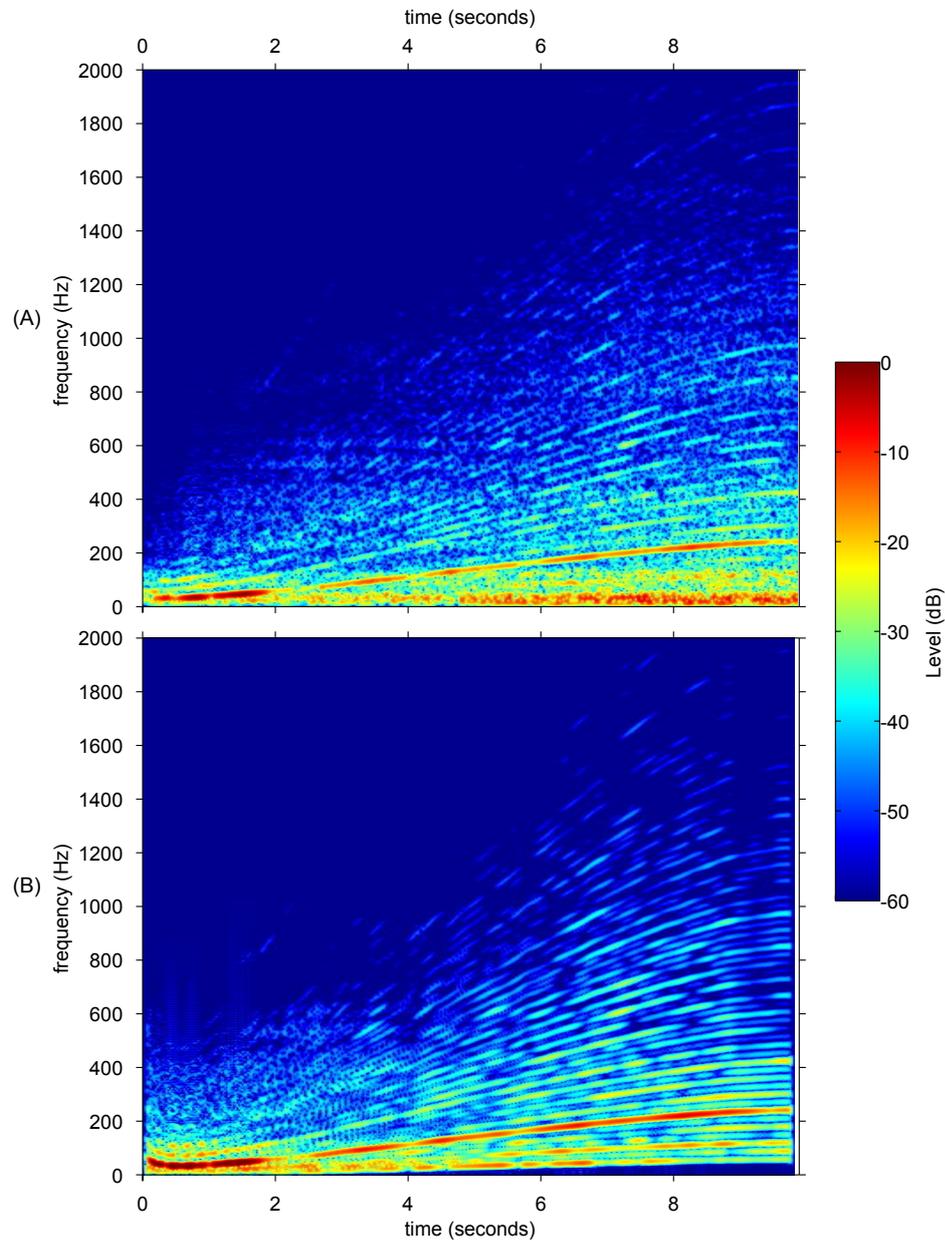


**Figure 36:** Graphic representation of the computed lookup matrix  $\mathbf{CO}_{\text{lookup}}$  from an engine run-up of a 4-cylinder gasoline engine vehicle. All values have been normalized to the element with the highest value.



**Figure 37:** Graphic representation of the computed lookup matrix  $\mathbf{CO}_{\text{lookup}}$  from a 3-cylinder diesel engine run-up of a vehicle. All values have been normalized to the element with the highest value.

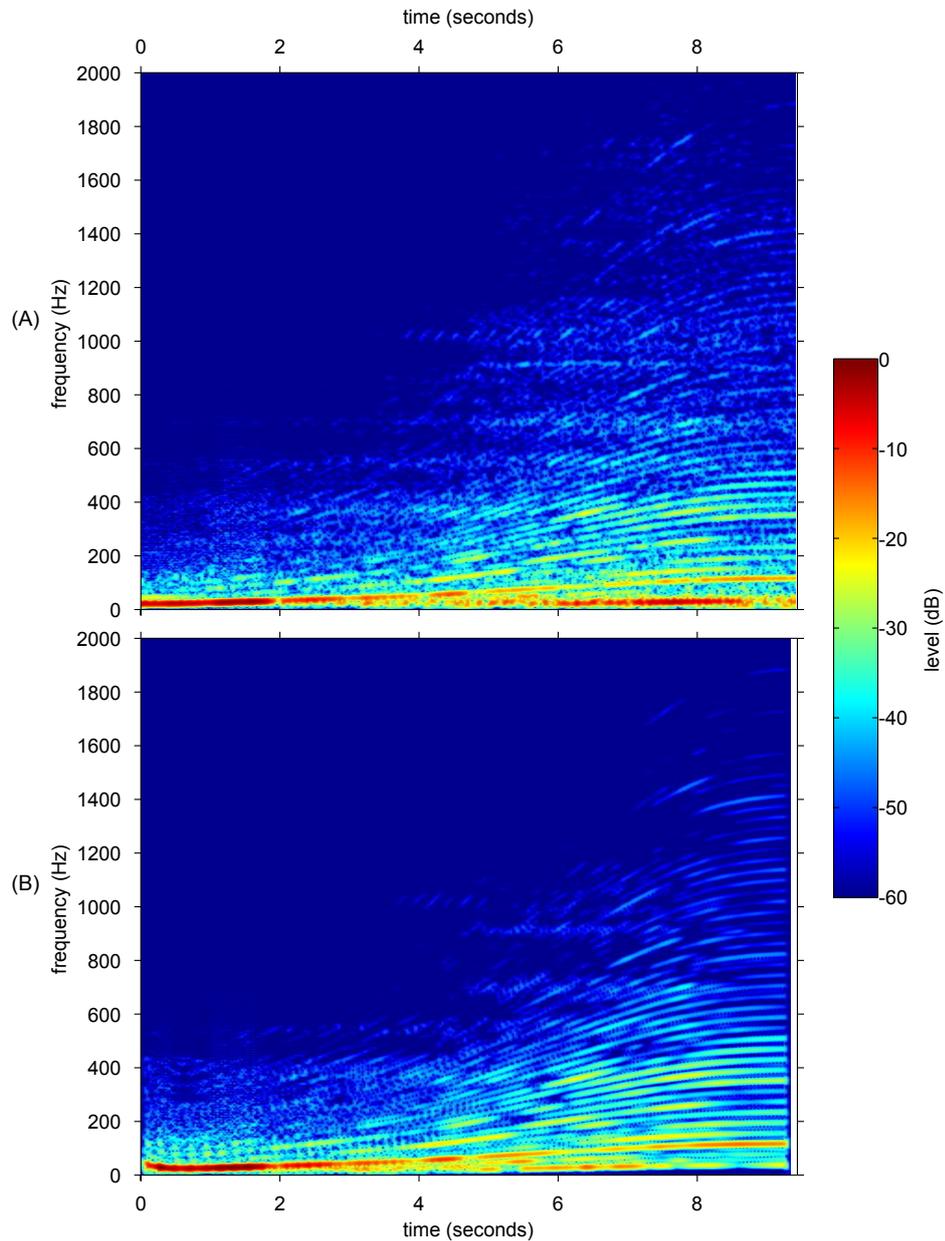
A comparison between the spectrogram of the original and the synthesized engine sound in figure 38 clearly shows certain engine orders in the synthesized sound, which originate from noise of the original recording.



**Figure 38:** (A): Spectrogram of an engine run-up of a 4-cylinder gasoline engine vehicle. (B): Synthesized engine sound based on the computed lookup table  $C0_{lookup}$  from the original audio recording of the run-up.

As a solution for this problem, an extension of the analysis method which checks the content of every extracted signal by the Vold-Kalman filters for noise is proposed. If the amount of noise exceeds a certain level, no sound synthesis parameters will be extracted from this signal.

For the second example, in which the same recording of a 3-cylinder diesel engine of a compact car as in section 3.1.4 will be analysed, the same Vold-Kalman filter settings as in the first example have been chosen. Compared to the lookup table in the first example, the lookup table for  $C0$  in this example in figure 37 shows several distinctive frequency-dependent resonances, which appear as one-sided parabolic rises in the plot.



**Figure 39:** (A): Spectrogram of an engine run-up of a 3-cylinder diesel engine vehicle. (B): Synthesized engine sound based on the computed lookup table  $C0_{lookup}$  from the original audio recording of the run-up.

## 4 Engine sound design software

Besides the development of an analysis method for the extraction of sound synthesis parameters, a graphical user interface for an existing implementation of the engine sound synthesis model has been developed within this project. The user interface enables an easy engine sound design process by controlling the synthesis model, which had already been implemented in the graphical programming language Pure Data. The former MATLAB-based graphical user interface was written as a stand-alone executable, which required a special MATLAB compiler runtime (MCR) in order to run the software on the target machine. Since applications developed and executed in this way are not famous for their performance and require a lot of hard disk space<sup>3</sup>, the programming language C# has been chosen for writing a new front-end with a graphical user interface for the synthesis model. As part of the .NET Framework, C# also offers the possibility of using the Windows Presentation Foundation (WPF) framework to design graphical user interfaces in the declarative language Extensible Application Markup Language (XAML). The visual elements of the interface have been written in XAML and are supported by the underlying control logic, which is implemented in C#. A brief comparison of the different aspects of the used .NET Framework and MATLAB can be found in table 5:

**Table 5:** Comparison between the .NET Framework and MATLAB

| Aspect                       | .NET Framework                       | MATLAB                                                             |
|------------------------------|--------------------------------------|--------------------------------------------------------------------|
| Kind of environment          | Software framework                   | Environment for numerical computing, visualization and programming |
| Main programming language(s) | C#, VB.NET, J#, XAML                 | MATLAB language                                                    |
| Kind of language(s)          | compiled to CIL code at compile time | interpreted at runtime                                             |

Within this project, C# and XAML have been used to develop a graphical user interface, which serves as a front-end to control the sound synthesis model. The communication between the user interface and the model in Pure Data has been realized via the TCP protocol. The next sections give an overview about the Engine Sound Tool v.0.1.8.9 and describe the most important functions of this software.

### 4.1 Overview

After the start-up of the Engine Sound Tool, the main window of software is displayed as in figure 40. At this point, the synthesis model is already active in the background

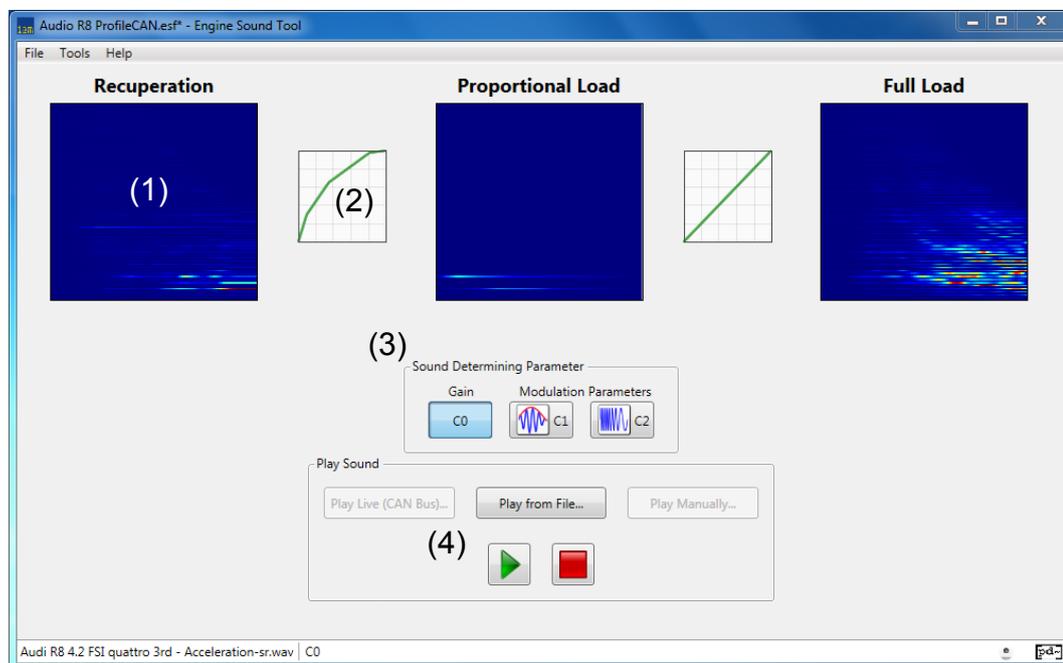
<sup>3</sup> a MATLAB compiler runtime 7.17 32-bit for example requires more than 800 megabytes of hard disk space

and has been reset to its default values via a TCP connection between the user interface and Pure Data.

The main window gives an overview about an entire engine sound profile, which includes all 9 lookup matrices and the 6 weighting vectors as described in section 2 and depicted in figure 8. While the lookup tables are visualized as coloured plots as in figure 34, the weighting vectors are displayed as a 2-dimensional plot, where the horizontal axis represents the different elements of the vector and the vertical axis its values. However, all visualizations in this window are only intended to give an overview of the current engine sound profile - the actual manipulation of the lookup tables and weighting vectors happens in different windows of the software.

Besides giving an overview about the actual loaded engine sound profile, the main window of the software offers several ways for the playback of an engine sound:

1. via live CAN bus data, which include the actual torque and engine speed values to choose the proper control parameters  $C0$ ,  $C1$  and  $C2$  for the oscillators (not fully implemented in this version).
2. via a special ASCII-encoded file, which contains time series of engine speed and torque data. In this playback mode, previously recorded engine speed and torque data can be used to synthesize an engine sound.
3. via a manual graphical adjustment of the engine speed and the torque (not fully implemented in this version).

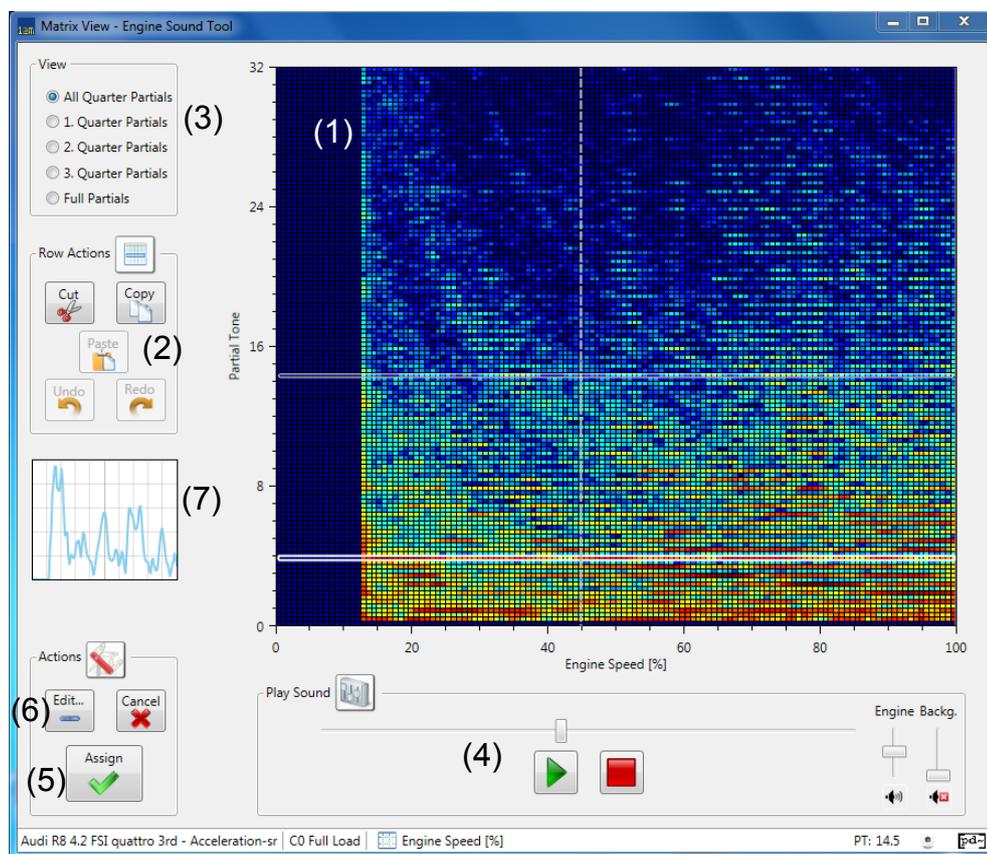


**Figure 40:** Main window of the software Engine Sound Tool - visualizations of the lookup tables (1), the weighting vectors (2) of the current engine sound profile. (3) enables switching between the different synthesis parameters. (4) offers multiple ways for playback of the engine sound by controlling the engine speed and the torque.

The actual editing of an engine sound (profile) happens in the matrix editor, which can be reached by clicking on one of the lookup tables.

## 4.2 Matrix editor

The matrix editor is used for global editing of one of the nine lookup tables. Each row, which contains a certain synthesis parameter over the whole engine speed range for one oscillator, can be cut, copied and inserted at a different position of the table by selecting the row in the matrix plot of the window. These actions can also be applied to multiple rows and work in the same way as for the software Windows Explorer of the operating system Windows 7.



**Figure 41:** Matrix editor of the software Engine Sound Tool - One or multiple rows can be selected in the matrix plot (1) and copied or cut with the buttons in (2). The radio buttons in (3) allow the user to display only a subset of the lookup table in the matrix plot. Changes in the lookup table can be verified by ear via the player (4) before they are finally applied with the assign button (5). If single values in the row of the lookup table have to be altered, the edit button (6) is used to edit the selected row in the engine order editor. Furthermore, a 2-dimensional representation of the selected row is displayed in (7).

Sometimes it is necessary to edit only a subset of rows of the table, for example the parameters of the 5<sup>th</sup>, 9<sup>th</sup> and 13<sup>th</sup> oscillator, which correspond to the engine orders 1.25, 2.25 and 3.25. Therefore, the group box in the upper left corner of the window offers different views to display for example only the 1<sup>st</sup>, 5<sup>th</sup>, 9<sup>th</sup> and 13<sup>th</sup> row of the lookup table for an easier editing process.

In order to verify changes made within the matrix editor by ear, the player below the matrix plot can be used to play the engine sound including edits, which have not been saved yet. With the horizontal slider, the sound of the engine can be explored in real-time for different engine speeds. If the user is satisfied with the current content of the lookup table, the assign-button can be clicked to save all changes - the matrix editor will then be closed and the main window becomes active again.

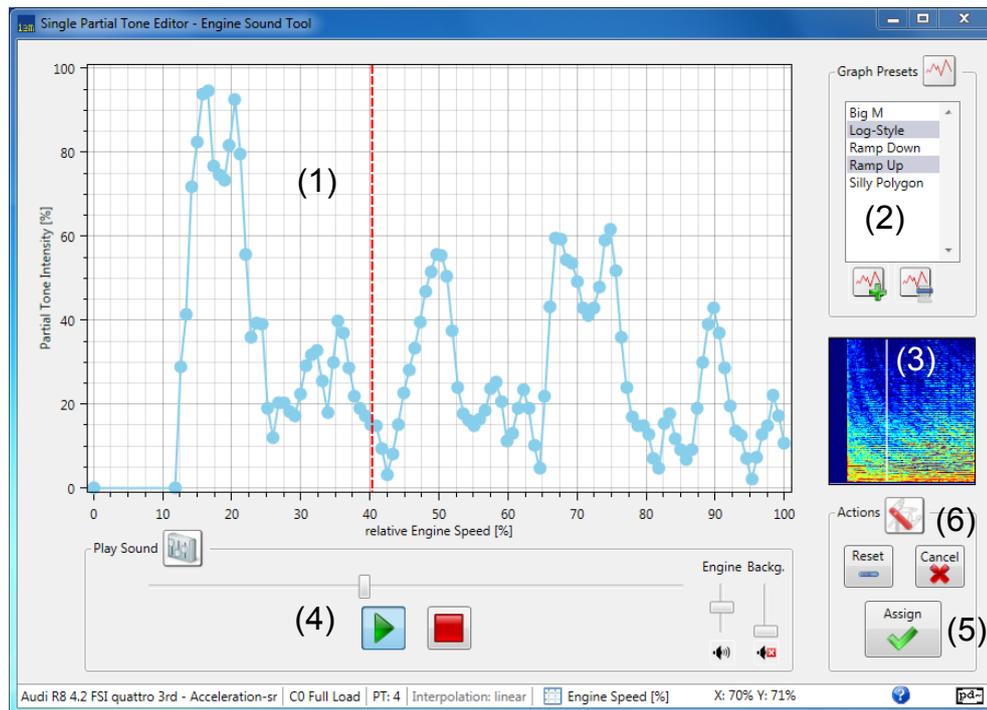
If detailed changes for a single engine order are necessary, the user has to select the lookup table row corresponding to this order. The small preview plot on the left side of the window presents the values of the synthesis parameter as a 2-dimensional function. A click on the edit-button opens the engine order editor for the selected lookup table row.

### **4.3 Engine order editor**

The engine order editor enables the user to modify one of the sound synthesis parameters for a single oscillator, whose behaviour is described by the corresponding row of the lookup table. The coloured representation of the selected row is then converted to a 2-dimensional function and displayed in the center of the engine order editor, as depicted in figure 42. In addition, a node is placed at each position, where the function exceeds a certain gradient threshold. Besides altering the position of these nodes with the mouse, new nodes can be created by simply clicking into the plot where the position of the new node shall be. Unnecessary nodes can be removed with a left-click on the nodes. If the shape of the function is satisfying and shall be remembered by the software, the current graph can be saved as a preset in a list, which is located in the right half of the window.

Every change made in the selected row of the lookup table will be displayed immediately in the small lookup table preview plot next to main graph in the window. Furthermore, the vertical dashed line marks the actual selected synthesis parameters from the lookup table for a specific engine speed, which has been set in the player below the main graph. This player serves the same purpose as the player in the matrix editor window.

If the user is satisfied with the current values of the synthesis parameter for the oscillator, the assign-button can be clicked to save all changes - the engine order editor will then be closed and the matrix editor window becomes active again.



**Figure 42:** Engine order editor of the software Engine Sound Tool - the selected row of the matrix editor is visualized as a 2-dimensional function in (1) and can be altered by simple adding or moving existing nodes. Alternatively, a previously saved function from the list in (2) can be selected to replace the current function in (1). Every change made for the oscillator is immediately visible in the lookup table preview plot in (3). For evaluating the current edits within the context of the whole engine sound, a player with controllable engine speed is available (4). With the buttons (5) and (6), the applied changes can be saved to the lookup matrix or just discarded.

## 5 Summary and outlook

Within this project work, an analysis method for the extraction of sound synthesis parameters from an interior audio recording of a motor vehicle has been developed. These synthesis parameters are used for an engine sound synthesis model, which was developed at the Institute of Electronic Music and Acoustics (IEM). In order to enable an easy engine sound design process, a modern graphical user interface has been programmed with the .NET Framework to alter extracted sound synthesis parameters or to create an entire new engine sound.

For the extraction of the sound synthesis parameters, a 2-step analysis method has been designed and implemented in MATLAB. In the first analysis step, a pitch detection and tracking algorithm is applied on the audio recording in order to estimate the engine speed as a function of time. Based on the frame-to-frame peak matching method, which was originally presented as part of a speech analysis-synthesis technique, an advanced engine speed estimation algorithm has been developed, which operates in the frequency domain. The designed algorithm has been applied on a database with 98 interior audio recordings of engine run-ups and showed satisfying results both in terms of accuracy and reliability. For 96 % of all recordings, a suitable pitch track could be found, which serves as a basis for the engine speed estimation. A comparison between the measured engine speed and the estimated engine speed in the region of the pitch track revealed that the average mean quadratic deviation equals to 0.46 Hz or 28 revolutions per minute for the recordings in which a suitable pitch track could be found. After applying the pitch track gluing post-processing method once, an engine speed estimation which covers at least 90 % of the recording could be obtained for 20 % of the recordings of the database. If the pitch track shifting method was applied 3 times, an engine speed estimation which covers at least 90 % of the recording could be obtained for 29 % of the recordings. Preliminary tests with a combined adaptive approach of the pitch track gluing and the pitch track shifting post-processing method showed that the number of engine speed estimations with a coverage of a least 90 % can be further increased. Therefore, a next step for improving the engine speed estimation algorithm would include the designing of an iterative algorithm, which selects the most suitable post-processing method to extend the pitch track until it reaches a certain minimum length - for example to a length, where the pitch track covers at least 90 % of the recording.

In the second step of the analysis method, the actual extraction of the sound synthesis parameters is carried out. 128 Vold-Kalman filters are sequentially applied on the audio recording to extract time-domain signals. Each of them contains information about a single order of the engine sound. For a proper traction of the engine orders over time, the estimated engine speed is used as information for the Vold-Kalman filters. Afterwards, the sound synthesis parameters for each oscillator of the engine sound synthesis model are determined from the corresponding time-domain signal. During the development process, it was found that the recording of an usual engine run-up is not suited for calculating a meaningful amplitude modulation depth and the phase deviation. Both parameters have been defined in the synthesis model in order to create a natural variation in the engine sound and for controlling its roughness. The determination of the

peak amplitude for the oscillators was carried out successfully for all recordings in the database. A spectral comparison between the original and synthesized engine sounds showed only little differences and pointed out that the overall analysis-synthesis concept works satisfactorily. However, one deviation from the original engine sound was the presence of quarter engine orders, even if the original engine sound did not contain any quarter engine orders. This could be traced back to the quarter engine order resolution of the synthesis model, which is obviously not necessary for every type of engine. Therefore, a verification procedure is proposed, which estimates the amount of noise in every extracted time-domain signal in order to exclude signals, which do not contain any engine order information. This would avoid the accidental determination of synthesis parameters for an oscillator, which would then synthesize an engine order, which is not part of the original engine sound.

In the second part of the project work, a front-end with a graphical user interface for the synthesis model has been written in the programming languages C# and XAML. The overall software called Engine Sound Tool enables a user to import an engine sound profile based on the extracted sound synthesis parameters of a recorded engine sound. Several graphic editors offer the possibility of editing an engine sound profile and listen to changes in real-time. Since the analysis method for the extraction of the sound synthesis parameters has been implemented in MATLAB, a step for improving the software would be a direct integration of the analysis method in the software.

## References

- [1] Various authors "*User manual for the IEM Engine Sound Tool V.1.17*" 2012: Institute of Electronic Music and Acoustics.
- [2] Vibrate Software Inc. "*Engine speed related vibrations*" Retrieved July 26 2014, from [http://www.vibratesoftware.com/html\\_help2014/2011/Diagnosis/Engine\\_Speed\\_Related.htm](http://www.vibratesoftware.com/html_help2014/2011/Diagnosis/Engine_Speed_Related.htm).
- [3] K. Kasi, S. Zahorian "*Yet another algorithm for pitch tracking*" 2002: Proc. of Acoustics, Speech, and Signal Processing of the IEEE International Conference
- [4] de Cheveigné, A., and Kawahara, H. "*YIN, a fundamental frequency estimator for speech and music*" 2002: Journal of the Acoustical Society of America
- [5] A. von dem Knesebeck, U. Zölzer "*Comparison of pitch trackers for real-time guitar effects*" 2010: Proc. of the 13<sup>th</sup> Int. Conference on Digital Audio Effects
- [6] U. Zölzer "*DAFX - Digital Audio Effects*" 2002: John Wiley & Sons
- [7] P. Boersma "*Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound*" 1993: Proc. of the Institute of Phonetic Sciences, University of Amsterdam
- [8] R. McAulay, T. Quartieri "*Speech analysis/synthesis based on a sinusoidal representation*" 1986: IEEE Transactions on Acoustics, Speech, and Signal Processing, pp. 744-754
- [9] H. Vold, J. Leuridan "*High Resolution Order Tracking at Extreme Slew Rates Using Kalman Tracking Filters*" 1993: SAE Paper No. 931288
- [10] J. Tuma "*Setting the Passband Width in the Vold-Kalman Order Tracking Filter*" 2005: Proceedings of the International Congress on Sound and Vibration, Lisbon, Portugal.
- [11] C. Feldbauer, R. Höldrich "*Realization of a Vold-Kalman Tracking Filter - a Least Squares Problem*" 2000: Proceedings of the COST G-6 Conference on Digital Audio Effects, Verona, Italy
- [12] M. van der Seijs "*Second generation Vold-Kalman Order Filtering*" 2012: Matlab Implementation, Retrieved August 30 2014, from <http://www.mathworks.com/matlabcentral/fileexchange/36277-second-generation-vold-kalman-order-filtering>.
- [13] H. Herlufsen, S. Gade "*Characteristics of the Vold-Kalman Order Tracking Filter*": Proceedings of the Society for Experimental Mechanics

## List of Figures

- 1 Block diagram of the analysis-synthesis system for the engine sound model. The analysis part consists of an engine speed estimation from an interior audio recording of a vehicle, which serves as input for the sound synthesis parameter extraction together with the engine speed estimation. The determined sound synthesis parameters are then sent into the synthesis model, which (re)creates an actual engine sound. . . . . 5
- 2 Spectrogram of an engine run-up of a sports car at the driver's seat. The rising full engine orders (1<sup>st</sup> to 6<sup>th</sup> are marked with black solid lines) as well as the half engine orders (2<sup>nd</sup> to 6<sup>th</sup> are marked with black dashed lines) are clearly distinguishable. . . . . 7
- 3 Temporal representation (left) and magnitude spectrum (right) of a sinusoidal carrier amplitude modulated by a sinusoidal signal.  $m$  represents the ratio between the peak amplitude of the modulator and the carrier ( $\frac{\hat{x}_{AM}}{\hat{x}_C}$ ). . . . . 8
- 4 Oscillator of a single engine order - the synthesis parameter  $C0$  represents the peak amplitude of the carrier,  $C1$  the amplitude modulation depth and  $C2$  stands for the phase deviation. The angular frequency  $\omega_C$  is controlled by the engine speed  $f_{rot}$ . . . . . 8
- 5 Synthesis parameter matrices  $\mathbf{C0}_{lookup}$ ,  $\mathbf{C1}_{lookup}$  and  $\mathbf{C2}_{lookup}$  - the row index  $i$  corresponds to the engine order with the number  $k$  equal to  $0.25i$ . The column index indicates the synthesis parameter at a specific engine speed bin. . . . . 9
- 6 Separate set of 3 matrices for the torque states "full load", "proportional load" and "full drag". . . . . 10
- 7 Shape of a weighting factor vector between two torque states . . . . . 11
- 8 Separate set of 3 matrices (violet) for the torque states "full load", "proportional load" and "full drag" as well as the 6 different weighting vectors (green) between two torque states (index  $fp$  - "full load" and "proportional load", index  $pf$  - "proportional load" and "full drag"). . . . . 11
- 9 Visualization of the sound synthesis model, which consists of 9 lookup matrices (violet) and 6 weighting vectors (green). The engine speed and the torque serve as control parameters and determine the exact selection of the actual synthesis parameters from the lookup matrices. The chosen parameters  $C0$ ,  $C1$  and  $C2$  as well as the engine speed are used as input for a single oscillator which synthesizes one engine order. The final engine sound is obtained by superimposing the output of all 128 oscillators. . . . 12

|    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |    |
|----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 10 | Block diagram of the analysis procedure. The interior audio recording is used to estimate the engine speed as a function of time. The result of this estimation and the audio recording itself serve as input for the sound synthesis parameter extraction. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 13 |
| 11 | Temporal representation (a) of a signal consisting of a sinusoidal with the frequency $\frac{f_s}{64}$ and a sinusoidal with $f = \frac{f_s}{16}$ . The corresponding ACF (b) shows the obvious maximum at a lag $l = 0$ (red circle) and the 2 <sup>nd</sup> maximum (green circle) at $l = t_0 = 64 > l_{min}$ . . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 15 |
| 12 | Long-term prediction with a 1 <sup>st</sup> -order FIR filter. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 16 |
| 13 | Temporal representation (a) of a signal consisting of a sinusoid with the frequency $\frac{f_s}{64}$ , a sinusoid with $f = \frac{f_s}{32}$ and a sinusoid with $f = \frac{f_s}{16}$ . The corresponding magnitude spectrum (b) shows three peaks, and the peak with the lowest frequency (red) is considered to be the fundamental frequency. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 17 |
| 14 | Block diagram of the engine speed estimation procedure, not including post-processing. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 19 |
| 15 | Spectral peak picking applied on a magnitude spectrum. 4 out of 10 peaks (circles) are discarded (blue circles), since they lie below the minimum peak height (-44 dB, horizontal dashed line). No lower peaks are present within the minimum peak distances ( $\pm 6$ Hz, vertical dashed lines) of the peaks (red circles), therefore no additional peak will be discarded. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                     | 20 |
| 16 | Different scenarios for frame-to-frame peak matching. If the distance between the pitch track with the frequency $f_n^j$ and $f_{m-1}^{j+1}$ is smaller than any other possible distance and lies within the matching interval (outlined by dashed lines), the pitch track is extended by $f_{m-1}^{j+1}$ (a). The pitch track is declared as "dead", if there are no frequencies within the matching interval (b). If a frequency is favoured by two existing pitch tracks, the one with the smallest matching distance will be extended. The remaining pitch track is declared as "dead", if no other frequencies are within the matching interval (c). The last scenario (d) shows the "birth" of a new pitch track, if a detected frequency has not been occupied by an existing pitch track. . . . . | 21 |
| 17 | Graphic pitch track representation (black lines) after applying the frame-to-frame peak matching procedure on the STFT of an engine run-up - number of peaks = 20. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 22 |
| 18 | Pitch tracks after applying the frame-to-frame peak matching procedure with 10 peaks (a), 20 peak (b) and 30 peaks (c) on the STFT of an engine run-up. A higher number of peaks leads to more pitch tracks with information about the engine speed as well as to more short pitch tracks uncorrelated to the engine speed. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 23 |

|    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |    |
|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 19 | Close look at pitch tracks (black lines with a red circle for every peak) after applying the frame-to-frame peak matching procedure with 20 peaks and a matching interval of 6 Hz (left) and 3 Hz (right). At 3 Hz, several pitch tracks were not detected as one continuous pitch track. . . . .                                                                                                                                                                                                                             | 23 |
| 20 | (a) Pitch tracks of a vehicle with a 4-cylinder gasoline engine after applying all 3 basic pitch track selections methods (method a) - red, method b) - blue, method c) - green) and true engine speed (orange) in revolutions per second. While methods b) and c) select a pitch track with information about the desired 2 <sup>nd</sup> engine order, method a) selects a long pitch track belonging to a higher order.<br>(b) Pitch tracks of a vehicle with a 3-cylinder diesel engine after applying method c). . . . . | 25 |
| 21 | Pitch track (black line), which has been interrupted by a peak gap with a length of 54 ms. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                            | 26 |
| 22 | Selected basic pitch track (black solid line), which has to be extended. While pitch track gluing from the left would require multiple attempts with only very short pitch tracks (black dashed lines), a very long pitch track (black dashed line above the basic pitch track) provides much more information at twice the frequency compared to the basic pitch track. . .                                                                                                                                                  | 27 |
| 23 | (A): Spectrogram of an engine run-up of a 4-cylinder gasoline engine vehicle and the corresponding measured engine speed (black solid line). (B): Pitch tracks (black lines) after applying the frame-to-frame peak matching procedure with the selected basic pitch track (red line) and the chosen pitch track candidate (blue line) for pitch track gluing. . . . .                                                                                                                                                        | 29 |
| 24 | Comparison between measured engine speed (blue) $f_{rot}$ and estimated engine speed (red) $f_{rot,est}$ . . . . .                                                                                                                                                                                                                                                                                                                                                                                                            | 30 |
| 25 | (A): Spectrogram of an engine run-up of a 3-cylinder diesel engine vehicle and the corresponding measured engine speed (black solid line). (B): Pitch tracks (black lines) after applying the frame-to-frame peak matching procedure with the selected basic pitch track (red line) and three shifted pitch tracks (blue, green and orange line). . . . .                                                                                                                                                                     | 31 |
| 26 | Comparison between measured engine speed (blue) $f_{rot}$ and estimated engine speed (red) $f_{rot,est}$ . . . . .                                                                                                                                                                                                                                                                                                                                                                                                            | 32 |
| 27 | Block diagram of the sound synthesis parameter determination procedure. The original interior audio recording as well as an estimation of the engine speed over time serve as input for the Vold-Kalman filters. Each of the 128 extracted engine order signals is then used to determine the actual parameters for the lookup matrices $\mathbf{C0}_{lookup}$ , $\mathbf{C1}_{lookup}$ and $\mathbf{C2}_{lookup}$ . . .                                                                                                      | 33 |
| 28 | Measured frequency response of a 1 <sup>st</sup> -order Vold-Kalman filter with a bandwidth $\Delta f = 4$ Hz (blue), 8 Hz (red) and 12 Hz (black), and a relative center frequency of 0.0016 in relation to the sampling frequency. . . . .                                                                                                                                                                                                                                                                                  | 37 |

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| <i>M. Czuka: Extraction of sound synthesis parameters from driving noise</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 61 |
| 29 Measured frequency response of a 1 <sup>st</sup> (blue) and 2 <sup>nd</sup> -order (red) Vold-Kalman filter with a bandwidth $\Delta f = 16$ Hz and a relative center frequency of 0.1042 in relation to the sampling frequency. . . . .                                                                                                                                                                                                                                                                                                                    | 37 |
| 30 Temporal behaviour of a Vold-Kalman filter, measured with a tonal burst. (a) shows the temporal envelope of the burst after applying a 1 <sup>st</sup> -order (blue) and a 2 <sup>nd</sup> -order (red) filter with a constant center frequency of 100 Hz and $\Delta f = 16$ Hz. In (b), the envelope of the same burst after applying the same 1 <sup>st</sup> -order filter is shown, but with a bandwidth $\Delta f$ of 2 (blue), 4 (red) and 8 Hz (green). Both sub-figures also include the ideal envelope of the burst as reference (black). . . . . | 38 |
| 31 Individual frequency responses of multiple 1 <sup>st</sup> -order Vold-Kalman filters with a -3dB-bandwidth of 3.75 Hz (blue) and overall frequency response of the filters (red). The higher -3dB-cut-off frequency of the filter $k$ equals to the lower -3dB-cut-off frequency of the filter $k + 0.25$ . . . . .                                                                                                                                                                                                                                        | 40 |
| 32 Spectrogram of the original interior recording of a car with an 8-cylinder engine (A) and a magnified section of it (D), superposition of spectral magnitudes of all 128 extracted engine orders (B) and a magnified section of it (E), spectral magnitude differences (C) between (A) and (B). . . . .                                                                                                                                                                                                                                                     | 42 |
| 33 Extracted temporal representation of a 2 <sup>nd</sup> order of a 4-cylinder diesel engine (blue) and its corresponding envelope (red), computed with the Hilbert transform. . . . .                                                                                                                                                                                                                                                                                                                                                                        | 43 |
| 34 Graphic representation of the computed lookup matrix $\mathbf{CO}_{\text{lookup}}$ from an 8-cylinder engine. All values have been normalized to the element with the highest value. . . . .                                                                                                                                                                                                                                                                                                                                                                | 44 |
| 35 Schematic engine speed progression of a typical audio recording in the database (dashed line) and required engine speed progression (solid lines) in order to calculate a meaningful $C1$ . . . . .                                                                                                                                                                                                                                                                                                                                                         | 45 |
| 36 Graphic representation of the computed lookup matrix $\mathbf{CO}_{\text{lookup}}$ from an engine run-up of a 4-cylinder gasoline engine vehicle. All values have been normalized to the element with the highest value. . . . .                                                                                                                                                                                                                                                                                                                            | 47 |
| 37 Graphic representation of the computed lookup matrix $\mathbf{CO}_{\text{lookup}}$ from a 3-cylinder diesel engine run-up of a vehicle. All values have been normalized to the element with the highest value. . . . .                                                                                                                                                                                                                                                                                                                                      | 47 |
| 38 (A): Spectrogram of an engine run-up of a 4-cylinder gasoline engine vehicle. (B): Synthesized engine sound based on the computed lookup table $\mathbf{CO}_{\text{lookup}}$ from the original audio recording of the run-up. . . . .                                                                                                                                                                                                                                                                                                                       | 48 |
| 39 (A): Spectrogram of an engine run-up of a 3-cylinder diesel engine vehicle. (B): Synthesized engine sound based on the computed lookup table $\mathbf{CO}_{\text{lookup}}$ from the original audio recording of the run-up. . . . .                                                                                                                                                                                                                                                                                                                         | 49 |

|    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |    |
|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 40 | Main window of the software Engine Sound Tool - visualizations of the lookup tables (1), the weighting vectors (2) of the current engine sound profile. (3) enables switching between the different synthesis parameters. (4) offers multiple ways for playback of the engine sound by controlling the engine speed and the torque. . . . .                                                                                                                                                                                                                                                                                                                                                       | 51 |
| 41 | Matrix editor of the software Engine Sound Tool - One or multiple rows can be selected in the matrix plot (1) and copied or cut with the buttons in (2). The radio buttons in (3) allow the user to display only a subset of the lookup table in the matrix plot. Changes in the lookup table can be verified by ear via the player (4) before they are finally applied with the assign button (5). If single values in the row of the lookup table have to be altered, the edit button (6) is used to edit the selected row in the engine order editor. Furthermore, a 2-dimensional representation of the selected row is displayed in (7). . . . .                                             | 52 |
| 42 | Engine order editor of the software Engine Sound Tool - the selected row of the matrix editor is visualized as a 2-dimensional function in (1) and can be altered by simple adding or moving existing nodes. Alternatively, a previously saved function from the list in (2) can be selected to replace the current function in (1). Every change made for the oscillator is immediately visible in the lookup table preview plot in (3). For evaluating the current edits within the context of the whole engine sound, a player with controllable engine speed is available (4). With the buttons (5) and (6), the applied changes can be saved to the lookup matrix or just discarded. . . . . | 54 |

## List of Tables

|   |                                                            |    |
|---|------------------------------------------------------------|----|
| 1 | Definition of the three torque states . . . . .            | 10 |
| 2 | Peak picking parameters and their description . . . . .    | 20 |
| 3 | Chosen engine speed estimation parameters . . . . .        | 30 |
| 4 | Chosen Vold-Kalman filter parameters . . . . .             | 46 |
| 5 | Comparison between the .NET Framework and MATLAB . . . . . | 50 |