NVH Active Sound Feature Specification

Interior Propulsion Sound Enhancement

Authors:

C. Störig – cstoerig@Ford.com

S. Ackers – sackers1@Ford.com

This document does apply to the Phoenix In-Vehicle Infotainment-Platform Project only.

Version 1.3 (DRAFT)

UNCONTROLLED COPY IF PRINTED

Version Date: May 3rd, 2021

Ford CONFIDENTIAL

1 Revision History

Date	Version	Notes
May 1st, 2021	1.2	Initial release; content/requirements merged based on ESE- and EVSE-specification documents, requirements updated to meet current design assumptions
May 3rd, 2021	1.3	Minor revisions due to proof reading, reference specification documents added

Content

1 Purpose of this document

This document describes the requirements of the interior Propulsion Sound (PS) for the Ford Phoenix infotainment system platform. The content of this document is focused on NVH-specific requirements.

In the following sections, the technical capabilities and technical features of the PS-feature are described. The included PS-synthesis algorithm is required in order to modify and optimize the powertrain sound inside of the vehicle cabin by utilizing the in-vehicle infotainment system and audio system infrastructure (AHU – Audio Head Unit). The PS-sound synthesis algorithm is required in order to deploy Ford's Global Powertrain Sound Quality DNA and associated NVH-interior sound quality targets.

The requirements within this document do apply to the PS-feature only.

2 References

#	Reference Document	Originator
1	'Active Noise Control for Vehicle Interior Application for ICE-Powertrains (ANC)', Filename: Phoenix_IVI_ANC_NVH_Specification_Draft_Version_1.4.pdf, Draft Version 1.4	C. Störig, S. Ackers - Ford NVH
2	'Versatile Binary Format Specification 3.1', Filename: VBF-00.06.15.004-008.pdf, Version: 8	B. Waldeck - Ford EESE
3	'Phoenix Active Noise Control/ Propulsion Sound Enhancement Functional Specification', Filename: Phoenix ANC-PS Functional Specification.docx, Draft Version 1.0	J. Hartman - Ford EESE

3 Introduction and Common Propulsion NVH-related Definitions

Within this document, the terms ‘Active Noise Cancellation’ resp. ‘Active Noise Control’ (‘ANC’) refer to the system's capability to reduce components of the powertrain’s interior sound at arbitrary signal frequencies.

The term ‘Electronic Sound Enhancement’ (‘ESE’) refers to the system's capability to increase the powertrains original operating sound at arbitrary sound frequencies. The terms ‘Electronic Sound Enhancement” and “Active Sound Design” can be used as synonyms. In the following, the term ‘Electronic Sound Enhancement’, ‘Engine Sound Enhancement’ or the abbreviation ‘ESE’ will be used.

The term ‘Electric Vehicle Sound Enhancement’ (‘EVSE’) refers to the system’s capability to add more complex non-combustive sound elements for enhancement of hybrid and fully electric vehicle propulsion sound.

The term ‘Propulsion Sound’ (‘PS’) refers to the total collective propulsion sound enhancement feature set of ESE (combustion related sound focus) and EVSE (non-combustive related sound focus).

The so-called ‘*engine order(s)*’ and ‘*engine harmonic(s)*’ are related to the engine crank shaft rotational frequency and are a narrow-band spectral component of the (quasi-periodic) engine sound signal. The *engine order–spectrum* is created by the (quasi-)periodic firing events of the cylinders. Equation 1 shows the relationship between engine order index and the instantaneous rotational frequency [Hz] resp. number of rotations per minute [rpm, min^{-1}].

The control frequency of the PS synthesis algorithm must be synchronized to the engine speed (instantaneous rotational frequency of the engine’s crank shaft – tacho-input signal) or other input control parameters resp. sensor signals, e.g. vehicle speed [kph].

The (graphical) relation between engine order frequency and engine speed resp. the engine’s rotational frequency is shown in Figure 8 (function plot – half order spacing).

Equation (*5*) shows the mathematical relation between the crankshaft rotational frequency, engine order frequency and engine order index.

2 Terms and Descriptions

Term	Description
ASF	Active Sound Feature
ANC	Active Noise Cancellation, Active Noise Control
ESE	Engine Sound Enhancement
EVSE/EV-ESE	Electric Vehicle (Electronic) Sound Enhancement
PS	Propulsion Sound
RNC	Road Noise Cancellation
HIL	Hardware-in-the-loop

3 Requirements

3.1 Common Ford DNA Powertrain Sound Quality Requirements

The PS-feature must allow arbitrary modifications of the interior powertrain sound character by adding additional sound signal components by utilizing the in-vehicle audio system (audio head unit (AHU)).

Here, the original ICE or EV powertrain sound (ICE = Internal Combustion Engine, EV = Electric Vehicle) is the *primary sound* contributor while the PS-feature generates the *secondary sound* signals.

The ESE component of the PS-feature must provide the capability of a *plausible* sound presentation with a *natural* sound character. The generated ESE-sound – as a secondary contribution to the *primary* powertrain sound – should be perceived as a component of the original vehicle powertrain sound.

The ESE component must provide suitable signal processing methods and sound synthesis parameters in order to deliver the functional basis for plausible results (e.g. *narrow-band engine order synthesis algorithm*).

The EVSE component must provide suitable signal processing and sound synthesis parameters to deliver a variety of complex synthesis contributions (e.g. sample-based, granular, wavetable).

The PS synthesis algorithm must be real-time capable and interactive, so that it is able to follow the instantaneous operations of the vehicle powertrain system in response to the operations of the driver, e.g. accelerating and de-accelerating.

The PS synthesis algorithm must react without noticeable delays, drop-outs and/or artifacts according to the dynamic changes of the vehicle PT-system and therefore of the related sensor signals (e.g. engine speed information via the vehicle's CAN-Bus-Network).

The existing infrastructure (AHU) of the vehicle must be used in order to generate the PS signals inside of the vehicle cabin, e.g. by utilizing the DSP-infrastructure of the Audio Head Unit.

The PS synthesis algorithm must be integrated into the software and hardware-architecture of the head unit.

The PS synthesis algorithm must be controlled by input parameters delivered via the vehicle's CAN-Bus network (physical sensor data signals).

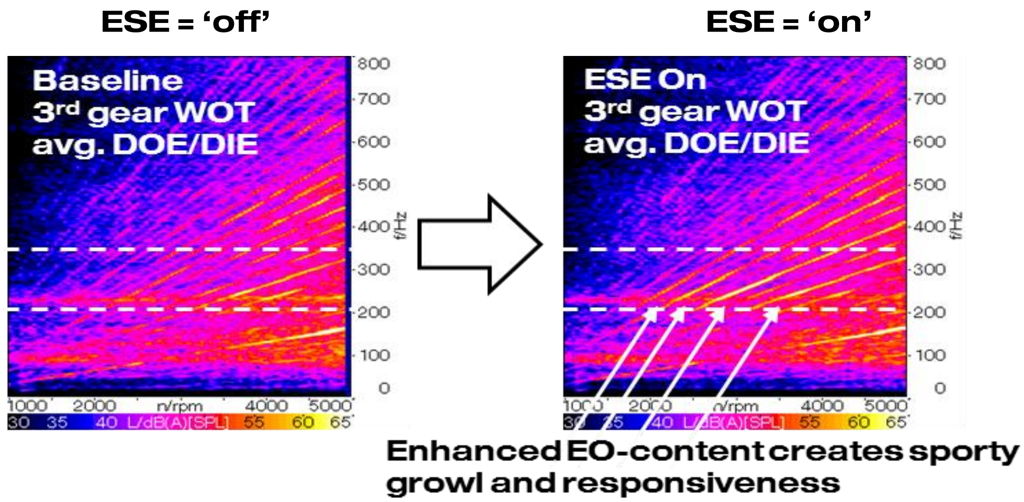


Figure 1: Vehicle interior noise measurement data: Short-time Fourier transform vs. engine speed of a run-up maneuver (3^{rd} gear - 'baseline with ESE = 'off'' (left) and with activated PS-feature (right). The enhanced engine order content is clearly visible in the right spectrogram.

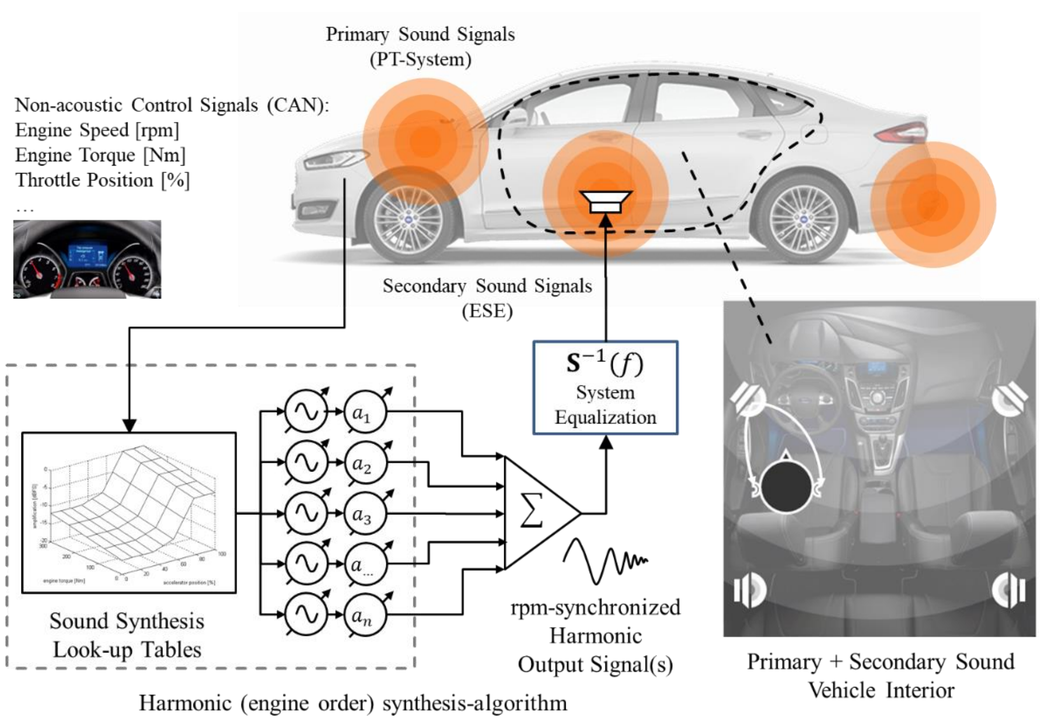


Figure 2: Overview: Active Acoustic System in order to modify the interior powertrain sound of the vehicle (ESE - Engine Sound Enhancement). The additional sound contribution is synthesized and processed by using the DSP-System of the audio head unit (AHU) and are presented by using the interior loudspeaker system (real-time sound synthesis.).

Figure 3: PS-feature: Interior powertrain sound modification: Real-time synthesized engine order signals are presented by using the vehicle's head unit and interior loudspeaker system (e.g. door woofers, center speakers and/or subwoofer).

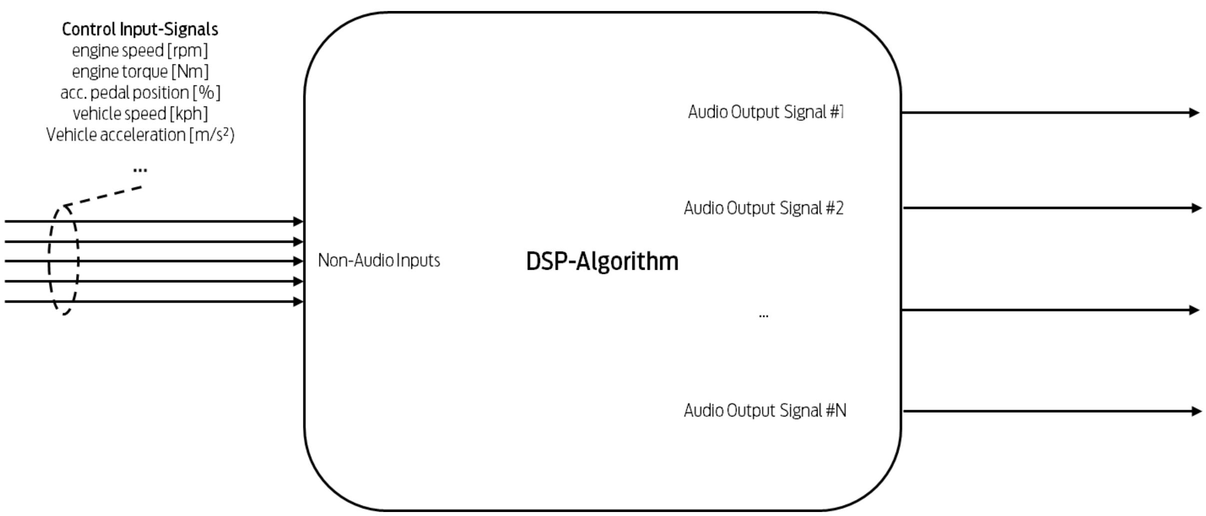


Figure 4: Block Diagram of the PS-feature. Here, the control inputs are shown in order to control the real-time synthesis (e.g. engine speed, pedal position, vehicle longitudinal acceleration, etc.). The sound synthesis is controlled by using suitable control signals (CAN-messages/signals), such as instantaneous engine speed, engine torque and instantaneous position of the throttle pedal. Furthermore, the stream outputs are shown that are routable towards the physical output channels of the audio head unit (AHU).

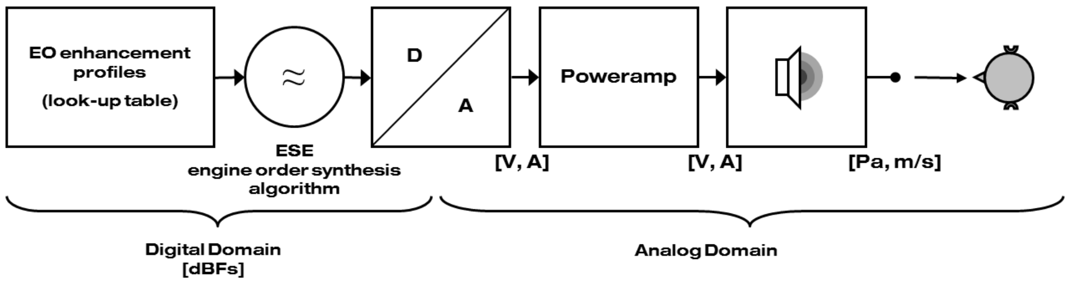


Figure 5: Block-diagram: The engine order signals are created in the digital domain (using the DSP-capabilities of the audio system) as discrete-time signals. At the system output, the digital sound streams are made available as analog signals (after D/A-conversion), amplified and emitted by the sound actuators (loudspeaker system).

3.1.1 General Propulsion Sound Requirements and Features

Item	Value/Amount	Unit	Comment
Supported audio input channels (microphone/sensor inputs)	8	-	Audio input channels to be freely configurable to support the equalization feature of the synthesis algorithm. The audio inputs are not physical parts of the infotainment system but of the required external sound periphery (Tuning PC Pro-audio grade sound interface/sound card) for measurement/system identification purpose.
Supported audio streams (loudspeaker/actuator output streams)	6	-	Routing of digital output streams to be freely assignable to physical system-outputs (analog power-amplifier-to-loudspeaker outputs)
Simultaneously usable/effective engine orders (additive synthesis algorithm)	32	-	Synchronization signals, order index (N_{eo})/order frequency range [Hz] and order phase shall be freely controllable for each engine order (within specified ranges)
Operating mode/control mode	-	-	Open loop/playback; closed-loop control resp. 'sound-to-target'-capability ([dBSPL]) is <u>not</u> required
Switchable drive modes	8	-	Propulsion sound parameter sets need to be switchable online/during runtime (engine running/vehicle driving) in order to cover different acoustic scenarios, such as vehicle drive modes and operating states of the vehicle/PT-system. Suitable crossfade/interpolation algorithms need to be supported for the sound synthesis parameters and/or the secondary path transfer function datasets stored within the PS-tuning profile resp. required tuning parameters in general.
Grain-/sample-based synthesis engine	2 instances		Engine speed-adaptive granular synthesis algorithm to create broadband and impulsive sound patterns (e.g. engine order clusters with modulated residual noise components; complements the additive synthesis algorithm (superposition of engine orders); to be pitch-able/synchronized with the additive synthesis algorithm
ESE Supported frequency range: f_{min}	20	[Hz]	Lower frequency limit
ESE Supported frequency range: f_{max}	2000	[Hz]	Upper frequency limit
EVSE Supported frequency range: f_{min}	20	[Hz]	Lower frequency limit
EVSE Supported frequency range: f_{max}	12000	[Hz]	Upper frequency limit
PS-synchronization signal input #1: Engine crank shaft rotational speed range	600	[rpm]	Lower rotational speed limit/min. value: e_{min}
	8400	[rpm]	Upper rotational speed limit/max. value: e_{max}
PS-synchronization signal input #2: vehicle speed	0	[kph]	Lower rotational speed limit/min. value: e_{min}
	400	[kph]	Upper rotational speed limit/max. value: e_{max}
Further control input-signals	1	[rpm]	synchronization signal (engine crank shaft rotational frequency)
	1	[rpm]	synchronization signal (vehicle speed)
	1	[Nm]	engine torque
	4	[Nm]	wheel torque
	1	[%]	percentage of max wheel torque

	1	[%]	accelerator pedal position
	1	[kph]/[mph]	vehicle speed
	1	[m/s^2]	longitudinal acceleration
	1	-	vehicle drive mode
	1	-	auto-tow haul mode
	1	-	HMI-controlled ‘customer specific’-drive mode
	1	-	HMI-controlled Propulsion Sound on/off
	1	-	gear position/ratio
	2	-	tailgate-/liftgate opening status
	1	-	sunroof/panoramic roof opening status
	1	-	convertible roof opening status
	1	-	active exhaust status signal
	4	-	audio system status signals
	4	-	side door opening status signals
	2	-	tailgate-/liftgate opening status

Table 1: General PS-feature Requirements and Features

3.1.2 Signal Path

This section describes the requirements and functionality related to the control and audio-signal path of the ANC-feature.

3.1.3 Signal Path – High-Level Overview

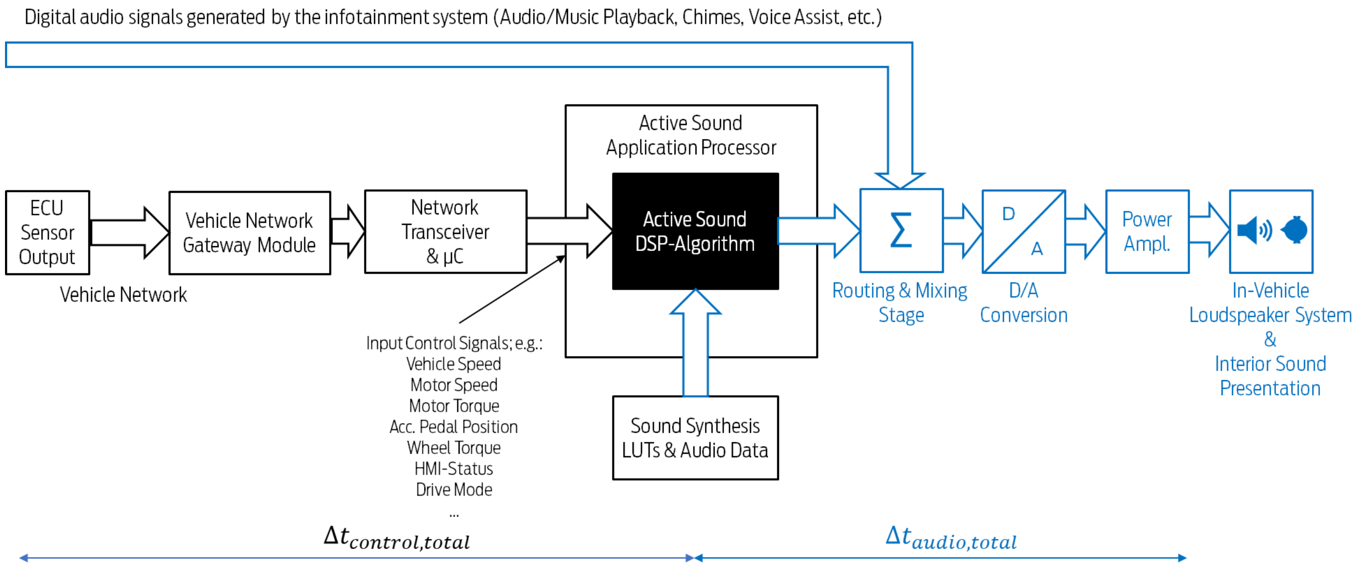


Figure 6: Generic block diagram on control- and audio-related components within the signal path from a high-level perspective. The control signal path is shown in black color; The audio signal path is shown in blue color. Further, variables for the control- and audio-latency are shown below the block diagram (i.e. $\Delta t_{control,\,total}$, $\Delta t_{audio,\,total}$).

3.1.4 Control Signal Path

The PS-feature must be capable to interface with the host system, involved functional blocks and components that are required to send and to receive the control signals required to ensure an error-free and efficient execution of the feature.

All data objects and parameters that are related to the audio signal must be managed within the PS-feature ecosystem - namely on the host system, within the DSP-algorithm, within the PS-application controller and within the PS-tuning toolchain.

Further, all related data must be storable within the actual PS-tuning projects and within the PS-tuning datasets resp. PS-calibration files.

3.1.5 Audio Signal Path

The PS-feature must be capable to interface with the host system, involved functional blocks and audio components (e.g. external amplifiers, additional modules that are responsible to distribute the PS-output-streams to the actuator outputs).

All data objects and parameters that are related to the audio signal must be managed within the PS-feature ecosystem - namely on the host system, within the DSP-algorithm, within the PS-application controller and within the PS-tuning toolchain.

Further, all related data must be storable within the actual PS-tuning projects and within the PS-tuning datasets resp. PS-calibration files.

3.2 Critical Feature Requirements

In the following sections, the critical requirements in terms of latency and timing are described.

3.2.1 Real-Time Requirements: Host System and PS-feature

The active sound system and involved functional components needs to provide real-time capabilities to ensure a stable system operation during runtime without artifacts. The active sound system needs to satisfy hard DSP-real-time requirements. Functional and performance issues related to the overall system design and feature execution on the host system, on the DSP-subsystem and on further required components need to be avoided - such as real-time violations, e.g. causing audio dropouts, glitches, buffer underruns, noticeable delays, degradations in system performance and audio quality, etc..

3.2.2 Signal Path: Temporal Behavior & Allowable Latency

The term ‘latency’ is the time duration resp. time delay that a signal needs to pass from the signal source to the sink within a given system.

A breakdown analysis and optimization of the signal path considering each contributing component is required. Special care needs to be taken as soon as multiple functional blocks and components are included in the signal path. The impact of each functional blocks and components on system behavior and resulting temporal behavior shall be analyzed and quantified in order to ensure an efficient, robust and stable performance.

The following requirements related to the allowable latency values both for the status, control and audio signal path will be described in the following paragraphs.

Figure 6 (high-level block diagram) gives an overview over the different signal paths that are most relevant in terms of latency impact.

3.2.3 Audio Signal Path: Temporal Behavior & Allowable Latency

The input-to-output latency (IO-latency) refers to the time duration resp. time delay (Δt in [ms]) that an input signal needs to pass from sensor input to the actuator output on the audio system.

$\Delta t_{IO,audio,\,total}$ is the total allowable IO-latency of the audio signal path. The total latency of the audio signal path may not exceed:

$$\Delta t_{IO,audio,\,total} \leq 3\,ms \tag{1}$$

3.2.4 Control Signal Path: Temporal Behavior & Allowable Latency

$\Delta t_{control,\,total}$ does refer to the total allowable time latency (in [ms]) of the control signals passed from the physical resp. reference sensor (here engine crankshaft and/or transmission output shaft) through the connected ECU(s) through the vehicle network to the input of the PS-DSP-algorithm.

For ANC the control signals can be specified into different categories with different allowable latency resp. timing priority, i.e.:

1. Synchronization-Signal: This is a critical signal with a high timing priority signal. A synchronization signal is used to track and to adapt the instantaneous frequency of the synthesized sound-signal components:

$$\Delta t_{control,\,sync,total} \leq 30\,ms \tag{2}$$

2. Dynamic Control Signals: This type of signal is used in order to modify the dynamic behavior (amplitude envelope) of the synthesized audio/sound signal. For instance, this can be achieved by the application of tunable weighting factors to the different sound components generated within the sound synthesis DSP-algorithm:

$$\Delta t_{control,\,dyn,\;\;total} \leq 100\,ms \tag{3}$$

3. Status Signals: These are low-priority signals used e.g. to so that the PS-feature can react and adapt due to changed operating conditions within the vehicle environment. The vehicle system will provide status information (e.g. in order to switch PS-parameters resp. tuning profiles; e.g. according to the recently selected drive mode, HMI-status signals, active sound on/off-switch, etc.).

$$\Delta t_{control,\,status,total} \leq 500\,ms \tag{4}$$

3.2.5 Control Signal-Path: Further PS-input signals

The control signals will to be transferred over the vehicle CAN-Network through the host system, through the PS-application processor to the PS-DSP-algorithm.

Additional internal signals on the host system will be transferred to the PS-DSP-algorithm.

Suitable interfaces must be available to enable data communication between the PS-feature, the PS-application processor, the host system and the vehicle system/vehicle network.

3.3 ESE – Narrow-band/engine order synthesis algorithm

Within this section the properties and features of the narrow-band synthesis algorithm (*'engine order harmonic synthesis'*) are described. The term narrow-band means, that the PS-feature is capable of synthesizing harmonic signals that are composed by a superposition of harmonic components (sine/cosine signals resp. harmonics). These harmonics correspond to the term 'engine order' or 'engine harmonic'.

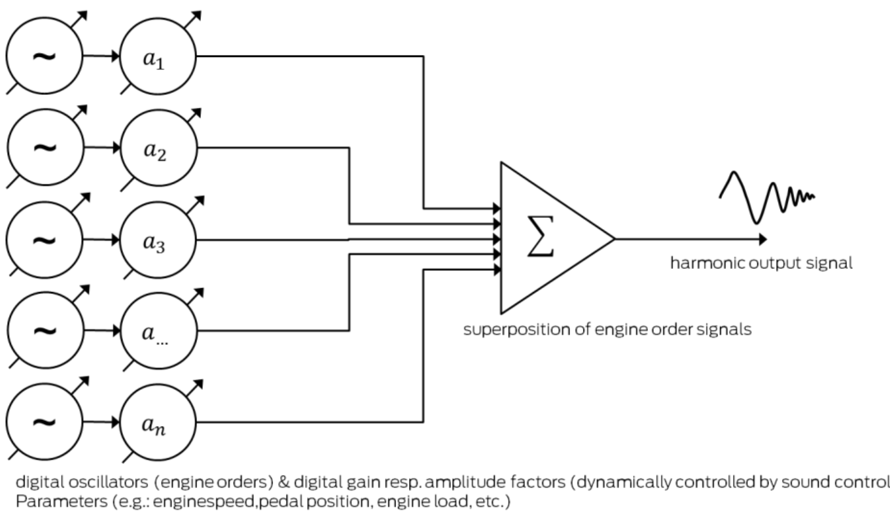


Figure 7: Block-diagram: ESE-narrow-band sound synthesis algorithm: Here, the digital oscillators (DCOs) and digitally controlled amplifiers (DCAs) are shown. Each pair of a digital oscillator and DCA synthesizes corresponds to a single engine order. The DCO frequency is synchronized to the instantaneous rotational frequency of the engine. Furthermore, the phase angle of each engine order can be controlled. The sinusoidal output-signal of a DCO is weighted with an weighting factor a_i. Weighting factors are a function of the assigned control input signal (e.g. engine torque, vehicle speed or accelerator pedal position).

The relation between the instantaneous engine order frequency and the rotational frequency of the engine crankshaft resp. the instantaneous engine speed is described in the following equation:

$$\mathrm{f_{eo}(t)} = \frac{\mathrm{e(t)}}{60}\,\mathrm{i_{eo}} = \mathrm{f_{rot}(t)\,i_{eo}} \qquad (5)$$

Equation 1: Relation between engine order frequency and rotational frequency.

with:

- $f_{eo}(t)$:= *instantaneous engine order frequency* - [Hz]
- $\mathrm{e(t)}$:= *instantaneous engine speed* - [min^{-1}]
- $\mathrm{f_{rot}(t)}$:= *instantaneous rotational frequency - [Hz]*
- $\mathrm{i_{eo}}$:= *engine order index*

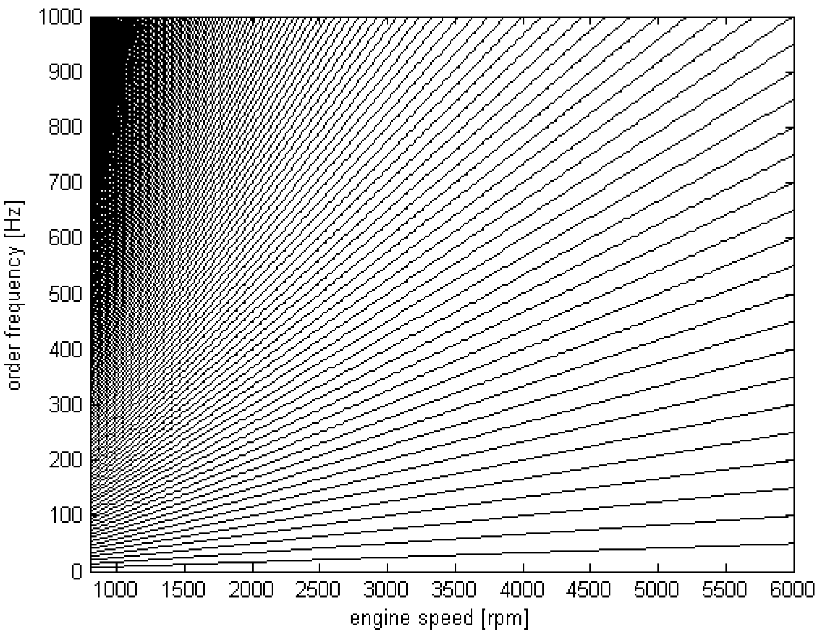


Figure 8: Relationship of engine speed [rpm] and engine order frequency [Hz] (starting with the 0.5th engine order). Here, the frequency lines for a half engine order spacing are shown.

3.3.1 Narrow-band Engine Order Synthesis Algorithm – Properties of the digital oscillators

The output signal of a single digital oscillator needs to be a sinusoidal function whose frequency and time-course is synchronized to the crankshaft instantaneous rotational frequency resp. the instantaneous engine speed of the powertrain.

Furthermore, the amplitude of the output-signal needs to be modulated by an amplification factor, which is dependent of/controlled by input parameters such as accelerator pedal position, engine torque, etc.

The equation for a single engine order with index i_{eo} is as follows:

$$s_i(t) = A_i\left(s_{torque}, s_{acc.ped.pos.}\right)\,\sin\left(i_{eo}\,\frac{2\pi e(t)}{60} + \varphi_{0,i}\right) \qquad (6)$$

Equation for a digital oscillator (narrow-band engine order synthesis) and amplitude weighting factors

with

- i_{eo} := *engine order index, e.g. i = {2nd, 2.5th, 3rd, …}*
- A_i := *time-variant amplification factor in* [dBFS] *or in* [dBrel] - *depends on control signals (accelerator pedal position, engine torque, etc.).*
- $e(t)$:= *instantaneous rotational frequency - serves as a frequency reference [rpm] for the synthesized engine orders.*
- $\varphi_{0,i}$:= *phase angle of the particular engine order: The phase offset is a relative value between the synthesized engine orders.*

- s_{torque} := *engine torque signal* [Nm]
- $s_{acc.ped.pos.}$:= *position of the accelerator pedal* [%]

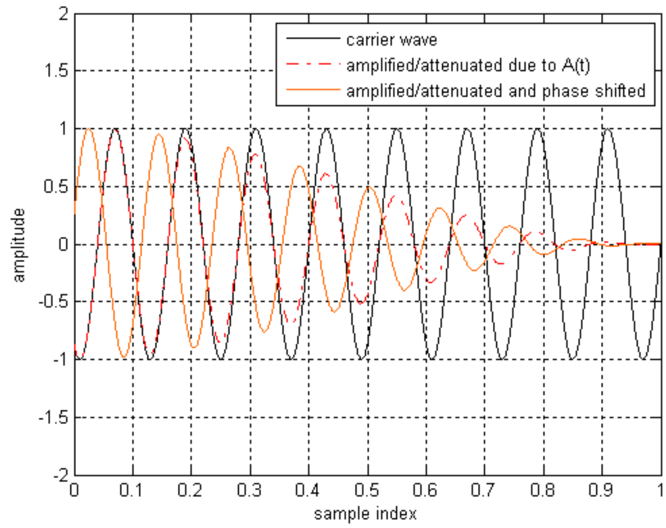


Figure 9: Example of an oscillator output signal: narrow-band carrier wave (sinus-function), scaled carrier wave with time-variant amplification factor ($A_i(t)$) and phase shifted and amplitude scaled carrier wave.

3.3.2 Available frequency range of the synthesized signals

The synthesis algorithm needs to cover a frequency range of f_{min} = 20 Hz up to f_{max} = 1.1 kHz for each engine order. Thus, the internal sampling rate of the synthesis algorithm needs to fulfill the following criterion:

$$f_s \geq 2\ f_{max} \geq 2.2\ kHz \tag{7}$$

3.3.3 Amount of synthesizable engine orders

The PS-feature needs to support 32 engine orders simultaneously.

The DSP-hardware needs to provide enough DSP-resources in order to synthesize all 32 engine orders simultaneously with a sufficient sample rate, look-up table resolution (s. above) and without audible artifacts.

3.4 ESE - Engine order-related sound synthesis parameters

In the following sections the requirements for order-related parameters for the sound synthesis are described.

3.4.1 Engine order level/narrow-band amplitude

The narrow-band level [dB(SPL)] of each engine order needs to be freely definable in a parameter table, e.g. amplitude vs. engine speed [rpm].

Moreover, various frequency-weighting curves - such as ([dB(SPL(A))] and [dB(SPL(B))]) - need to be selectable and applicable to the engine order data (amplitude vectors) in order to consider the loudness perception in human hearing.

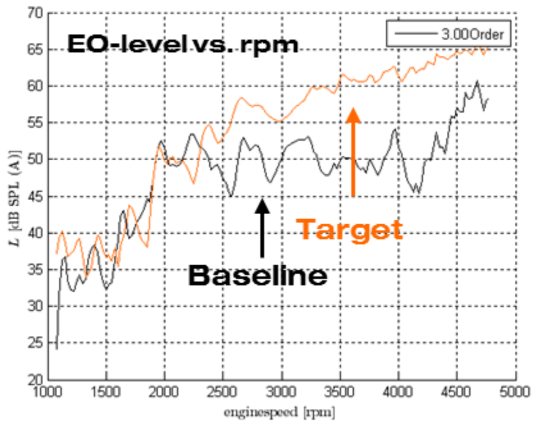


Figure 10: Passenger Compartment/Vehicle interior: Engine order level (3rd engine order - narrow band - [dB(SPL(A))]*) vs. engine speed* [rpm]. *Here, the baseline condition and the engine order level with activated synthesis (target) is shown.*

3.4.2 Engine order phase

The phase of each engine order needs to be storable in a parameter table, namely in [°] vs. engine speed [rpm].

The phase-values need to be available within in the following interval:

- φ ε [-180° … +180°]
- φ ε [-π rad… +π rad], alternatively.

Alternatively, if the engine order tables are in complex format, then the phase is already contained in the engine order tables and no separate table for the phase-data is required.

The PS-feature needs to support at least *static* phase offset values that can be chosen arbitrarily in the specified range (s. above) and that can be separately assigned to each generated engine order oscillator. This allows to assign pseudo-randomized phase offsets to each synthesized engine order.

Alternatively, the data can be stored in complex format (Cartesian format - real part and imaginary part).

Within the tuning tool the order-related data - as described above - shall be displayed and editable in terms of amplitude, order level and order phase.

3.4.3 Interpolation for real-time synthesis

The control parameters need to be interpolated between neighboring table values, due to the limited resolution. Here, methods, such as *linear*, *quadratic* or *spline* interpolation can be applied.

The following quantities need to be interpolated, when read from the look-up tables:

- A_i := *engine order amplitude/level at particular engine speed*
- $\varphi_{i,}$:= *phase values of the particular engine order at particular engine speed*
- *Weighting factors* [dB(relative)] - *dependent on the Control-input signals (via CAN)*

Alternatively, it is sufficient to apply a smoothing filter to the time signal at the output of the engine order synthesis stage (DCO + DCA) which creates the same or better quality than the interpolation-algorithm of the look-up tables (s. above).

3.4.4 Control-Input Signals/CAN-Signals

The system needs to support the following Control-input signals, namely engine speed [rpm] plus up to x simultaneously usable control parameters.

The NAI-Signals need to be freely assignable to one of the NAI-related look-up tables (s. subsection 3.7.4).

3.4.5 Control Signals – Ranges and Limits

The range of the CAN-Signals must be mapped to the assigned lookup-table automatically.

The required ranges of the control signals are shown in

Table 2.

Quantity:	Minimum limit:	Maximum limit:	Unit:	Synchronization Signal	Weighting factor	Comments
Engine Speed	600	7200	rpm	Yes	Yes	*to be supported as an weighting factor
Engine Torque	-120	600	Nm	No	Yes	-
Accelerator Pedal Position	0	100	%	No	Yes	-
Vehicle Speed*	0	655.35	kph	Yes	Yes	*to be supported as a synchronization signal (order frequency)
% Max Wheel Torque	-100	100	%	No	Yes	-
Vehicle Longitudinal Acceleration	-0.25g	+6.0g	m/s^2	No	Yes	-

Table 2: Supported control signals (Control Input Signals - provided via the vehicle's CAN-network) and supported ranges. In addition, it is shown whether the control signals need to be used as weighting factors and/or as synchronization signals (order frequency).

3.5 EVSE Sound Synthesis Related Requirements

The required capabilities and functionality to support advanced sound design use cases for electric vehicles will be described in the following sections.

3.5.1 EVSE Sound Synthesis Algorithm – High Level Overview

The EVSE component of the PS-feature must support grain-/sample-based synthesis algorithms in order to create more complex sound patterns as compared with the engine order synthesis algorithm, both with stochastic and harmonic signal content.

The synthesis algorithm must be capable to create natural sounding sound patterns without looping interpolation or any other artifacts (e.g. by resynthesizing sound patterns with impulsive order content resp. clusters of engine orders/engine harmonics and residual noise components).

The EVSE-system must provide suitable methods and subroutines to analyze, extract, decompose, store and re-synthesize existing or simulated interior noise measurement data (i.e. time-signals, e.g. engine –run-up maneuvers).

The PS-ecosystem must provide look-up tables to store the extracted synthesis parameters on the host platform flash memory. Furthermore, the data must be included in the PS-calibration file.

The synthesis algorithm must be capable of reproducing statistic variations of the source data in the re-synthesis for a plausible sounding result.

The EVSE-DSP-algorithm must support the additive synthesis-algorithm and the grain-/sample-based synthesis simultaneously.

The EVSE-DSP-algorithm must support multiple independent instances of the grain-/sample-based engine.

The synthesis algorithm must support the same or at least a subset of the control parameters of the additive-synthesis-algorithm.

The synthesized signals must be modifiable in terms of fundamental frequency F_0 resp. in pitch controlled by synchronization input signal.

The tuning tool must provide a suitable user interface to analyze, to set and modify the tuning parameters of the grain-based synthesis engine(s).

The EVSE-feature must support a set of freely connectable and configurable DSP-blocks in order to provide a flexible environment to create customized and complex synthesis scenarios and sound designs. The tuning tool SW-application has to support the configuration, storage and SW-download of the DSP-block-configurations as a part of the sound dataset resp. calibration file and together with the remaining PS-tuning parameters and PS-related data.

3.5.2 EVSE Tuning Parameters & Perceptual Requirements

The following audio-acoustic - and corresponding psychoacoustic - quantities shall be modifiable by the sound synthesis algorithm and the tuning tool (‘sound synthesis parameters’). These required parameters are grouped after the physical quantities and additionally after the primarily affected psychoacoustic quantities:

- Level/Gain/Loudness
- Dynamic Level/Dynamic Gain/Time Envelope/Dynamic Loudness
- Mixing Level/Relative Loudness/Source Balance
- Spectral Magnitude/Spectral Envelope/Spectral Centroid/Character/Timbre/Color/Specific Loudness

- Spectral Phase/Hilbert Envelope/Temporal Modulation/Temporal Fine Structure
- Spectral Modulation/Frequency Modulation/Amplitude Modulation
- Harmonic-/Order Index Value
- Harmonic-/Order Phase Value
- Harmonic-/Order Level
- Harmonic-/Order Level
- Fundamental Frequency/Repetition Rate/Pitch Shift/Pitch Envelope/Virtual Pitch
- In-vehicle Loudspeaker Routing & Mixing
- Loudspeaker Equalization/Binaural Localization/Spaciousness/Sound Source Width
- Audio Output Limiting/Soft-Clipping
- Control Signal Processing; Scaling, shifting, mapping and routing to relevant sound synthesis blocks

3.5.3 EVSE Algorithm DSP-Objects

The interior active sound feature needs to support the following DSP-blocks with the following capabilities:

1. Order-based synthesis blocks (sinusoidal signal digital oscillators)
2. Sample-based synthesis blocks: 'Wavetable'/'lookup-table' digital oscillators with arbitrary audio signal content

The instantaneous frequency (= repetition rate) of each oscillator-block needs to be controllable by an assignable synchronization signal (e.g. vehicle speed) in order to re-pitch the original waveforms generated by the oscillator. The oscillators may make usage of digital audio buffers which are able to store and to load an arbitrary waveform from non-volatile and/or volatile memory (ECU-flash memory and/or from RAM).

The oscillators need to support a high-quality digital audio resampling resp. interpolation algorithm in order to create resampled (resp. re-pitched) waveforms of the original oscillator waveform.

Further, the digital oscillators need to allow to modify the static phase offset resp. the instantaneous phase shift stored in a look-up table and controlled by the synchronization signal.

The system needs to provide enough DSP-resources to host multiple instances of the DSP-blocks in parallel during runtime. The relation on DSP-load and the amount of the instances used in parallel/simultaneously during runtime shall be analyzed in detail in the context of minimum and maximum allowable resource usage ('resource budget'; e.g. DSP-load, required memory, sampling rate, latency, etc.). A breakdown is required in order to understand the possibilities and constraints of the host system (Ford-electrical architecture and infotainment system).

3.5.4 EVSE - Further Sound Processing DSP-Objects

The active sound feature needs to support further DSP-blocks for additional synthesis control and audio processing tasks. Figure 11 shows an example block diagram of an interior active sound algorithm. The block diagram is slightly modified in order to support interior sound presentation over the in-vehicle loudspeaker system (e.g. by using suitable transducers such as low-, mid-woofers, tweeters and subwoofer-systems).

The example block diagram shows the following DSP-blocks that can be controlled in a static and/or dynamic manner by the control signals and sensor data transferred over the vehicle network (e.g. accelerator pedal position, torque information and others). Table 3 summarizes the purpose, essential requirements and amount of the involved DSP-blocks.

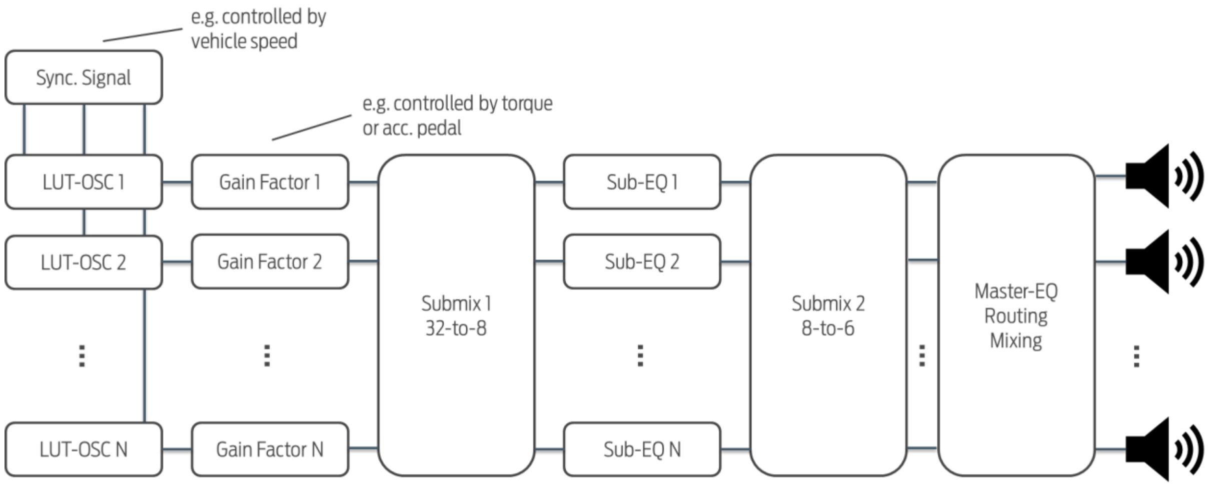


Figure 11: Example Block Diagram of a customized configuration of an EVSE-DSP-algorithm:

DSP-Block:	**Purpose:**	**Amount:**
Synchronization Signal Control Signal	Control oscillator repetition rate resp. instantaneous frequency [Hz] Reference quantity to control re-pitching/resampling of the oscillator-waveform Two synchronization signals must be supported simultaneously Input control signal needs to be freely assignable/routable and scalable to each oscillator-block (synchronization-input)	2
LUT OSC 1 … N	waveform/LUT-synthesis (sinusoidal and/or assignable audio wave table data)	32
Gain Factor	amplification and attenuation factor of oscillator output (amplitude weighting factor), supports static and dynamic gain control for multiple input signals	32
Sub-Mix 1	grouping and (down-)mixing of LUT-oscillator output signals oscillator output signals must be freely groupable, mixable and routable mixing must be statically and dynamically controllable 32 Inputs-to-8 Outputs	1
Sub-EQ	equalization/spectral filtering of grouped oscillator signals (Submix 1 output streams) multistage parametric EQ each Sub-EQ-block needs to support individual EQ-/Filter parameters EQ and filter parameters need to be statically and dynamically controllable	8
Sub-Mix 2	grouping and (down-)mixing of Sub-EQ output signals freely groupable, mixable and routable mixing must be static and dynamically controllable 8 Inputs-to-6 Outputs	1

Master-EQ/Routing/Mixing	Compensation/fine-tuning to vehicle characteristics (interior sound behavior) multistage parametric EQ per channel inter-channel-differences (e.g. inter-channel phase- and level-differences), Allpass-filter-network, IIR- and/or FIR-filters calculation of EQ-curves need be supported based on vehicle cabin and audio system impulse response data resp. complex transfer function data for each secondary transfer path acquisition of audio-acoustic impulse response and/or complex transfer function data needs to be supported by the EVSE-feature, host system and tuning tool mixing/routing matrix to address all available power amplifier outputs resp. loudspeaker channels as supported by the host system 6 Inputs-to-6 Outputs	1

Table 3: Example of a customized configuration of an EVSE-DSP-Algorithm: DSP-functional blocks & resources: purpose, requirements and amount.

3.5.5 EVSE - Configurable DSP-objects

The EVSE-feature must support different types of DSP-objects resp. DSP-blocks with different capabilities and purpose, such as control logics, processing/conditioning of control signals and audio signals. The following types of DSP-blocks shall be supported by the PS-feature:

		Category		
Description	**Purpose**	**logics proc.**	**control proc.**	**audio proc.**
2D-transfer function	lookup-table transfer function, editable, f(x)	N/A	X	X
abs	calculate absolute value	N/A	X	X
add	add values/signals	N/A	X	X
amplitude modulation	modulate multiple signals	N/A	X	X
amplitude modulation/LFO	modulate signal/low frequency oscillator	N/A	X	X
AND	logic AND	X	X	N/A
cast	data type conversion	X	X	X
clipping	clip value/signal	N/A	X	X
decision	switch-case decision	N/A	X	X
delay	delay signal by x samples/x seconds	N/A	X	X
division	divide values/signals	N/A	X	X
EQU	==	X	X	N/A
FFT	Fourier-transform/spectral audio processing	N/A	N/A	X
FIR	FIR-filter	N/A	N/A	X
frequency modulation	modulate signal by using another signal	N/A	X	X
gain	amplify/attenuate signal	N/A	X	X
generator/oscillator	generate arbitrary waveform	N/A	X	X
GEQ	>=	X	X	N/A
grain synthesis	time granulation of audio signal, pitch-able	N/A	N/A	X
IFFT	inverse Fourier-transform/spectral audio processing	N/A	N/A	X
input	signal input	X	X	X
LEQ	<=	X	X	N/A
LP/HP/BP - dynamic	filter signal - adaptive center/corner freq.	N/A	X	X
LP/HP/BP - static	filter signal - fixed center/corner freq.	N/A	X	X

mix	signal mixer	N/A	X	X
Multiband-EQ	multiband-EQ, filter-array/loudpsekaer-EQ	N/A	X	X
multiply	multiply signals	N/A	X	X
NEQ	!=	X	X	N/A
noise generator	generate pseudoranom noise signal (pink, white, …)	N/A	X	X
NOT	!	X	X	N/A
OR	OR	X	X	N/A
output	signal output	X	X	X
random number generator	generate random numbers, event-based trigger	N/A	X	X
routing matrix	route signal(s) from source to sink	N/A	X	X
shepard-filter	shepard-filter	N/A	N/A	X
shepard-tone	shepard tone-generator	N/A	N/A	X
sign	return sign of signal	N/A	X	X
soft limiter	soft clipping of input signal with adj. Threshold	N/A	X	X
store parameters	store parameters of the DSP-object	X	X	X
subtract	subtract signals	N/A	X	X
timer	set timer, adj. time-intervall/time-constant/tau	X	X	X
timer-ramp	generate timed ramp-signal, adj. time-constant/tau	N/A	X	X
wave-player	Loop-able and pitchable wave-player	N/A	N/A	X
wave-player	wave-player, event-based trigger	N/A	N/A	X
XOR	XOR	X	X	N/A

Table 4: EVSE - Configurable DSP-objects

The DSP-objects must be manageable within the PS-tuning tool SW-application. The data resp. signal flow must be editable in such a manner that DSP-objects can be connected with each other ('patched together'). The signal flow is represented by the graphical objects ('widgets') within the user interface of the tuning tool SW-applications.

3.5.6 EVSE - Control Signals

The following *generic*[1] input signals provided by the vehicle network - providing required sensor data and corresponding physical quantities - shall be supported in order to allow fully interactive and real-time capable modification of the sound synthesis parameters. Further, the *generic* control signals are classified in terms of criticality of control latency and timing. The control signal can be divided into three categories in terms of criticality, namely:

1. '**High**-priority signals'
2. '**Medium**-priority signals'
3. '**Low**-priority signals'

Additional requirements on allowable signal latency are mentioned within section 'Control Signal Path: Temporal Behavior & Allowable Latency'.

Moreover, the control signals can be divided into different categories according to properties, purpose and usage:

1. **Synchronization signals:**

   Signals with '*quasi-continuous*' properties; e.g. used to control of oscillator repetition rate/instantaneous frequency/pitch shift

2. **LUT-input signals:**

   Signals with '*quasi-continuous*' properties, e.g. used to control order/oscillator gain LUTs or other sound modifier-LUTs; e.g. dynamic gain factors, filter-cut-off-frequency, mixing-level, etc.)

3. **Status Signals:**

   Signals with '*event-based*' properties. These signal-types that can attain discrete numeric values or states with limited range and resolution. These signals are e.g. suitable to control transition of sound profiles ('EVSE-Drive Modes'), sound activation/deactivation or to be processed of by additional logic-blocks.

Control Signal:	**Physical Unit:**	**Purpose/Usage:**	**Signal Priority:**
Vehicle Speed	[kph]	Synchronization control, LUT-input control	high
Wheel Rotational Speed	[rad/sec] or [rpm]	Synchronization control, LUT-input signal	high
Wheel Torque	[Nm]	LUT-input control	medium
Motor Torque	[Nm]	LUT-input control	medium
Motor Speed	[rad/sec] or [rpm]	Synchronization Signal, LUT-input control	high
Accelerator Pedal Position	[%]	LUT-input control	medium
Vehicle Start/Stop-Button	Unitless	LUT-input control, logic-block control	low
Drive Mode	Unitless	LUT-input control, logic block control	low
Propulsion Sound on/off	Unitless	LUT-input control, logic block control	low
Gear-lever Position	Unitless	LUT-input control, logic block control	low

Table 5: Control signals categorized according to purpose/usage and timing priority (control latency)

[1] Ford uses a generic description of signals and physical quantities required for the active sound feature, as the signal database is currently under development and not finalized yet.

The active sound feature needs to support an input stage to manage the control signals. The input stage for the control signals needs to support additional DSP-blocks with real-time-processing capability, such as:

- time-smoothing/time-weighting (integration/lowpass- filtering)
- scaling resp. multiplication
- shifting resp. application of static offset values
- limiting resp. clipping

The required DSP-blocks need to be applicable to each control signal individually resp. the control signals need to be assignable to a set of the mentioned DSP-blocks.

3.6 Vehicle Drive Mode Support – PS-Drive Modes

The PS-feature must be capable to switch between different sound designs resp. sets of tuning parameters according to the currently selected drive mode and operating conditions on the vehicle; e.g. 'normal', 'sport', or 'track'-mode.

Therefore, the PS-feature must fulfill the following requirements:

3.6.1 Content of supported PS-Drive Modes

- independent PS-drive modes with different sound designs featuring:
    - freely assignable engine order indices across all sound designs
    - freely assignable engine order look-up tables based on a pool of 72 engine order tables
    - freely assignable sound synthesis control parameters (Control Input Signals)
    - variable engine order level data vs. control parameters, e.g. such as engine speed or vehicle speed – supported via corresponding look-up tables
    - variable data content for gain/amplitude-factor-related look-up tables multiple assignments of lookup table resources required (e.g. mapping of e.g. table 1, table 2, table x to multiple drive modes required)
- additional modes which allows to apply a static gain factor in order to change the sound level of the independent modes. The gain factor must be freely assignable to one of the six (6) sound designs.

The propulsion sound designs need to be switchable during runtime. The system must be capable to change between sound designs without noticeable delays, drop-outs or any other artifacts in the propulsion sound. Suitable signal processing methods need to support a smooth transition between sound designs (e.g. crossfading/morphing/interpolating between the previously selected and the new propulsion sound design parameter sets.

3.6.2 Drive Mode Selection - Tuning and Management

- support of multiple control- resp. status input signals is required
- simultaneous usage of drive mode input signals is required
- functionality to assign and control the PS-drive modes must be fully controllable in the PS-tuning tool
- drive mode selection must be controllable by suitable assignment resp. mapping tables
- The PS-feature must allow flexible and bijective assignments regarding input signal states and associated PS-drive modes.

- PS-drive tuning parameters must be storable in the described assignment tables
- PS-drive modes need to be selected based on a calculated index value. Here, the calculated index value is the calculation result of a subroutine of the PS-feature. This subroutine evaluates the content of the drive mode assignment table by using a suitable logic.

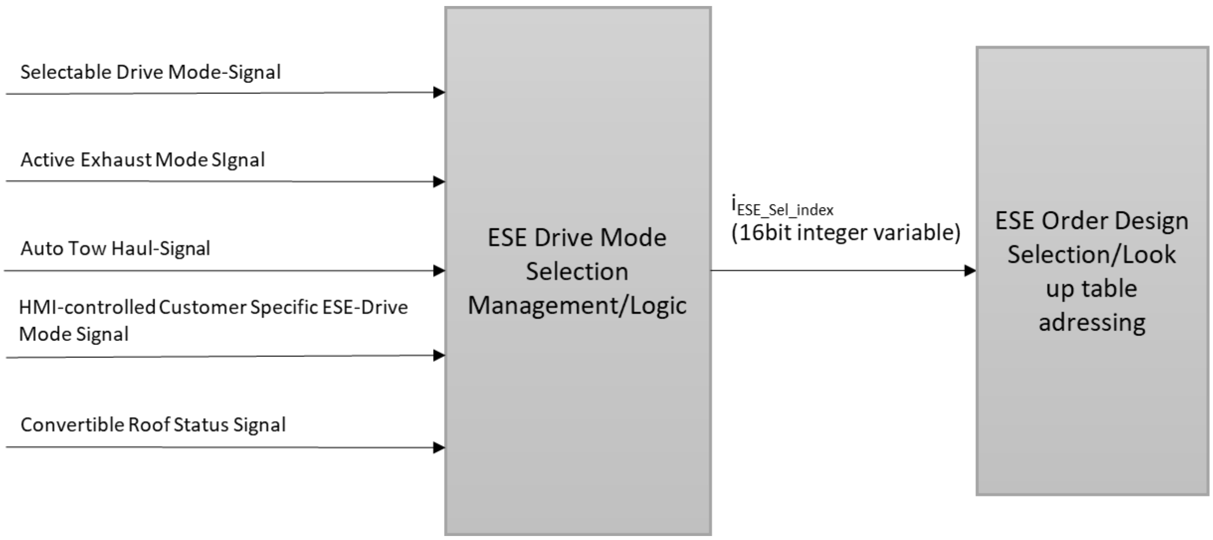


Figure 12: PS-Drive Mode Selection: Principle block diagram/schematic diagram: Input signals/interfaces.

3.7 Common Sound Synthesis Parameters and Look-up-tables

3.7.1 Look-up-tables - Engine-order profile data

The resolution of the engine order profiles resp. the parameter tables need to fulfill the following conditions:

- Engine speed resolution: $\Delta r_s \leq 40$ rpm
- Engine order level/amplitudes: word length/numeric precision ≥ 16 bits (fixed point or floating-point format)
- Engine order narrow-band phase: word length/numeric precision ≥ 16 bits (fixed point or floating-point format)
- Alternatively, the engine order profiles (see above) can be stored as complex data with at least 16-bit resolution of the real and imaginary part in fixed-point or floating-point format.
- Engine order index: The engine order index needs to be freely definable: word length/numeric precision ≥ 16 bits (fixed-point or floating-point format)

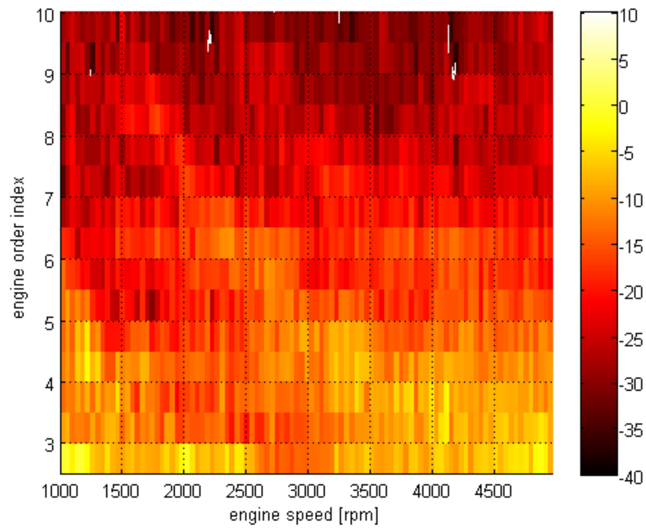


Figure 13: Color-map of the engine order profiles: Order levels [dB(SPL)] *(z-axis) vs. engine order index (y-axis) vs. engine speed (x-axis).*

3.7.2 Look-up-tables – Gain/Weighting factors

The amplification factors need to be storable in a single lookup table/matrix or in various look up-tables that provide the same functionality as a single look-up table for all amplitude-factors:

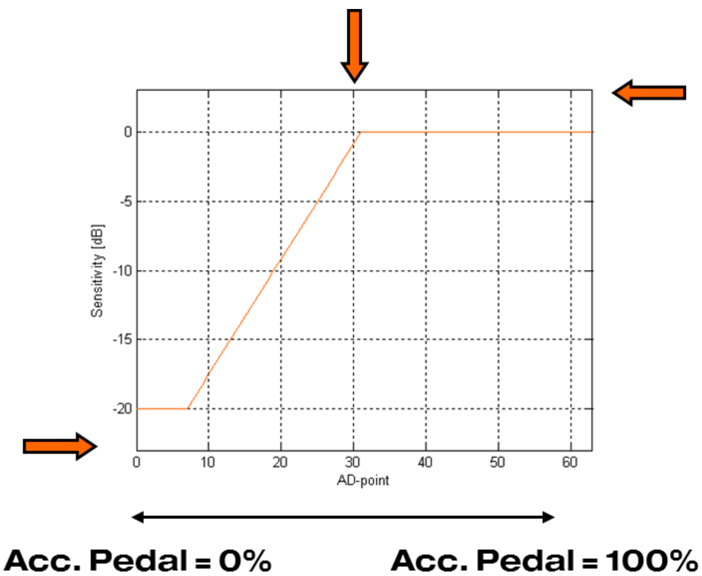


Figure 14: Example of a control parameter sensitivity curve (position of accelerator pedal) vs. amplification factor [dB(relative)].

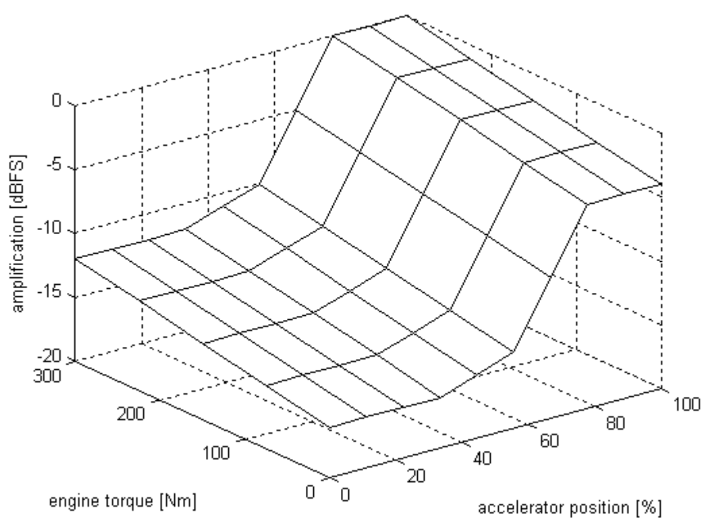


Figure 15: Example of a look-up table/parameter map based on accelerator pedal position [%] *(x-axis), engine torque* [Nm] *(y-axis) and the resulting amplification factor* [dB(relative)] *(z-axis).*

3.7.3 Look-up-tables – Control factors - size and resolution

The look-up-tables which are used for the control parameters shall have the following number of sample-points $N_{samples} \geq 64$. The absolute range (minimum and maximum values) of the lookup-tables depend on the specified ranges of the control signals used.

3.7.4 Look-up-tables – Control factor related look-up-tables

The PS-feature needs to support 60 different control-related look-up tables.

3.7.5 Look-up-tables – Control factor related lookup-tables – routing block

The control signals must be routable to the different lookup-tables.

Each look-up table input may accept only one of the control signals. Multiple control signals may not be routed to a single look-up table.

The output signals of the look-up tables have to be routable in order to control the dynamic gain blocks resp. dynamic weighting factors that modify the output signals of the audio oscillators resp. the DSP-blocks responsible for the sound synthesis.

The PS-DSP-algorithm must support routing matrices in order to control the routing and mapping of the control signals.

3.8 Audio Signal Path & Routing

The digital output streams of the PS-DSP-algorithm must be routable to the physical output-channels of the infotainment system (power amplifier resp. actuator-outputs).

The synthesized sound signals will be presented by the interior vehicle loudspeaker system.

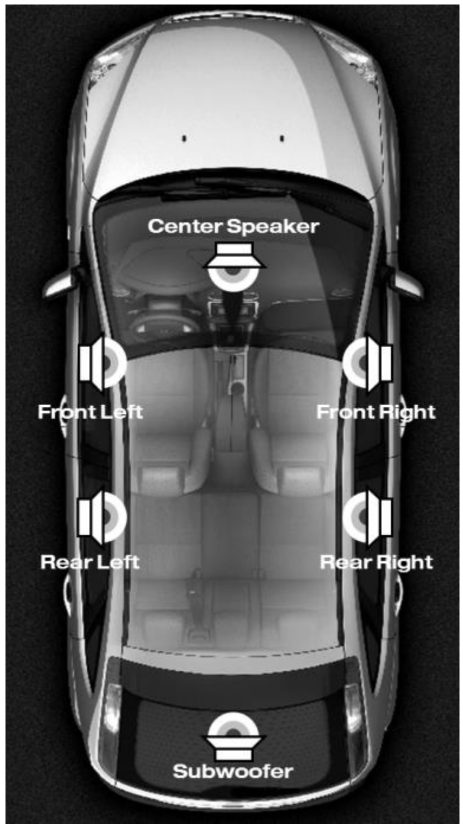


Figure 16: Example: Vehicle interior: Physical output channels (AHU/amplifier) and typical loudspeaker channels and positions.

3.8.1 Audio Signal Path: Output Streams: Interfaces, Connection and Configuration

The PS-DSP-algorithm must support independently conditioned PS-output streams.

The PS-feature must support suitable interfaces for the PS-audio streams, so that these can be transferred through the following (output-) signal path further down-stream resp. through the audio signal path of the host system and through the remaining audio components to the PS-actuators.

A high-level block diagram for the interior active sound features, the related output streams and the following audio signal path (PDC, PAC, add. audio components) is shown in *Figure 17* to better explain the high-level approach as chosen for the host system (Phoenix IVI-system architecture).

The stream outputs of the PS-feature (source) and the stream inputs of the host system (sink) need to be connected on a 'one-by-one'-basis: This means that each of the - individual - PS-stream output will be connected to a pre-defined - individual - stream input on the host-side. resp. The connection setup can be described by a 'connection graph' resp. by a pair of corresponding identifier values (IDs) indicating the involved PS-stream output and stream input on the host-side.

The 'connection graph' of the PS-stream outputs and the stream inputs on the host-side may not be changeable by e.g. the PS-tuning toolset resp. within a PS-tuning project that is created or modified by Ford NVH-engineers.

Optionally, the PS-feature may allow for deactivation of selected PS-output streams – e.g. if only a subset of PS-actuators resp. actuator out channels can be used for a vehicle configuration.

Stream Output ID - PS-component	Stream Input ID - Interface Point – Host System
0	0
1	1
2	2
3	3
4	4
5	5

Table 6: Connection of the PS-output streams and the stream inputs of the host system.

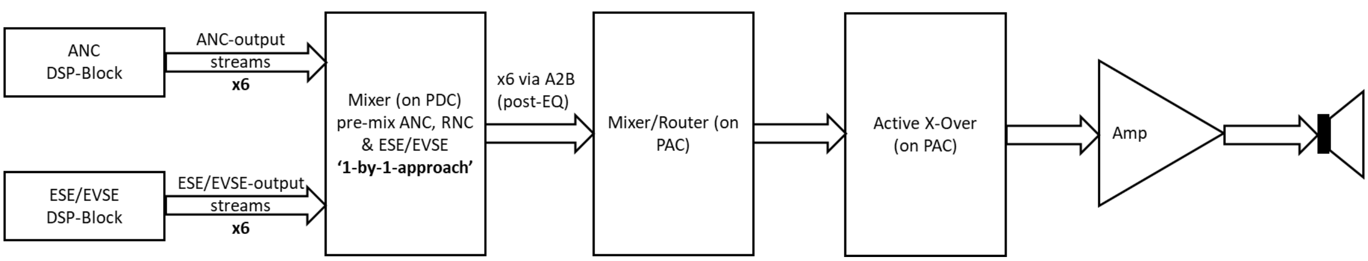


Figure 17: Audio via A2B: ASF-Mixing Routing Approach: Supported PS-output streams and X-Over for active channels (PDC → PAC → loudspeaker)

3.8.2 Audio-related Real-Time & Timing Capabilities

The PS-feature needs to provide real-time capability to ensure a stable system operation without artifacts (e.g. audio dropouts, audio glitches, audio buffer underruns, etc.). The PS-feature needs to fulfill hard DSP-real-time requirements.

3.8.3 Audio Signal Path – Routing/Mixing on external ECUs

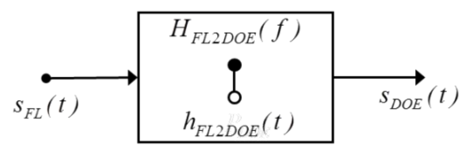
The system needs to be capable to control further mixing and routing of the PS-signals on additional ECUs included in the architecture of the infotainment system (such as additional external amplifiers or additional modules that distribute the PS-signals to the actuator outputs). All related data needs to be managed by the PS-ecosystem (namely the PS-DSP-algorithm and the PS-tuning toolchain). Furthermore, all related data needs to be storable within the actual PS-tuning projects and within the PS-tuning calibrations.

3.9 Audio Equalization-Stage and Secondary Path Transfer Functions

The PS-feature needs to be capable of compensating the acoustic behavior of different vehicle configurations. A vehicle model can have different upper body-related features - such as audio system configurations and/or body style configurations - that can affect the behavior of the secondary path transfer functions resp. secondary path impulse response functions of the vehicle cabin (s. Figure 18). An audio equalization algorithm allows to compensate acoustic differences between vehicle variants resp. system configurations and ensures that all available model variants will have the same PS-sound contribution and performance.

Moreover, the secondary transfer function characterizes the ‘acoustic sensitivity’ of the complete audio system. This means that a determined digital signal amplitude that is created by the PS-feature will cause a certain sound pressure amplitude within the vehicle cabin at a determined receiving point (e.g. head rest measurement microphone). This relation can be expressed by a complex transfer function.

The secondary transfer function is composed by the serial connection of successive system blocks, such as audio head unit (including D/A-conversion stages, reconstruction filters, and amplification stages), the electroacoustic actor systems (loudspeaker systems) and the acoustic propagations paths (room-acoustic transfer function (RTF) of the vehicle cabin).



$$s_{DOE}(t) = s_{FL}(t) * h_{FL2DOE}(t)$$

$$S_{DOE}(f) = S_{FL}(f) H_{FL2DOE}(f)$$

Figure 18: Secondary path transfer functions resp. impulse responses (here related to the driver’s head position). Here, the transfer function is affected by upperbody-specific features of the vehicle - such as loudspeaker-type, the routing of the audio-signal path and the acoustic properties of the vehicle cabin (sound package, body style/geometric properties, etc.). Moreover, the transfer function depends on source (loudspeaker) and receiver position (seat location). The block diagram shows how the loudpeaker input signal is affected by the transfer function towards the driver’s head. The output signal $s_{DOE}(t)$ is the result of the convolved input signal $s_{FL}(t)$ and impulse response $f_{FL2DOE}(t)$.

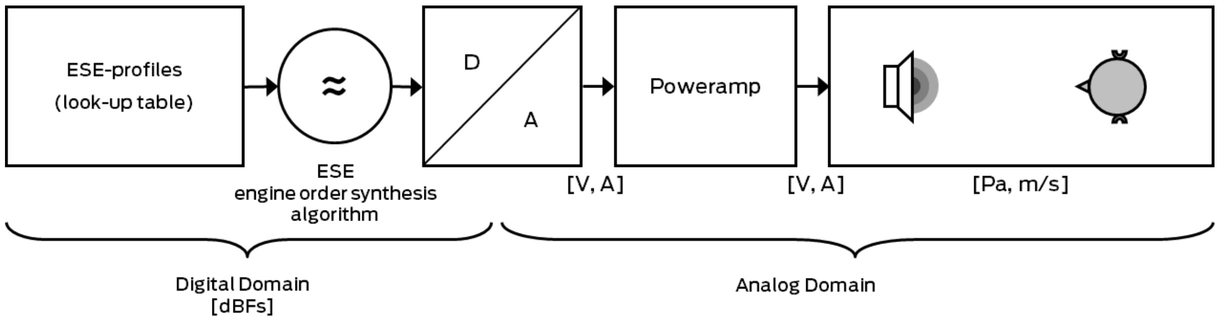


Figure 19: Block diagram in order to show the secondary path transfer function for a single path/physical output channel.

Various signal processing algorithms can be used for this purpose, e.g.:

- FIR-filters
- IIR-filters (LP, BP, HP) - non-parametric and parametric, shelving)
- all-pass-filters
- FFT/IFFT - frequency-domain/spectral processing
- Narrow-band weighting factors resp. weighting curves

3.9.1 General Requirements

The PS-feature needs to provide the following functionality:

- Each sound synthesis stage needs to be modifiable by using an equalization block, which has a fixed assignment to one of stream outputs of the PS-feature.
- Each equalization-block needs to be freely assignable to the physical outputs of the AHU.
- The amplitude response function and the phase response function of each supported loudspeaker channel need to be modifiable resp. by using the equalization block.
- The equalization algorithm needs to be capable to create a spatially homogenous sound response in all seat positions. Please refer to section 3.10 for additional information.
- Moreover, the PS-feature needs to support system calibration factors in terms of ‘absolute’ input level (e.g. digital level [dBFS]) and the resulting ‘absolute’ sound pressure level ([dB(SPL)], [dB(SPL(A))] in an appropriate receiving resp. reference position (microphone sensor position). The reference/receiving point(s) need to arbitrarily selectable by the development engineer. In practice, the reference point(s) corresponds to the head rest position(s) inside of the vehicle cabin (measurement microphones).
- The PS-tuning tool chain needs to maintain the calibration factors by storing the transfer function data. Moreover, the order levels need to be editable as ‘absolute’ sound pressure levels [dB(SPL)] with selectable acoustic spectral weighting curves supporting ‘linear’ [dB(SPL)], ‘A-weighted’ [dB(SPL(A))] and ‘B-weighted’ [dB(SPL(B))].
- The equalization feature needs to be capable to calculate the equalization parameters based on transfer function data with regularized and non-regularized (resp. narrow-band/raw vs. smoothed) frequency response data.
- The equalization algorithm and the tuning tool SW-application need to be support calculation of fractional octave smoothed TF-datasets (i.e. octave-, 3rd-octave, 6th-octave, Nth-octave, critical band/bark-smoothed).
- The tuning tool application needs to support suitable measurement, analysis and post-processing functionality to edit and to modify the TF-data used by the EQ-algorithm, e.g. averaging, curve editing, application of

compensation factors/offsets.

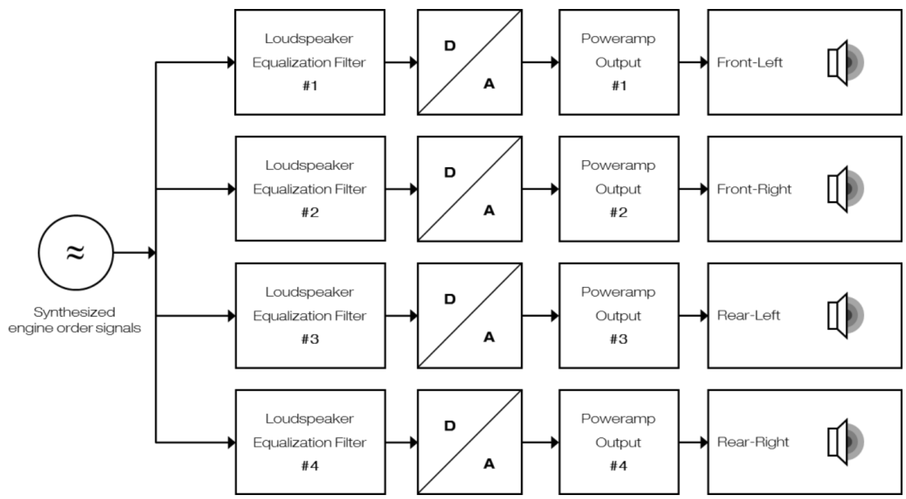


Figure 20: EQ-stage – block diagram: Synthesized engine order signals, equalization filter blocks and corresponding output channels of the audio system. Only the outputs for the door woofer loudspeakers are used within this example.

3.9.2 System Identification- and Data Acquisition-Capability

The PS-feature and PS-development toolchain needs to support appropriate functionality in order to acquire the required transfer function data for each secondary path (configured output channel to sensor position (measurement microphones)). This results in a matrix of measured transfer functions.

ESE stream output Outputs	Sensor Positions (measurement microphones)
#1	#1''
#2	#2''
#3	#3''
#4	#4''
#5	#5''
#6	#6''
-	#7''
-	#8''

Table 7: Stream outputsvs. input channels for the secondary path transfer function measurements (6-by-8).

The PS-development toolchain and PS-feature need to support the following items:

- Up to eight (8) sensor input-channels (measurement microphones) in order to measure transfer functions inside of the vehicle cabin.

- Generation/audio playback of suitable measurement signals for system identification (e.g. sine sweep, multi-sine, pseudo-random noise or other suitable signals)
- Microphone calibration functionality in order to calibrate the measurement inputs to sound pressure levels

The PS-DSP-algorithm needs to be capable to generate measurement signals on the host system.

3.9.3 Transfer Function Data Format - Amplitude Response

The amplitude response functions need to be measurable and storable as calibrated values, e.g. in [Pa/V] or in [Pa/bit]. Therefore, a calibrated measurement chain (calibrated measurement microphones, microphone preamplifiers, etc.) is required and needs to be supported by the sound design toolchain.

The AHU needs to be capable to synthesize and playback a suitable test signal needs to be available on the DSP-unit of the AHU for transfer function measurements/system identification (multi-sinus test signal).

Moreover, the test signal needs to be arbitrarily assignable to the supported output channels of the head unit in order to drive the corresponding loudspeaker systems which are utilized for ESE.

The amplitude response data needs to be storable in a suitable format in the transfer function dataset resp. the sound calibration file of the measured vehicle.

The upper and lower limits of the transfer function's frequency-axis needs to match to the supported frequency range of the sound synthesis algorithm.

Required Frequency Resolution: $\Delta r_t \leq 5$ Hz.

3.9.4 Transfer Function Data Format - Phase Response

The phase response data need to be storable in a suitable format in the ESE sound calibration file.

Required Frequency Resolution: $\Delta r_t \leq 5$ Hz

3.9.5 Transfer Function Data Format - Complex Number Format

Alternatively, the transfer function data of the secondary paths needs to be storable in complex format

Required Frequency Resolution: $\Delta r_t \leq 5$ Hz

Figure 21: Vehicle Cabin: Example of a transfer function measurement setup. The secondary path transfer functions are measured for each loudspeaker path (acoustic source) and various microphone positions (head rest driver and co-driver - front position). The microphones are for measurement purpose only and are not a part of the final PS-feature in mass production.

3.10 Sound Localization & Sound Staging Control

The perceived spatial sound image - based on the synthesized ESE-sound signals - needs to be controllable to in terms of auditory localization effects and phenomena (spatial resp. directional hearing) inside of the vehicle cabin.

The ESE-SW-algorithm resp. the PS-feature needs to support suitable signal processing countermeasures for a plausible and natural sounding presentation of the ESE-sound. Therefore, the PS-feature needs to support specific parameters to control the spatial impression of the synthesized ESE-sound, such as perceived direction and perceived source width (auditory source width). This can be accomplished by using the following control parameters (inter-channel differences, see

Table 8).

All parameters need to be assignable and to be applicable to each output stream of the PS-feature.

In addition, these parameters can be supported within the loudspeaker/cabin equalization algorithm of the PS-feature (see Section 3.8.1 for further reference).

Moreover, these parameters need to be accessible and editable within the sound design tool SW-application (sound design development software - GUI).

Each output channel of the PS-feature (stream output outputs) needs to support the described parameters.

3.10.1 Loudspeaker Localization Control – Tuning Parameters

Localization issues are given, if the presented ESE-sound contribution is localized in the same position as a single loudspeaker system (e.g. single door woofer chassis) or if the sound image is not balanced in the horizontal plane (e.g. the sound source is localized in outboard position). Figure 22 explains loudspeaker localization issue for a stereophonic setup. The described phenomena are also valid for multichannel loudspeaker setups.

Localization issues bear the risk that a single loudspeaker is perceived as an *isolated* sound source, which is undesirable in terms of a *natural* and *plausible* powertrain sound presentation.

The described issue holds for typical driving scenarios, e.g. that the subject's head is located near to the seats head rest position).

Suitable control parameters (see

Table 8 for details) allow to change the perceived 'sound stage' resp. 'phantom source' of the PS-sound, so that the passengers inside of the vehicle cannot determine that the sound is emitted by a single loudspeaker system resp. cannot isolate a single loudspeaker as a sound source.

The PS-SW-algorithm resp. the PS-feature needs to support suitable signal processing countermeasures to resolve loudspeaker localization issues for a plausible and natural sounding presentation of the presented PS-sound components.

Therefore, the PS-feature needs to support specific parameters to control the spatial impression of the synthesized PS-sound, such as perceived direction and perceived source width. This can be accomplished by using the following control parameters:

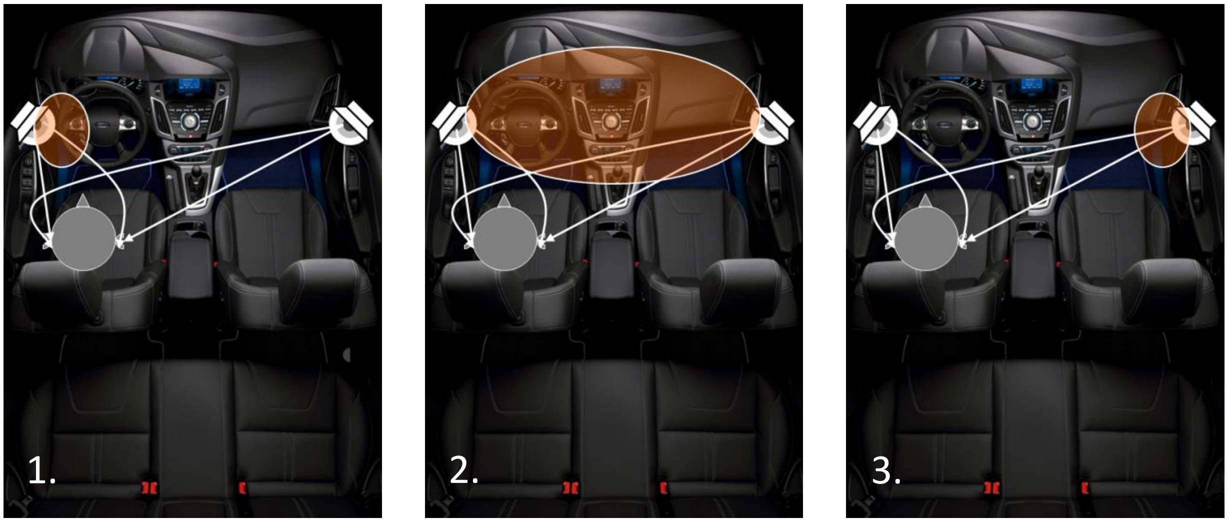


Figure 22: Examples of different sound image percepts in case of stereophonic presentation: 1: unbalanced sound image (focused sound image is localized in loudspeaker position (left loudspeaker channel – front position)), 2: well-balanced and wider sound image, 3. unbalanced sound image (focused sound image is localized in loudspeaker position (right loudspeaker channel – front position)). The highlighted area (red-colored oval) visualizes the perceived sound image from the driver's resp. passenger's perspective. The location and spatial extent of the perceived sound image can be controlled by manipulation of the inter-channel differences of the loudspeaker signals

Parameter	Unit	Ordinate Range (Min. Value)	Ordinate Range (Max. Value)
Phase response function	Φ in [°] or [rad]	-180° or -π rad	+180° or +π rad
Signal polarity/pol. inversion	-	-1	+1
Static time/sample delays (delay line buffer)	T in [sec]	0 ms	10 ms
Gain response function	dBFS	-96 dB FS	0 dBFS
Static Gain Factor - Output Channel	dBFS	-96 dB FS	0 dBFS

Table 8: Inter-channel parameters in order to control the perceived spatial sound image of the ESE-sound.

3.10.2 Sound Staging & Imaging Control

The PS-feature needs to support functionality to apply suitable signal processing algorithms in order to modify and to control the perceived source width and the perceived envelopment of the sound components.

The parameters need to be applicable to the supported output streams of the algorithm.

For instance, a decorrelation algorithm can be applied to the supported output streams resp. to the presented loudspeaker signals.

The optimization of the decorrelation parameters shall be based on the electro-acoustic transfer function data resp. impulse response data.

The transfer function resp. impulse response data shall include the compete audio signal path and thus reflect the system behavior of the electric and acoustic signal path(s).

The decorrelation parameters shall be calculated and optimized automatically or manually.

The optimization parameters need to be manageable and to be modifiable within the tuning tool SW-application.

The supplier may further make proposals on alternative methods and algorithms to control the perceived source width and the perceived envelopment.

3.11 Tuning Tool SW-Application and Sound Design Development

In the following sections the required capabilities and features of the tuning tool SW-application will be described.

The supplier has to provide a suitable tuning tool SW-application to Ford Motor Company. The existing tuning tool will be used as a basis – if suitable.

In case that additional functionality is needed within the tuning tool in order to support the requirements related to the PS-feature - as described within this document – then this must be implemented by the supplier.

The tuning tool needs to provide suitable interfaces to communicate with the active sound feature on the host system and other functional blocks as depicted in Figure 11 and Figure 28. Further, the tuning tool must support the following connectivity and functions:

1. Tuning Support: Download of binary tuning-datasets to ECU-flash/non-volatile memory
2. Industrialization Support: export of binary tuning datasets to be further processed in SW-release-work streams (product development gateways/mass production deployment)
3. Tuning Support: 'Live'-Tuning: transfer tuning parameters to volatile memory/RAM during runtime/tuning work streams

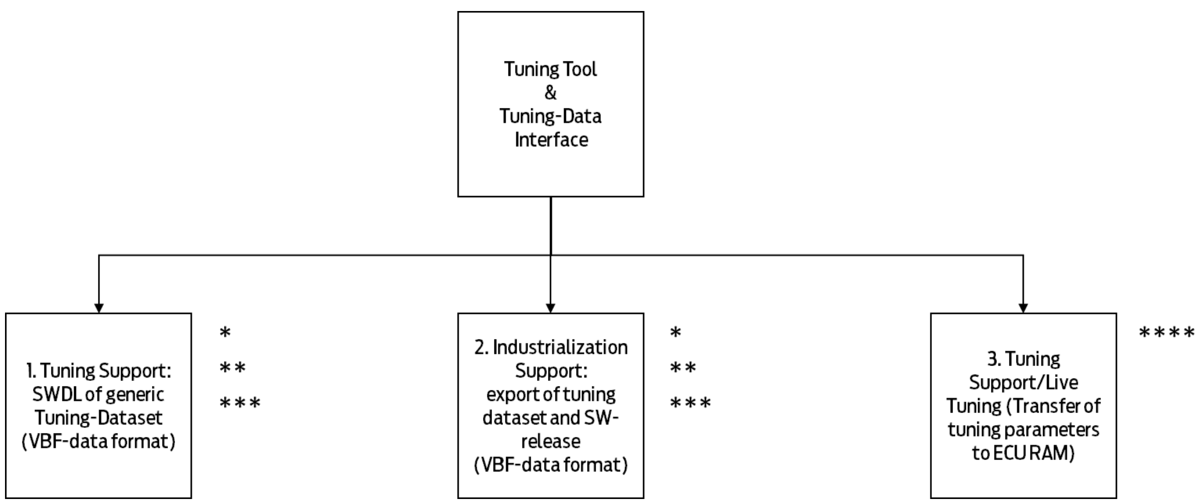


Figure 23: Tuning Tool – Tuning Interface and connectivity

Additional Comments:

*requires usage of a binary data format (ECU-Software Download)

**Branch 1 and branch 2 will probably require a consistent process in terms of tool resources (dataset export and post-processing).

***Content of tuning-datasets/calibration files and industrialized datasets/calibration files may slightly differ to additional meta data required for industrialization purpose.

****'Live'-tuning will probably require a data interface to tunnel all tuning parameters to the host system and to the application resp. active sound feature.

3.11.1 User Interface

The tuning tool needs to support sound development tasks from early sound prototyping ideas to mass production level. The following figures show examples of user interface design. The supplier must make design proposals for a user-friendly user interface and workflow supported by the tuning toolchain.

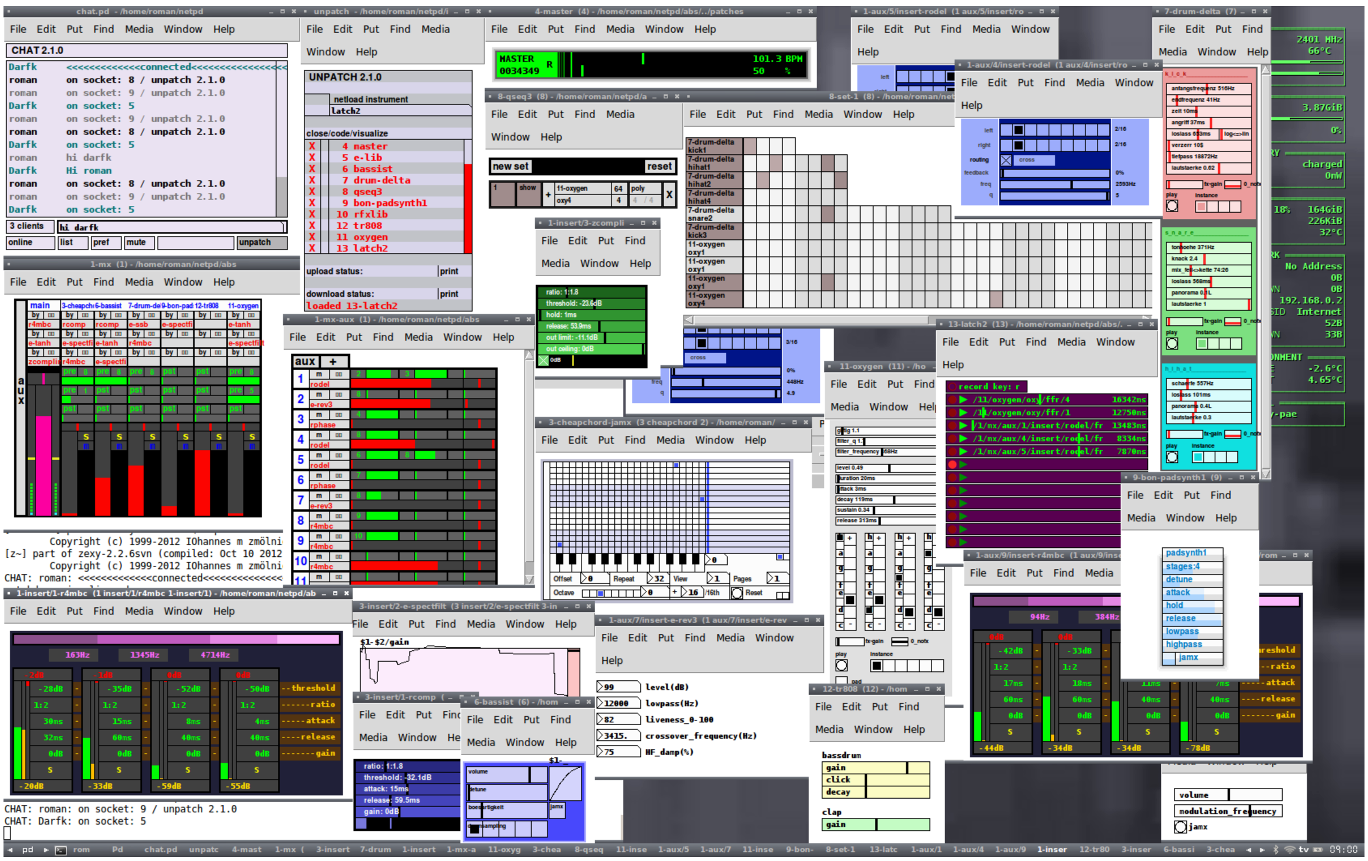


Figure 24: Tuning Tool - User Interface – Example – 'Pure Data' - Audio & Music Production SW-Application (courtesy of Wikipedia.org). The tuning tool must support capability to create customized elements of the user interface, such as dialog windows, table and plot widgets in order to edit and manage tuning parameters.

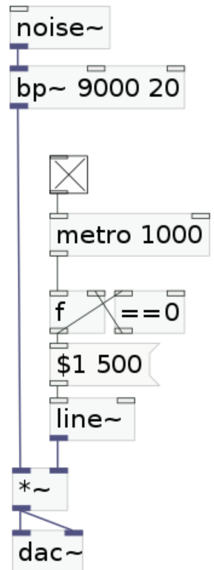


Figure 25: Tuning Tool - User Interface – Example – 'Pure Data' - Audio & Music Production SW-Application – visual data flow programming language - example on configurable DSP-objects ('Patcher') and data- resp. signal flow (courtesy of https://en.wikipedia.org/). The DSP-objects must be manageable within the PS-tuning tool SW-application. The data resp. signal flow must be editable in such a manner that DSP-objects can be connected with each other ('patched together'). The signal flow is represented by the graphical objects ('widgets') within the user interface of the tuning tool SW-applications.

Figure 26: Tuning Tool - User Interface – Example – Multiband Parametric EQ (courtesy of https://calf-studio-gear.org/)

3.11.2 Tunable Items

The tuning tool must support access, storage and modification of all sound synthesis parameters stored within the sound datasets and the parameters supported by the PS-DSP-algorithm.

3.11.3 Sound Design Project, Data Objects and Database-Manager

The tuning tool SW-application must support database functionality to ensure a use-friendly and efficient usage of tuning parameters and related data.

The tuning-parameters must be organized as data objects within the tuning database and within ‘project-projects’. The tuning projects must be managed within the tuning database.

The tuning tool SW-application needs to provide the following capabilities in order to manage tuning data in a user-friendly and efficient manner (‘database manager’):

- Object-oriented data management approach
- Data is organized within tuning projects
- Tuning projects contain different types of tuning-related data objects, e.g.
  - look-up table data
  - Configuration data (DSP-objects)
  - Transfer Function Data
  - Impulse Response Data
  - Meta data (e.g. descriptive data in order to identify sound design iterations)
  - EOL-Test reference datasets
  - And others

The tuning tool SW-application must provide capability to manage data objects within the database and within the tuning projects:

- Version control
- reuse of data objects across different sound design projects
- duplication
- deletion
- import
- export
- locking of data objects
- object query
- …and other capability that supports a efficient work flow

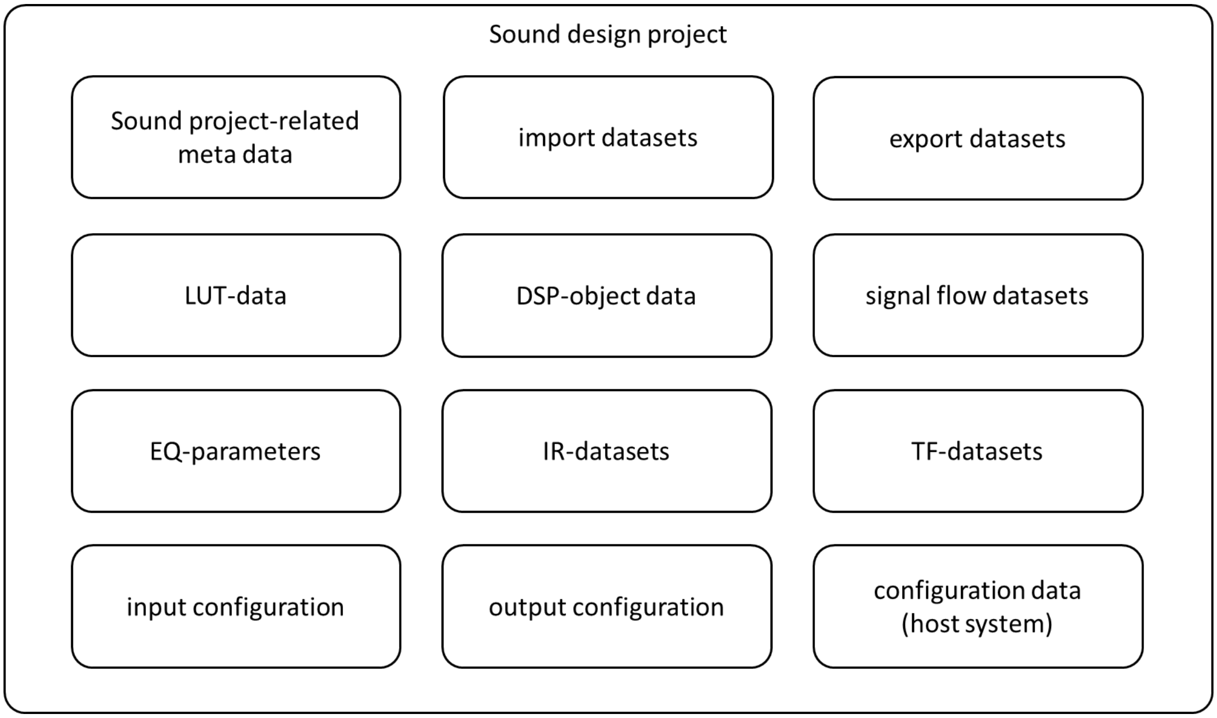


Figure 27: Sound design project and contained data objects. The objects must be manageable within the database file managed by the tuning tool SW-application.

3.11.4 System Status Monitor

The tuning tool needs to support to monitor the internal system and health status of the active sound feature during execution (System Status Monitor). The status monitor needs to be a functional block of the tuning tool. The following status information needs to be monitorable:

- initialization status
- input-signal status
- output-signal status
- system fault & defect status
- sound profile & dataset status
- instantaneous behavior of control- and audio signals (e.g. level meters and further types of signal analysis)
- other relevant information required in terms of tuning and diagnosis tasks and feature execution

3.11.5 Functional Blocks & Interfaces

Table 9 describes the structure and purpose of the involved functional blocks. Some functional blocks are an integral part of the active sound feature, while some potentially exist on the host-side, but whose need to interface with the active sound feature.

Functional Block:	Purpose:
Vehicle Interface	transfer of control signals (vehicle sensor and system data - vehicle operating conditions)
Customer Interface	transfer of customer input signals and acknowledged feature status (HMI)
Tuning Interface	supports communication with tuning-related functional blocks and tuning tool
Tuning Tool	develop sound sound synthesis parameters and sound datasets
Tuning Data (LUTs)	interface with volatile/non-volatile memory
Tuning Data (Audio)	interface with volatile/non-volatile memory
OTA-Interface	interface to support OTA-updates (SW-download) of sound profile datasets
Status Interface	transfer data related to diagnostics, configuration data and/or EOL-SW-download
Audio Component Interface	transfer output streams (sound synthesis) further downstream to loudspeaker
Application Control Proc.	SW-abstraction layer responsible to manage, monitor and control the active sound feature
Sound Synthesis Algorithm	core-algorithm on DSP-subsystem; real-time low-latency audio processing

Table 9: Functional blocks and Interfaces: Type and Purpose. The connection of the functional blocks and interfaces is depicted in Figure 28 (boundary diagram).

3.11.6 Active Sound Feature - Boundary Diagram

The following boundary diagram describes the involved functional blocks and related HW/SW-interfaces from a high-level point of view.

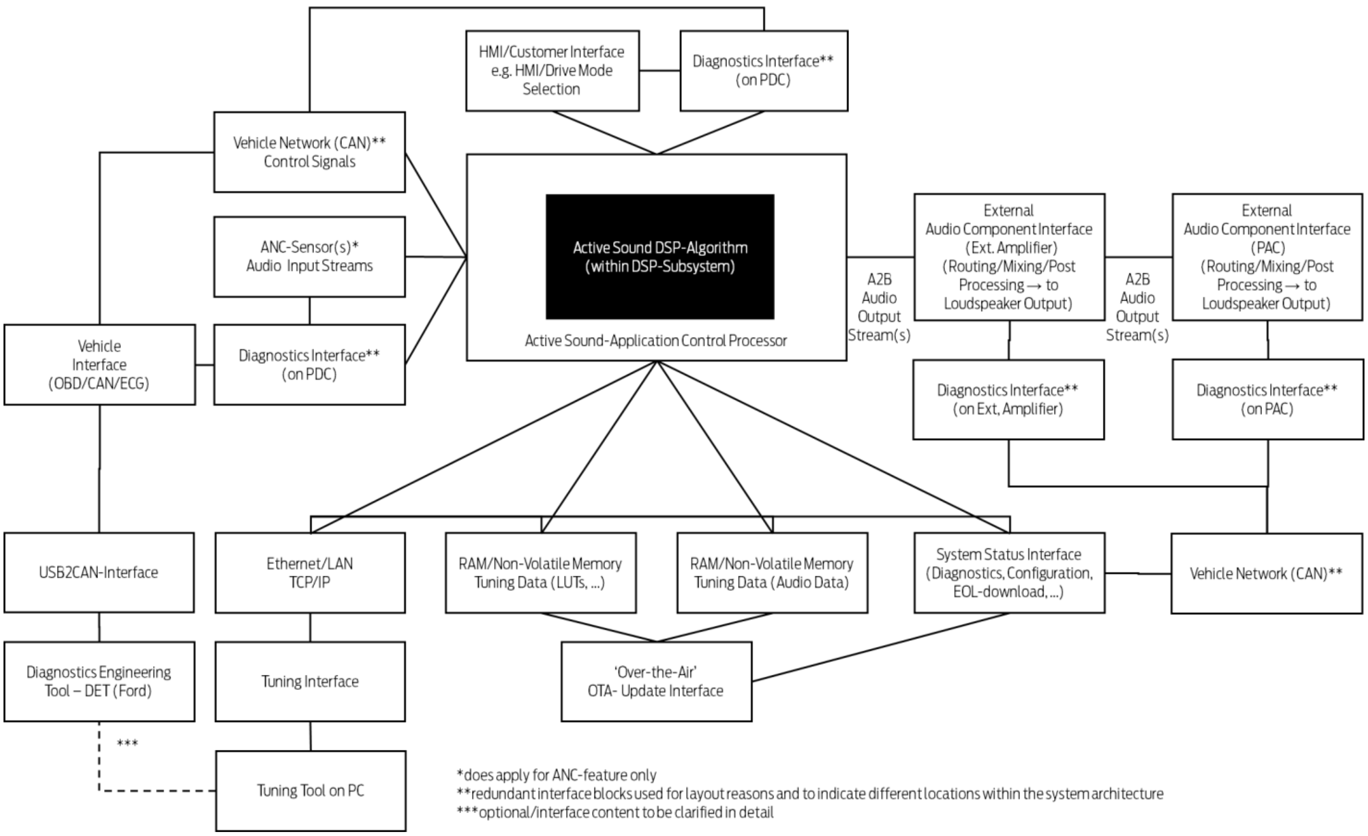


Figure 28: Boundary Diagram: High-level functional blocks and related HW/SW-interfaces (DRAFT).

3.11.7 Tuning Tool SW-Application - Online Communication Support/Tuning Control

The connection between the tuning tool application and the target resp. the PS-feature shall be bi-directional, so that data can be transferred from the tuning tool to the target and from the target to the tuning tool application.

Required tuning features:

- Direct transfer of tuning parameters to RAM during system operation (allows online revision of ANC-tuning-parameters for development purpose)
- SW-download of tuning profile(s) to system flash on the embedded target platform
- Online-deactivation of feature operation: ANC and ESE/EVSE-activation/deactivation need to be controllable independently from each other (ANC on/off, ESE on/off). Furthermore, both features need to be controllable simultaneously (ANC and ESE-activation linked)
- Monitor Sensor-Input- and Output-Signals (rms-audio levels for each input and output channel)
- Monitor Control Input-Signals (CAN-Signals, etc.)
- Trigger Transfer Function Measurement Routine, fetch/pick-up transfer function data
- Trigger system self-test routines, fetch/pickup test results and diagnostics results
- Diagnostics read-out (DTCs, ANC-Failure Modes, Test Signals, etc.)

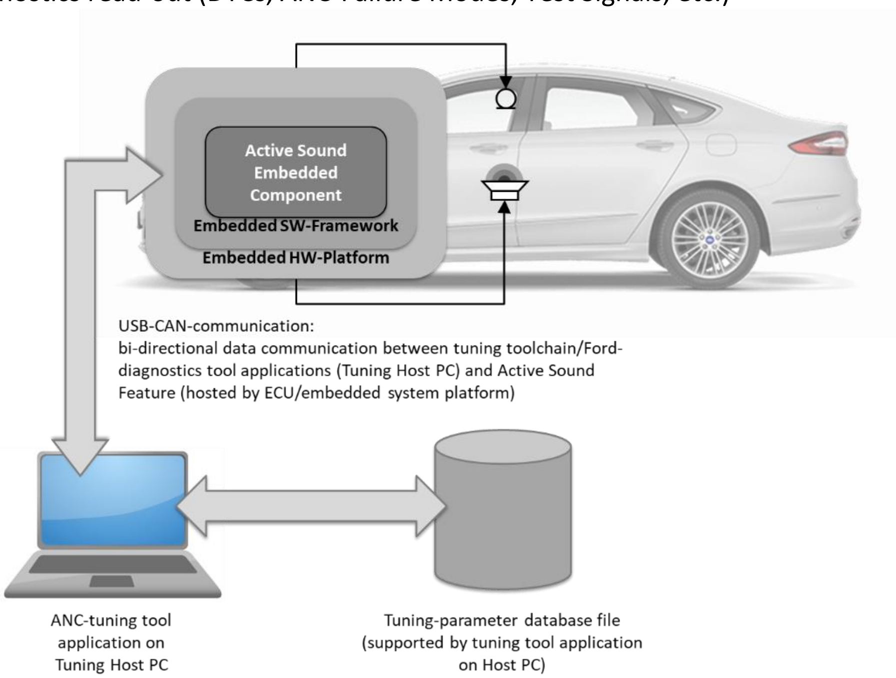


Figure 29: Communication between Tuning Toolchain-application (Host PC) and ECU (embedded system platform). Different HW- and SW-abstraction layers are shown additionally.

3.12 Automatic Tuning Capabilities

The PS-ecosystem needs to provide test methods and test routines in order to enable and execute automatic PS-tuning tasks. The PS-ecosystem needs to be capable to modify and optimize tuning parameters by using a target reference resp. a reference dataset (e.g. interior sound target curves or reference interior noise data). The PS-feature needs to provide capabilities to adapt the PS-parameters in such a manner that the sound response meets the target reference.

The PS-feature needs to provide suitable data acquisition resp. measurement-, data analysis and processing-capabilities in order to create the automatically tuned PS-profile.

The automatic tuning feature needs to consider the following tuning parameters to execute PS-profile tuning:

- Target sound pressure level and phase
- Spectral characteristics
- Temporal characteristics
- Spatial characteristics of the synthesized propulsion sound
- Control-related parameters
- Audio-related parameters
- Dynamic parameters

The tuning iteration process must be interruptible or stoppable as soon as the sound response on the tuning vehicle meets the reference target resp. is in a a tolerance-band of the target reference value. The tolerance band needs to be freely configurable within the tuning tool SW-application (within reasonable and supported ranges).

The tuning tool SW-application must provide a suitable user interface and features to support tuning tasks, data analysis, data visualization and system control.

The PS-ecosystem needs to support external peripherals for data acquisition (NVH-measurement system, audio-equipment/multichannel soundcard - compatible with Microsoft Windows 10operating systems).

The PS-Ecosystem must support data transfer between the PS-DSP-algorithm, target/host system and the PS-tuning tool SW-application (Tuning Host PC).

3.13 Rapid Prototyping Capabilities

The PS-ecosystem needs to support HW-devices and SW-tools with rapid prototyping capabilities in order to support pre-program phases, early vehicle prototype phases and/or generic tuning studies (e.g. on Pre-DCV-vehicle level, Ford-demo vehicles). The capabilities of the rapid prototyping system (RPS) need to be superior or at least identical with the capabilities of the PS-embedded feature on Ford embedded target platform(s).

The RPS has to provide a high flexibility in order to create various scenarios of PS-setups by providing a free configurability of audio and Control signal paths, input- and output-connections, control signals and tuning parameters.

Furthermore, the RPS needs to be integrated within the Supplier's PS-ecosystem resp. tuning tool chain framework. Here, the PS-tuning tool chain, SW-features and –interfaces need to be compatible for the RPS and the PS-embedded ecosystem, so that the NVH-engineer(s) can stay within the Supplier's toolset with a consistent work flow for later tuning iterations on Ford embedded platforms (e.g. on Ford DCV-, TT-, PP- and MP1-vehicle prototype level).

The rapid prototyping system and the program assumption PS-embedded ecosystem need to be compatible in terms of the all parameters required to develop a fully functional PS-tuning:

- amount of supported DSP-resources and interfaces
- tuning projects
- tuning parameters
    - adaptive filter parameters
    - secondary path transfer function data
    - PS-failure mode management

- look-up tables
- tuning meta data

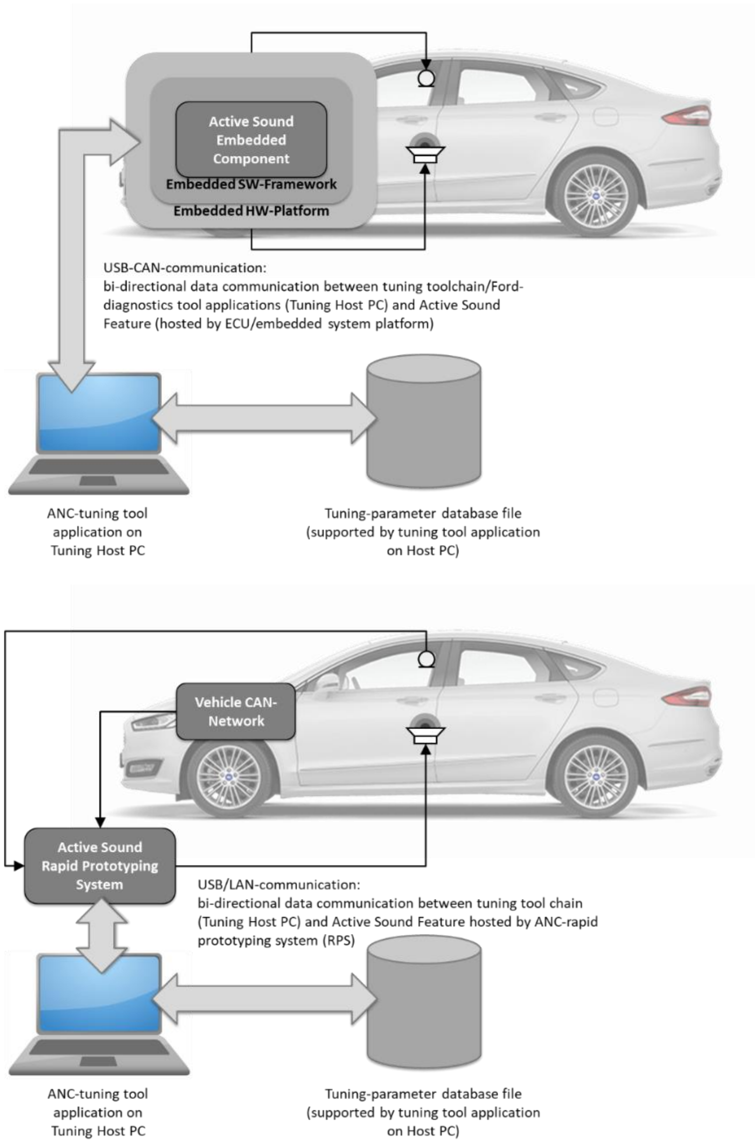


Figure 30: Top: Setup of the PS-ecosystem on embedded target (connections over the vehicle network are not shown here); Bottom: Setup of the PS-rapid prototyping system (RPS).

3.14 HIL-Support

The PS-ecosystem needs to provide capability to support development and system tests in a HIL-setup. Therefore, the system needs to provide functionality to be integrated in a HIL-setup with full functional features.

The PS-embedded ecosystem has to support the following tuning and test scenarios:

- (semi-)automatic tuning-support to develop PS-tuning parameters (in-vehicle and on bench)
- Interface tests
- system debugging
- system diagnostics

Furthermore, the PS-ecosystem and tool chain need to support tuning-/test cases based on virtual scenarios and virtual data, e.g. for simulated and/or calculated vehicle interior noise scenarios and simulated resp. calculated PS-dataset content (in particular PS-secondary path transfer function data, system and component level data (sensor and actuator data)).

3.15 Feature & System Diagnostics

The PS-feature needs to support the following diagnostics functionality in order to detect failure modes and to resolve error states.

3.15.1 PS-Setup Diagnostics – Test Methods - Manual Test Scenario

The PS-feature needs to provide test methods that allow to detect, indicate and resolve issues of the PS-setup on vehicle. Here, the PS-feature needs to provide test signals with freely configurable output level and frequency. The test signals need to be configurable for each output channel separately.

Furthermore, the PS-feature needs to support test methods on the input-side of the PS-feature (sensor-inputs). Here, the PS-feature needs to provide methods in order to detect indicate and resolve a malfunctioning or misconfigured PS-setup on vehicle.

3.15.2 PS-Setup Diagnostics – Test Methods – Automated Test Scenario

The PS-feature needs to support an automatic test routine in order to check for input- and output channel configuration. This test routine needs to support the detection of a malfunctioning/defective PS-feature resp. inconsistencies regarding to the actual PS-setup on vehicle.

3.15.3 Control Signal Diagnostics

The PS-feature must support diagnostics functionality to detect issues and failure modes with regard to the input-signals.

3.15.4 PS-Failure Mode-related Methods and Countermeasures

The PS-feature needs to provide functionality in order to detect, indicate and resolve failure modes resp. to apply suitable countermeasures.

Detection-, diagnostics-, indication- and issue resolution-methods of failure modes and system malfunctions must be supported by the PS-DSP-algorithm and the PS-ecosystem (including the tuning tool SW-application). Suitable countermeasures in order to avoid, solve and/or mitigate failure modes need to be supported by the PS-ecosystem.

The PS-ecosystem must support above methods for all available system interfaces and signal connections, such as

- all available system sensors (both audio and Control-related sensors)
- all available system inputs (both audio and Control related inputs)
- all available system outputs (both audio and Control-related outputs)
- all available system actuators
- additional communication paths – be it inside the PS-ecosystem between the relevant system abstraction layers
- algorithm internal failure modes

The PS-ecosystem needs to be capable to address failure mode countermeasures, such as

- muting/fade-out of audio input and/or audio output channels
- audio limiting methods both on input and output side
- control parameter hold or interpolation due to missing input signals
- intermittent and/or permanent feature deactivation

Execution of the countermeasure must depend on the severity of apparent failure modes.

The PS-DSP-algorithm must support different operation states that reflect above described failure mode-related functional content (automat state flow/state chart approach needs to be supported by the PS-DSP-algorithm). The PS-DSP-algorithm must be capable to switch between appropriate states according to detected/non-detected failure modes and feature initialization. The PS-component needs to provide suitable SW-interfaces in order to communicate with the host system to broadcast and to receive information on detected failure modes or malfunctioning system components.

Detected failure modes need to be broadcasted by the PS-DSP-algorithm to the tuning tool application as these must be clearly indicated to the tuning resp. test engineer PS-diagnostics monitor dialog/GUI).

Furthermore, all detected diagnostic parameters and detected failure modes need to be broadcasted via Ford's diagnostic interface in the format of a DTC (failure mode/issue indication) and as diagnostics messages consistent with the information indicated by the PS-tuning toolchain.

This functionality is covered within specification document 'Phoenix Active Noise Control/ Propulsion Sound Enhancement Functional Specification' (cf. 'References').

3.16 EOL-Programming and Test Routines

The following sections describe the feature requirements in order to support the vehicle production- and deployment process within Ford.

3.16.1 EOL-Programming & Deployment Support

Suitable methods shall be provided in the AHU and the supplier implementation to enable EOL-programming at Ford-production plants. The supplier shall not be required to provide storage of PS-parameters within the AHU except for the parameters programmed at Ford EOL.

The PS-toolchain needs to be capable of exporting raw data files (hex-format). In a second step, the hex-format files will be converted to a flash-able format (Ford vbf-format for EOL-download) by using a software tool provided by the PS-feature supplier (hex-to-vbf-conversion).

Ford shall be responsible for archiving and configuration management of EOL VBF files. ESE/EVSE and ANC shall be disabled by default. ESE/EVSE, ANC or both shall be enabled permanently by successfully running the appropriate activation routine.

License count due to ANC- and/or PS-feature activation will be carried out by using Ford's Ordering System (CMMS3-Tool).

The following license feature- and license-activation scenarios are possible:

1. ANC only
2. ESE/EVSE only
3. ANC and ESE/EVSE

It shall not be required that the ESE/ANC software executable program ("strategy") be downloaded at Ford EOL. Nevertheless, it is required that there is some provision for embedded target software program updates in the field.

The VBF for the DSP-executable and the VBF for the tuning calibration will be separate files from existing Ford program VBF files.

The ANC-ecosystem needs to support the following approaches and embedded target file systems for VBF-file generation:

1. file systems included in the VBF-file (payload and file system)
2. file systems excluded from the VBF-file (payload of tuning data only; for decreased file size)

In addition, the PS-ecosystem has to support suitable file compression/de-compression methods to further reduce time duration during EOL-file downloads in Ford-production plants/on Ford-production vehicles.

The expected tuning files size may not exceed a size of XX kB (1) and XX kB (2).

The vbf-file formats are described in specification document 'Versatile Binary Format Specification 3.1'

3.16.2 EOL Test Support and Issue Resolution

To provide an audit test capability for Ford production vehicles and production plants, the PS-feature shall be capable of performing a self-test (frequency response, by measuring a complex valued TF-matrix as a dataset on vehicle) using the vehicle audio infrastructure, the PS-sensors and the PS-actuators.

The purpose will be to check the correct operation of sensors, actuators and basic operation of the audio signal paths associated with ESE. This diagnostic is not required to test the performance of ESE.

Processing within the embedded target system (PS-embedded framework) shall automatically compare the measurements against reference values and threshold parameters stored in the AHU and report pass/fail - i.e. by comparing the actual measured response to a reference data set stored within the PS-calibration file.

All test results need to be broadcasted to the PS-tuning tool application (Tuning Host PC). The tuning tool application has to provide suitable analysis methods in order to indicate and trace functional issues of the PS-feature, e.g. such as malfunctioning/defective loudspeakers, microphones or signal paths.

Please refer to specification document 'Phoenix Active Noise Control/ Propulsion Sound Enhancement Functional Specification' for further information (cf. 'References').