Engineering Specification

xx MU5T-14G113-AA002

PART NAME: Engineering Specification – AVAS (Acoustic Vehicle Alert System)	PART NUMBER: xx MU5T-14G113-AA002

LET																							
FR																							
LET																							
FR																							

DATE	LET	SECT	REVISION	DR	CK	Engineer:
2020/04/20			Miscellaneous updates per discussion with phe5.			MSONNEN2
2020/04/03			Draft Release			MSONNEN2

Contents

Engineering Specification

xx MU5T-14G113-AA002

The information contained within this document Is Proprietary to Ford Motor Company

---

Disclosure or distribution to Unauthorized persons outside of Ford Motor Company is Strictly prohibited

---

1 Overview

This document outlines the signal flow and signal processing requirements for the Ford Audible Vehicle Alert System (AVAS). All hybrid and battery electric vehicles operated in the United States are mandated by the National Highway Transportation and Safety Administration (NHTSA) to have an audible alert at low speeds as described in Federal Motor Vehicle Safety Standard 141 (FMVSS 141) as of September 2020. Similar regulations have been adopted by global markets such as Europe ECE 138-01 and China GB/T 37153-2018. The signal processing requirements outlined in this document will allow the Ford AVAS system to meet these requirements as well as enhance the customer driving and ownership experience.

This document is not intended to cover functional software and hardware requirements that pertain to all Ford modules. For this information, reference the following functional specs:

Hardware: **ES-HU5T-14G113-A***
Software: **FS MU5T-14G113-A***

Should conflicting information exist between this document and a functional spec, the functional spec takes precedent.

Any feature listed as "**Preferred not required**" is not considered a formal requirement for the AVAS Signal Processing specification; however, additional consideration will be given to signal flows that incorporate these blocks and/or elements.

This document assumes the reader understands basic audio and digital signal processing principles; however, some definitions and their relevance to this document are outlined in section 2.

2 Definitions

- **Acoustic Vehicle Alert System (AVAS)**
- **Audio block** – refers to an audio function that outputs a generated or modified audio stream(s) per an audio or non-audio inputs (e.g. sample playback block, FIR filter block)
- **Audio element** – refers a single audio source within an audio block (e.g. a single audio sample within the sample playback block).
- **Audio plugin** – is a self-contained code and UI that can be "plugged in" to a DAW or audio application to add functionality to the original application. Typical plugins include equalization, reverb, delay, and virtual instruments.
- **Audio sample** – refers to a recorded source of audio used for playback. Sample could refer to various codecs including *.wav* and *.mp3*. "Sample" may also refer to a single data point in a discrete-time audio signal; however, this document will primarily use "sample" to refer to a recorded audio source of finite length.
- **Controller Area Network (CAN)** – vehicle bus standard designed to allow microcontrollers and devices to communicate with each other without a host computer.
- **Digital Audio Workstation (DAW)** – digital software application used for recording and editing digital audio.
- **Equalizer (EQ)** – hardware or software filter that accentuates or attenuates individual frequency bands of an audio signal.
- **Filter** – Infinite Impulse Response (IIR), Finite Impulse Response (FIR) – basic signal processing element that modifies frequency content.

AVAS
FIR
" "DAW

- **Noise** – Audio signal that has equal energy across all frequencies in the narrow-band (White) or octave-band (Pink).
- **Non-Audio Input (NAI)** – refers to any non-audio signal in the AVAS application used for function control. These will typically be CAN bus signals such as Vehicle Speed and Pedal Position.
- **Octave Band (1/n)** – a group or "band" of frequencies that doubles in frequency every "n" bands. Refer to regional AVAS regulations precise definitions of specific octave bands.
- **Oscillator** – a pure sine wave / tone.
- **Sample Rate / Sample Frequency** – discrete-time property of a signal that defines how many individual data points (or samples) are contained within a one-second window. The sample rate defines the Nyquist Frequency (half the sample rate) which is the upper end of the useable frequency bandwidth above which aliasing occurs.
- **Sound Pressure Level (SPL)** – widely used metric for characterizing the sound level in decibels (dB) as defined by $20*\log_{10}(P/P_{ref})$.
- **Synthesis** – refers to generation of sound through various methods including playback of an individual sine tone or complex playback and manipulation of multiple waveshapes and/or samples. Common methods of sound synthesis include Additive, Subtractive, Frequency Modulation (FM), and Granular Synthesis.

3 Audio Processing Specifications

- Reference functional specs ES-HU5T-14G113-A* and FS MU5T-14G113-A*.
- Sample Rate: >= 24 kHz
- Bit Depth: 16 bit
- MIPS (Million Instructions per Second): XXX.X
- Global Smoothing Parameters
    - Smoothing function should be used when transitioning from one sound / mode to another (e.g. park (no sound) to drive, drive to reverse).
- Bootup time to sound: reference functional specs ES-HU5T-14G113-A* and FS MU5T-14G113-A*.

4 Signal Processing Blocks

This section outlines the functions or "blocks" used within the signal flow. This section is divided into two sub sections – *Signal Generation* and *Signal Modification* – which pertain to the creation and alteration of all sounds used in AVAS playback.

4.1 Signal Generation

Signal Generation refers to the creation of audio either through playback of audio samples or other forms of digital sound synthesis (e.g. additive, subtractive, granular). Each signal generation block must have independent pitch and level control via non-audio input signals. At a minimum, each signal block must have independent pitch vs vehicle speed, pitch vs pedal position, level vs vehicle speed, and level vs pedal position. Some elements within a given block require independent pitch and level control such as unique audio samples as outlined in section 4.1.1.

4.1.1 Sample Playback

- Minimum of 4 unique audio samples
    - 24 kHz (per section 3 Audio Processing Specifications)
    - 16 bit (per section 3 Audio Processing Specifications)
    - 5.0 sec maximum sample length
- Independent sample lengths. Not every sample must be the same sample length.

- Simultaneous playback
- Independent pitch and level control via non-audio inputs (e.g. Vehicle Speed, Pedal position)
- Configurable loop start/stop/duration. Reference Figure 4.1.
- Unique IIR/biquad filter per sample (e.g. 4 unique samples = 4 unique filters)
- Preferred not required
    - Loop type selection (Forward Loop, Forward Backward Loop)

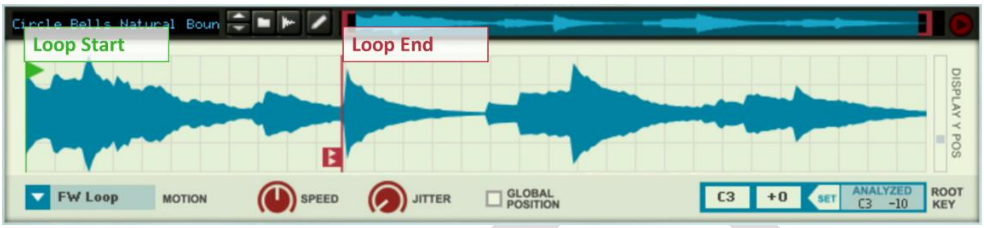


Figure 4.1 Example audio sample with configurable loop start and end (Reason Studios Grain)

4.1.2 Oscillators

- Minimum of 32 individual oscillators
- Independent pitch and level control via non-audio inputs (e.g. Vehicle Speed, Pedal position)
- Configurable phase (0-360 or -180 to 180 degrees)

4.1.3 Noise

- Configurable pink/white noise source.
- Independent level control via non-audio inputs (e.g. Vehicle Speed, Pedal position).
    - Pitch control not applicable for noise function.
- Minimum 2 unique IIR/biquad filters.

4.2 Signal Modification

Signal modification refers to the modification of existing audio signals that were previously generated in the signal flow. Examples of basic signal modifications include filters (IIR and FIR), delay, reverb, etc. The minimum required signal modification blocks are listed below; however, additional blocks may be considered and/or requested if feasible with the associated AVAS system hardware.

4.2.1 Infinite Impulse Response (IIR) Filter

- Minimum of 10 IIR/biquad 2nd order filters at various points in signal flow
    - Samples (x4)
    - Noise (x2)
    - EQ (x4)
- Configurable filter type:
    - All pass
    - Low pass
    - High pass
    - Band pass
    - Band stop
    - Peak/Notch

 - Low shelf
 - High shelf
- Preferred not required:
 - Option to choose higher order filter (4th, 6th order)
 - Ability to control filter coefficients with NAIs (e.g. filter cutoff adjusts with Vehicle Speed).

4.2.2 Finite Impulse Response (FIR) Filter

- 512 taps minimum
- Must include the following methods for selection of filter coefficients:
 - GUI – Generate filter coefficients per user drawn target response.
 - Table – Generate filter coefficients per user entered target response.
 - Import – User imports filter coefficients directly.
- Must display magnitude and phase of target response (user input) and actual response. If user imports filter coefficients, must only display actual response. Reference Figure 4.2.

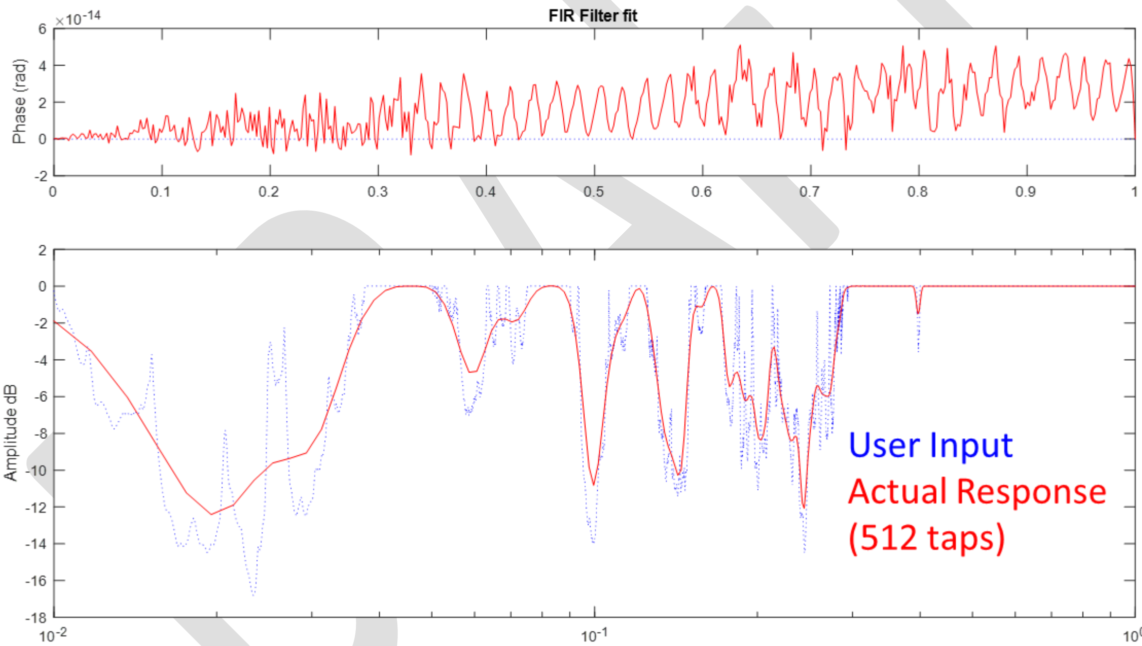


Figure 4.2 Example FIR filter user input and actual response (via MATLAB).

4.2.3 (Preferred not required) 1/3rd Octave Graphic Equalizer

1/3 octave graphic equalizer that spans at least the legal AVAS bands if not the entire AVAS system frequency range. Reference Figure 4.3 for a graphic equalizer that highlights the low band group and high band group per FMVSS 141.

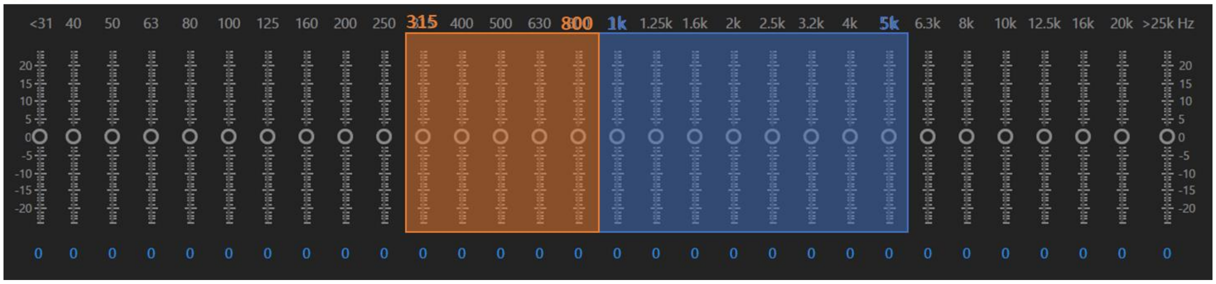


Figure 4.3 Graphic Equalizer with highlighted FMVSS legal bands (via Adobe Audition)

5 Signal Flow

Reference Figure 5.1 for an example top-level signal flow. This signal flow demonstrates the main audio blocks discussed in section 4 Signal Processing Blocks. CAN signals are represented by the red lines and audio by the black. The gain symbols (triangles) are meant to show level modification controlled by CAN signals. Note, while this is merely an example, **<u>the tuning tool must include a master gain function</u>**. Whether the other gains are in the top level or within an audio block is left unspecified; however, the example here shows them in the top level. Figure 5.2 shows an example Sample Playback audio block signal flow.

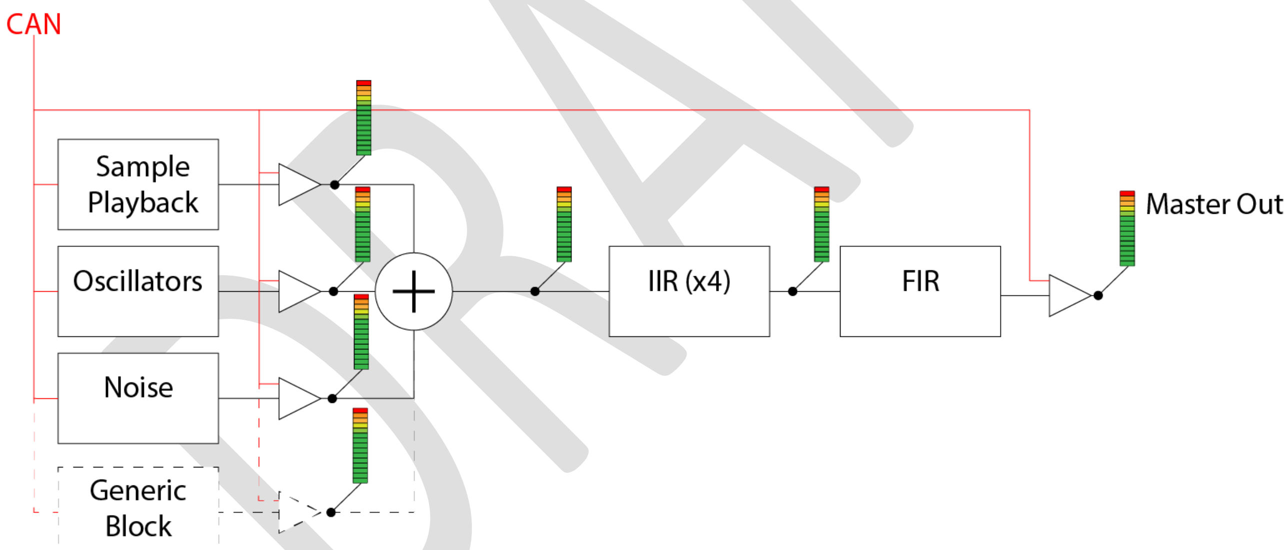


Figure 5.1: Example Signal Flow – Top Level

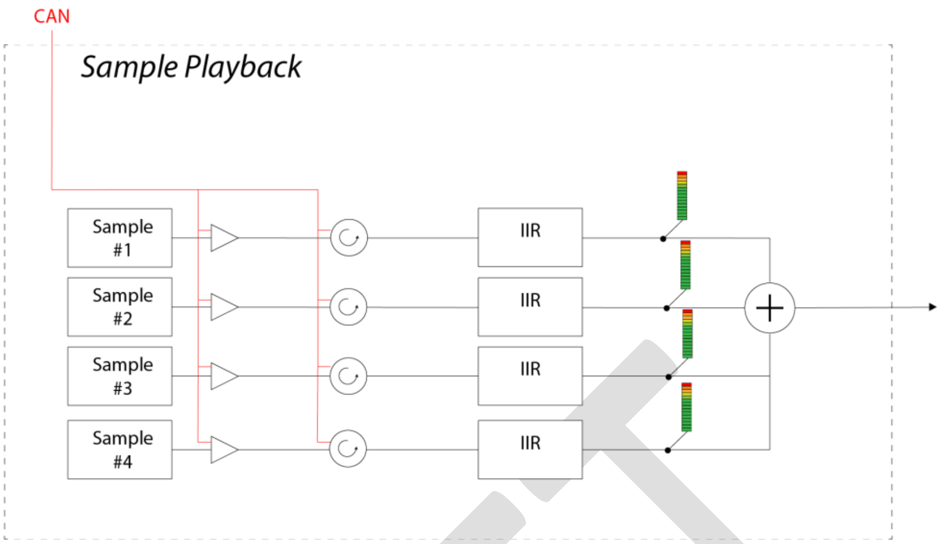


Figure 5.2 Example Sample Playback audio block signal flow.

6 Tuning Interface

Tuning Interface refers to the Graphical User Interface (GUI) by which the sound designer/engineer may tune, create, design, modify, tweak, and/or interface with the AVAS sound. Many existing programs rely on a tuning interface for sound production. The most applicable of these programs is the Digital Audio Workstation (DAW). The DAW is the main tool in almost any sound designer/producer's toolbox. There are numerous DAWs available ranging in quality and price; however, some of the industry standards are Ableton Live, Pro Tools, Logic, and Cubase. A common freeware DAW is Audacity. While these applications have unique specialties and use cases, they share many common features and can accomplish many of the same outcomes. Any good AVAS tuning interface should adhere to these common practices. This section will attempt to outline some of the most relevant of these practices.

Another notable sound tool is the audio Application Programming Interface (API). One example audio (and video) API is Max MSP by Cycling 74. This program helps create detailed audio signal flows and can be used to design independent applications such as an AVAS tuning interface. Ableton, which produces the popular DAW Ableton Live, acquired Cycling 74 in 2017 and offers a unique cohesion between the two softwares called Max for Live.

6.1 Audio Settings/Preferences

The AVAS tuning interface should allow for real-time simulation and playback of the AVAS sound as it would be fed to the in-vehicle system. The tuning interface should therefore use standard audio drivers for Windows 10 (e.g. WASAPI, DirectSound). Use of ASIO drivers is preferred, but not required. The tool should also allow use of standard audio sampling rates for playback including at a minimum 44.1kHz and 48kHz.

6.2 Mixer

The tuning tool should implement some type of mixer view that displays the level of all individual audio elements (e.g. a single audio sample) and/or blocks (e.g. sample playback block). The mixer view should incorporate standard features such as the ability to mute, solo, and adjust level of all audio elements and/or blocks. The mixer view should also include a Master level and any other important points in the signal flow (e.g. prior and after global EQ filters). Reference Figure 6.1 and Figure 6.2 for example mixer views.

Figure 6.1 Example Mixer Window (Ableton Live)

Figure 6.2 Example Mixer View (Pro Tools)

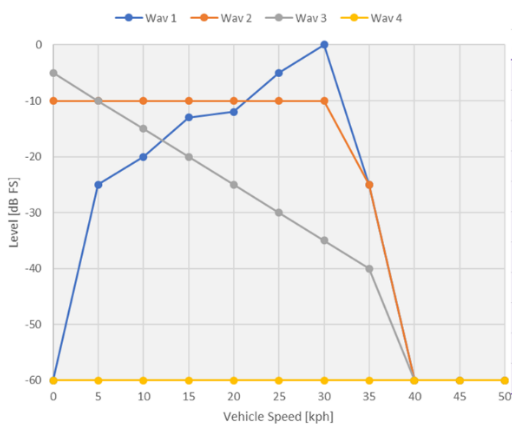
6.3 (Preferred not required) Timeline

The tuning tool should include a basic timeline/editor view that shows where playback is occurring in any given audio sample. This will feature only applies to user loaded audio samples (e.g. sample playback block) and not to digitally generated signals. This view should show the waveform of the audio samples and a cursor that moves through the waveform during playback. Seeing as playback may involve varying length audio files, each file should have an independent cursor.

6.4 Control Signal Interface

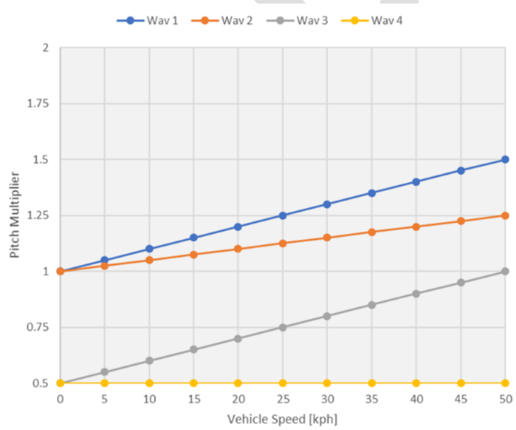
Each audio element and/or block must be controllable by at least Vehicle Speed and Accelerator Pedal Position NAI signals. The simplest method for this interface is an XY plot where the independent variable (X) is the NAI signal and the output is a controllable output of the audio signal such as pitch and level. Reference Figure 6.3 and Figure 6.4 which illustrate basic control parameters of the sample playback block. An acceptable control interface should have the following features:

- Graphical User Interface of input and output signal (XY plot)
- Table of input and output signal.
- Audio blocks should be configurable to show 1 or all elements on the same XY plot and table. Reference Figure 6.3 and Figure 6.4 that show all four audio sample control signals on a single plot and table.
- Linear interpolation between user input points.
- Vehicle Speed precision of 1 kph or less.
  - *** Note Figure 6.3 and Figure 6.4 **improperly** shows a precision of 5kph. This is meant only to be an example as it was obviously created in Microsoft Excel.
- Pedal Position precision of 1% or less (range from 0-100%).
- Pitch Multiplier precision of 0.001 or less.
  - Pitch modification algorithm should be smooth and have no noticeable discrete changes in pitch.



	Level dBFS			
Speed (kph)	Wav 1	Wav 2	Wav 3	Wav 4
0	-60	-10	-5	-60
5	-25	-10	-10	-60
10	-20	-10	-15	-60
15	-13	-10	-20	-60
20	-12	-10	-25	-60
25	-5	-10	-30	-60
30	0	-10	-35	-60
35	-25	-25	-40	-60
40	-60	-60	-60	-60
45	-60	-60	-60	-60
50	-60	-60	-60	-60

Figure 6.3 Example sample playback control interface – Level vs. Speed



	Pitch			
Speed (kph)	Wav 1	Wav 2	Wav 3	Wav 4
0	1.00	1.00	0.50	0.50
5	1.05	1.03	0.55	0.50
10	1.10	1.05	0.60	0.50
15	1.15	1.08	0.65	0.50
20	1.20	1.10	0.70	0.50
25	1.25	1.13	0.75	0.50
30	1.30	1.15	Plot Area 0	0.50
35	1.35	1.18	0.85	0.50
40	1.40	1.20	0.90	0.50
45	1.45	1.23	0.95	0.50
50	1.50	1.25	1.00	0.50

Figure 6.4 Example sample playback control interface - Pitch vs. Speed

6.5 Signal Analysis

In addition to basic level metering capabilities as described in section 6.2 of this document, the tuning tool must also include a basic spectrum analyzer. This may be a built-in spectrum analyzer or an external VST plug-in. Capability to allow for external VST plug-ins is preferred but not required. There are numerus commercial applications available for use with any DAW that supports VST plugins. One free spectrum analyzer plugin is Voxengo Span as shown in Figure 6.5. The spectrum analyzer should be implemented such that any point in the signal flow may be analyzed. In other words, anywhere there is a level meter, the capability for spectrum analysis should also be present.

Figure 6.5 Example VST plugin spectrum analyzer (Voxengo SPAN)

6.6 Tuning Interface Miscellaneous

This section attempts to cover any remaining items that must be considered in the design of the tuning interface. Items are listed below:

- All graphical interfaces and XY plots should allow for zoom / scaling features to enable more fine adjustments of control signals.
- Any plot that displays frequency along an axis should be able to display in both linear and logarithmic scale.
- The tuning tool interface should be created for Windows 10 (or future version of Windows) and take advantage of typical applications such as the ability to resize windows and snap to place in a desktop environment.
- The tuning tool should indicate the software version level.

6.7 File Save/Import/Export

The tuning tool must be able to save a current session file that can be opened in the application at a later date. The tuning tool must include an export feature for creation of vbf file(s) to be flashed into the AVAS module and used in the vehicle. The tuning tool must include an import feature to load previously exported vbf file(s).

7 (Preferred not required) Simulation and Hardware/Software In-the-Loop (HIL/SIL)

In order to reduce cost and development time, there is a considerable effort within the company to reduce the need for physical vehicle testing. Current AVAS methods rely heavily on a physical vehicle prototype for tuning legal sound and subjective sound quality. Preference will be given to any AVAS software package that uses HIL/SIL methods that allow for greater tuning efficiency and reduced need for physical testing. Some desirable features are discussed below.

7.1 Predicted legal status via measured vehicle transfer functions

One desirable AVAS software feature is the ability to predict legal status based on signal flow output and previously measured vehicle transfer functions. Transfer function measurements would be measured from the AVAS speaker location to the legal microphone positions for the various regional requirements. For example, FMVSS would include five microphone locations: Front Left, Front Center, Front Right, Rear Left, and Rear Right. The AVAS software should be able to use the signal flow output and pass this through the five individual transfer functions to the legal microphone locations and determine what the response would be at these locations. The software should know the latest legal level requirements and compare the calculated output to these values. The software should also allow for user input of unique target set in order to ensure sufficient margin above the legal limit.

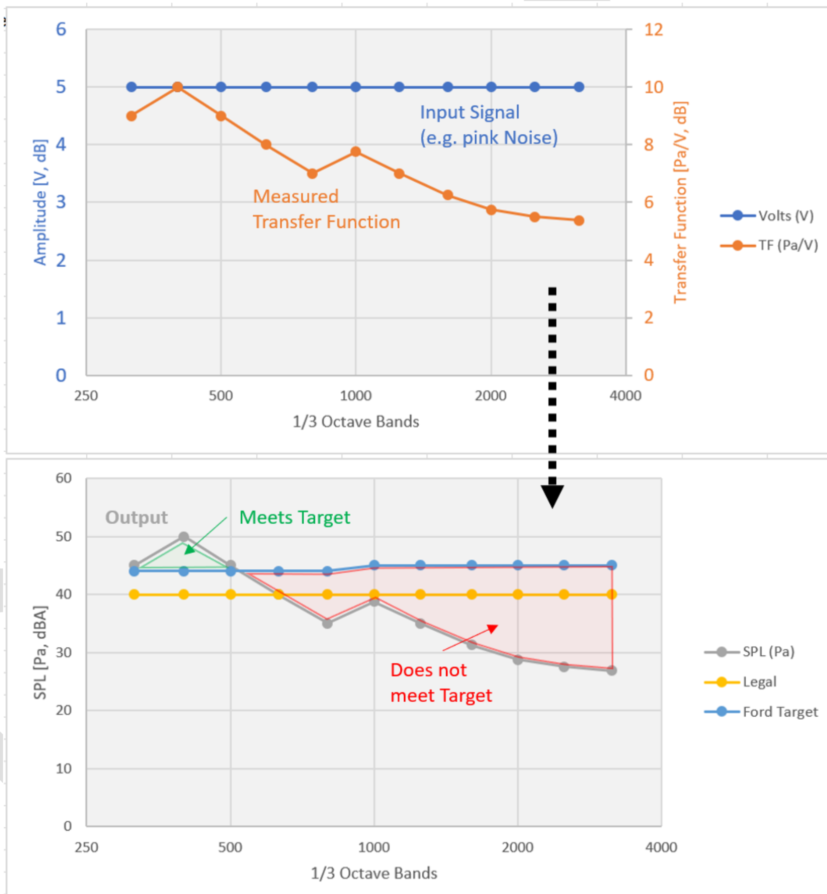


Figure 7.1 Example legal/target level projection.

7.2 Desktop playback via measured vehicle transfer function

Similar to prediction of the legal level using measured transfer function data as outlined in section 7.1, another desirable feature would be projected playback at the legal microphone locations and potentially all around the vehicle. As a start, this method would use measured transfer function data to project what the vehicle would sound like at various points around the vehicle. The first obvious points to use would be the legal location, but this method could be extended to include any locations with measured transfer function data. This method likely becomes more complicated than legal projection due to binaural audio concerns and other details that come into play for accurate simulation of human perceived sound.

8 Other

9 Appendices

9.1 Appendix A – Functional Spec Documents

- Hardware: ES-HU5T-14G113-A*
- Software: FS MU5T-14G113-A*