Sonic Tonic For Audio Coding

Frank Foti
Omnia Audio
Cleveland, Ohio

ABSTRACT

Achieving great sounding coded audio is easier said than done. What are the critical elements that set apart great sounding digital channels or streams, especially at lower bitrates? Improving codec performance is accomplished through innovative signal conditioning and processing means. Coded audio is now a way of life in the professional and consumer sound industry. It is commonplace in all forms of media that utilizes sound in one form or another. This paper is not written to sell the reader on the benefits of coding, or how it works. The writer assumes this to be a *given*, that the reader understands what audio coding is. (If not, you might consider moving onto the next paper in the proceedings!) The purpose here is to investigate what transpires within the coded transmission system, where the stumbling blocks are, and seek methods that provide smooth sonic sailing. This presentation provides detailed and comprehensive information regarding the causes of perceptable problems in audio coding, how to avoid them, along with a method that improves the sound quality of coded audio.

CODECS: THEY'RE HERE TO STAY

Codecs are common in practically every audio transmission system throughout the world: FM, AM, HD-Radio, DAB, DRM, television, multicasting, podcasting, netcasting, satcasting, and just about every form of *casting* you can think of. Quality sound, especially at low bitrates, requires comprehensive understanding of the coded system, as well as knowing what must be applied, prior, to audio content that insures consistent sound performance. It is more than just plugging sound gear together, configuring the applications and sha-zzam, great sound appears. For the codec naysayers out there, if you feel that Life gave us lemons, well, we're about to make lemonade!

PACKING THE PIPELINE

Data reduced audio systems have changed our world! With the rapid growth of ISO/MPEG Layer-III (MP3) and subsequent additional methods, the capability of transmitting multiple channels of audio is commonplace in this day and age. Within the data payload that once contained a single stereo pair, many stereo channels now exist. Almost seems like yesterday when high quality stereo feeds at 128kbps, via ISDN, were thought to be the best our world could ever expect. Typical of technology, the bar just kept rising and it continues to do so. Today, in the year 2007, a listener can experience high quality stereo, as well as surround, digital broadcasting at bitrates much lower than what we felt were the maximum a few short years ago.

This paper/presentation grew out of efforts to seek improved performance of coded audio at lower bitrates (24kbps – 48kbps). Critical listening to the performance of a new conditioning algorithm, designed to improve vocal intelligibility, revealed two significant results: voice reproduction was noticeably improved due to the new algorithm…but…the enhanced midrange uncovered and/or disclosed negative aural discolorations in the presence and high frequency range.

MOVING TARGETS

Since the early 1990's, audio coding has been around the professional sound industry. Codec developers have been on a fast track, and they continue to be. Audio quality, judged by MPEG (Motion Picture Experts Group) to be excellent at 256kbps and 128 kbps, are now offering the same judgment at bitrates much lower. As codecs improve payload efficiency, it becomes possible to add more transmission channels to the existing infrastructure. It's much easier to improve the data payload, as compared to expanding the pipe. This is how program services are able to expand their range of content offerings with additional channels.

To accomplish this requires using lower bitrate codecs. Lowering the birate increases potential degradation of audio performance. Advancement of codec design has allowed lower bitrates to be employed, and most codecs sound *decent* at these rates, but they are much more fragile with regards to distortion, and susceptible to artifacts. Due to the various types of codecs, and lower bitrates, makes getting a handle on the issues that annoy these functions a moving target, so to speak. The goal of this dissertation is to seek out the gremlins, then offer ways and means to avoid them. Getting under the hood and removing the veil of the codec itself is beyond the scope of this paper.

HISTORY

All transmission systems suffer from some form of problem, one way or another. Doesn't matter if the system is linear or not, suffice it to say that all of them have something to overcome. The simple phrase *no free lunch* applies.

The key to improving audio quality through a coded system is in understanding where the challenges are located, and what can be done to avoid causing them. Performance advancements in prior transmission methods came about due to investigating what caused the *ills* in that particular method. By example the FM-Stereo system: High frequency distortion and peak level overshoots were very common in early FM-Stereo generators. Both the preemphasis boost, and sharp cutoff of the required low pass filters, caused severe problems within the system. In-depth analysis of the system lead to the discovery of embedded preemphasis management and non-overshooting low pass filters, which dramatically improved FM-Stereo performance. By researching the difficulties within the system, and then utilizing the gathered information, it enabled new means by which the challenges were overcome. The same applies to coded audio systems too.

While the concern for FM-Stereo was distortion and overshoot, coded audio suffers from what are referred to as *sonic artifacts*. These are the perceptible annoyances that bother the listener. Most sound anomalies are categorized as one form of distortion or another. Most common are harmonic distortion (THD) and intermodulation distortion (IMD). Coding artifacts are neither. When they are perceived, they occur due to inadequacies of the coding algorithm. Basically, this is the point where the coder runs out of capability to reduce the audio data without the process of data reduction being heard. While there have not been specific technical terms assigned to describe these artifacts, they can be referred to as *swishy-swirly, underwater-like, gurgle-like,* and sometimes *synthetic-metallic*. All of these characteristics degrade sound quality, and reduce intelligibility.

PRIOR ATTEMPTS

Dynamic signal processing does provide benefits to coded audio. Dedicated audio processors that utilize look-ahead limiting and bandwidth control do improve sound performance, but they still do not reduce artifacts enough, at low bitrates…especially below 48kbps. HD-Radio, satcasters, podcasters, and netcasters employ bitrates at 24kbps, and lower in some instances. Reducing artifacts at these low rates usually requires severe bandwidth reduction, which in turn dulls the sound quality.

BUT WAIT, THERE'S MORE!

Additional, careful listening to lower bitrate coded audio revealed another underlying discoloration of the signal. Not necessarily artifact-like, and not really distortion, but the audio quality within the presence range, sounds like there is some type of degrading *ghost-like product* being carried along with the signal. Attempts to remove it via signal processing seemed to increase this characteristic. Careful listening to the output of the audio processor, prior to the encode/decode section sounded very clean. Upon adding the codec to the scenario, this annoyance returned. **Note:** This problem was observed with use of a common known codec for HD Radio…and…various audio processors of different designers/companies were used. All of them produced the same results.

A clue to the problem was revealed when the timing in one of the audio processors was modified to reduce the amount of fast-limiting applied to presence and high frequencies. (This did not remove the limiting in this spectra, but changed the manner in which the limiter's timing responded to transient signals.) The audio immediately opened up, along with clarity in the presence and high frequency range. The ghost-like products were gone. So…what's going on?

UNDER THE SCOPE: TRANSIENT IMD

Considering that the modification to the timing of the audio processor lead to the change in sound, consideration was given to the effect of processor induced IMD within the codec. The following simple test was crafted to observe the effects of IMD through a codec.

Figure- 1 illustrates the test setup. A multi-tone sinewave generator creates the source signals to stress the audio processor and codec. Frequencies were set to 400 Hz and 11.5kHz. The output from the audio processor was routed in two directions: to the input of a multi-channel spectrum analyzer, and to the input of an HD Radio encoder. The encoder was routed directly to a corresponding decoder, and its output was connected to the other input of the spectrum analyzer.

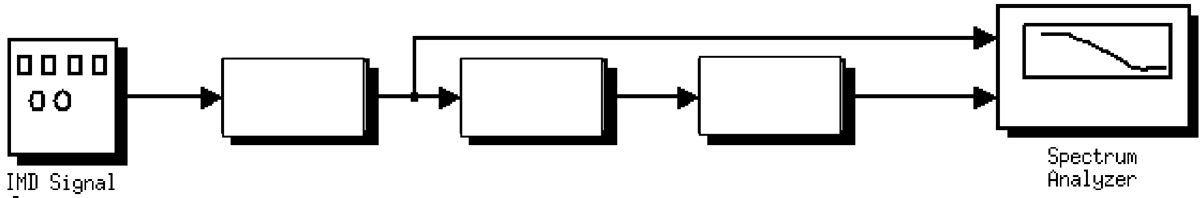

Figure - 1

The objective of this test is to observe whether or not any part of the dynamics function will generate distortion via the codec. The audio processor employed for the test is designed to condition audio in a coded environment. The back end processing utilizes look-ahead limiting, in place of hard limiting/clipping. This reduces THD components in the codec and eliminates aliasing in the system. Tone bursts of the twin tones were used, as this would simulate the effects of transient activity in the source signal, as well as activate the fast-limiting functions in the audio processor.

Figure – 2 is the spectral illustration of the tone bursts at the output of the audio processor. The twin-tones appear as would be expected. This is also the result when observed at the output of the codec when steady state tones are passed through the processor and codec together.

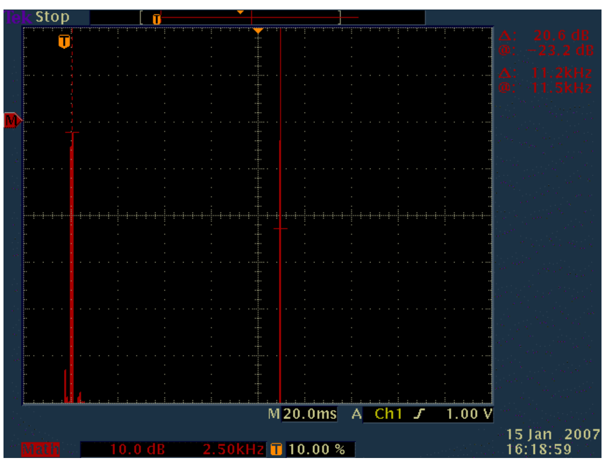

Figure - 2

Figure – 3 illustrates the output of the codec's decoder. Ah, Houston, we've got a problem! Notice the significant spectra around the upper frequency of 11.5kHz. Further investigation of the situation revealed that the transient activity upset the encoder and caused added modulation in the upper frequency domain. This is what was causing the added ghost-like product heard prior. Possibly the effect of the SBR function becoming upset at transient information? This is only a hunch, and again, trying to diagnose the issue is beyond the scope of this paper.

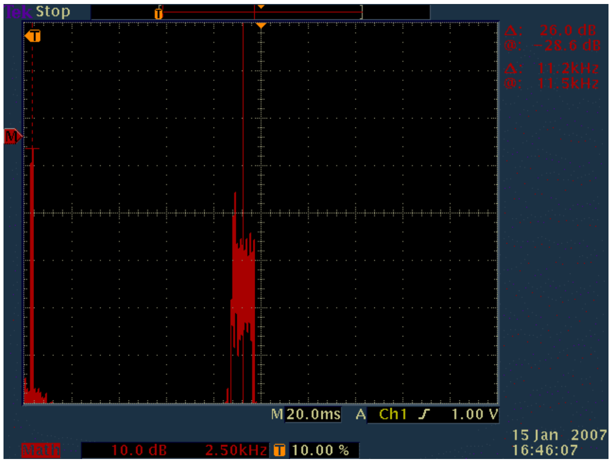

Figure - 3

The rigor of this test exhibits what appears to be severe IMD in the signal. While broadcast source material does not contain transient twin-tones, it does contain plenty of dynamically transient signals within this frequency range. The extent of this added IMD is dependent upon the transients embedded in the source material. Additionally, fast-limiting time constants in the audio processor are capable of exaggerating, and/or creating this problem.

LoIMD

As with most discoveries, there's an answer. In this case, further study of the presence and high-frequency limiting algorithms yielded a method to reduce processor induced IMD. Utilizing a proprietary new function known as *LoIMD*, the algorithm is capable of providing fast-limiting to control transients, yet without agitating the encoder and causing the afore mentioned IMD. When normal source content material is applied, the audio through the entire coded system is devoid of the ghost-like annoyances that were mentioned earlier.

The LoIMD function modifies the control function within a dynamics algorithm. Through internal analysis of the incoming dynamics, and IMD characteristics, the architecture of the control method is rearranged to provide a control signal that reduces, and sometimes eliminates IMD in the processed signal. The sonic result is cleaner sound for a given amount of gain control.
While the LoIMD function answers the issue regarding coding artifacts tied to induced IMD, another critical issue is achieving improved voice intelligibility. For low bitrate applications, this is vital.

CODEC PROVISIONING

Traditional dynamics processors are designed to fulfill the requirements of a medium where the functions are *static*, such as precision peak control and bandwidth limiting for conventional broadcasting, or the normalization needed for recording and mastering. Each of these functions is a known static entity. They are singular, one-dimensional functions where the target is known and the audio processor is designed to accommodate this.

The audio codec, on the other hand, is a moving target. No two codecs are alike, or sound the same. They vary in sonic quality based upon bitrate…AND…more importantly they vary within the same architecture based upon audio content! Here is where conventional audio processors fall short when used in a coding environment.

Until now, dynamics processing has been able to address *some* of the hurdles and artifacts generated by audio coding. The codec has the ability to adapt and modify its algorithm internally, in order to provide maximum throughput, and this alters the sonic artifacts created by the coding process. Unless an audio processor can do the same, it will *hit and miss* regarding how well it provisions the audio to avoid artifacts. Sometimes coded audio sounds acceptable, and sometimes it doesn't. Conventional processors play games with HF limiters and static low pass filtering to minimize coding anomalies. In order to condition audio in hopes of artifact avoidance, the processing will over-compensate audio bandwidth and dynamics. The result is dull, lifeless sounding audio that still contains audible gremlins.

SENSUS EXPLAINED

Sensus technology takes dynamics processing into a new realm. Instead of two-dimensional static architecture and functionality, Sensus adds a third domain where it modifies processing algorithms, architecture, and functions based upon conditions that are understood by the system. Simply stated, Sensus has the ability to sense what must be done to a signal, and then "rearrange the furniture" to accomplish its goal. There are numerous derivatives to this innovative tech, and it can be scaled to many different applications. Following is a discussion of how this method is applied to a processor used in a coded audio environment.

The Sensus algorithm detects troublesome content for a codec, modifies the processor's architecture, and then makes the appropriate changes. These could be dynamics, bandwidth adjustment, a combination of both, or the elimination of a not needed function. The result is consistent quality through the coded transmission system, even at low bitrates; i.e. 18kbps –

21kbps. Voice by example, especially without any other accompaniment, is very difficult to code at low bitrates without the quality and intelligibility suffering. This new process generates clean, smooth, intelligible, and clear audio that is consistent sounding no matter what the content is.

HEADROOM CONSIDERATIONS

Another important factor regarding the coded system is headroom. Digital systems have an absolute maximum ceiling of 0dBfs. Theoretically, audio levels for transmission should be able to be set right up to this level. But, depending upon the encode/decode implementation, overshoots may occur. This is not consistent from codec to codec, but more so due to the implementation of the codec by various manufacturers. Additional input low pass filters in the encoder may cause headroom difficulties. A well designed encoder will insure that any added input filter possess the same headroom as the system, along without generating overshot that reduces headroom. **Note:** Most filter overshot is of the 2dB – 3dB magnitude, but can exceed this amount depending upon filter characteristics.

It would be wise to test any codecs within a specified infrastructure to make sure that 0dBfs, is attainable without system overload or clipping. For this reason, setting the absolute peak level 2dB – 3dB below 0dBfs, offers insurance to avoid clipping.

PROCESSING FOR MULTICAST, STREAMS, DAB, DRM, ETC: SONIC TONIC

The advent of HD Radio[R] has introduced the capability to broadcast multiple content streams within the 96kbps digital channel. To facilitate multicast requires the use of lower bitrate audio coding. The broadcaster can choose the bitrate for each content channel, as well as the number of desired channels, with a maximum limit of seven. Therefore it is possible that extremely low bitrate audio channels will exist, and those will require dynamics processing capable of consistent sound quality that yields low, or no sonic artifacts.

Research, of which this paper is based upon, has yielded a new audio processor for multicast An innovative codec provisioning algorithm - using Sensus Technology, and LoIMD limiters, yields consistent audio quality that contains little, if any, coding artifacts. Yet, audio quality does not suffer the dull or muffled quality due to extreme bandwidth reduction that would normally be employed to mask codec "nasties."

Now it is possible for lower bitrate channerls to offer high quality and clear intelligibility through the use of a dedicated processor that employs the means to understand and handle the challenges of the coded audio path.

For those who wish to tweak on their own, with existing processing equipment, the following should be observed:

1. Avoid dense processing that contains fast limiting time constants. Try to reduce the attack time on functions when 5dB, or more, depth-of-compression is desired. This will reduce upper frequency processor induced IMD.

2. Make sure that the coding system provides full headroom. If the system clips, on its own before 0dBfs, then reset the maximum input level to avoid system headroom problems.

3. Low bitrates will benefit from bandwidth control. A static low pass filter will reduce artifacts. The tradeoff will be perceived high frequencies vs. quality. A specialized processor for coded audio will offer some dynamic method to accomplish this.

4. Do not use *any* final limiter that contains a clipper. The THD generated by the clipping function will cause more trouble than it's worth. Precision peak control is needed in the coded system. As mentioned prior, specialized processing for this medium will provide a look-ahead limiter to accomplish this task.

If the above four items are followed, improved coded audio will result. The following section offers further insight as to why a hard-limiter/clipper is a bad application for coded audio.

CODECS AND CLIPPING

Sound mediums require peak control to avoid loss of headroom and eventual system distortion. Precsion peak limiting is employed to accomplish this. Hard limiting, or peak clipping is used in conventional broadcasting, and it works quite well. The method does not technically degrade the system. (Overuse of final limiting is a subjective adjustment, and too much can degrade performance.) Suffice it to say that hard limiting does work as a precision peak controller within FM-Stereo and AM transmission.

The coded path offers a different set of challenges. It is not possible to overmodulate the system, as there is a precise peak ceiling of 0dBfs. Sorry, but +6dBfs is not possible! (Last statement provided for the Programming brethren.) Precision peak control is required, but the conventional method of clipping creates systemic problems, and those occur as aliasing products within the encoder. *Figure – 4* is an example of what happens to a 2kHz tone, when clipped and 15kHz low pass filtered in a conventional audio processor, used for FM-Stereo, and passed through the HD Radio codec. This problem is consistent with other codecs too.

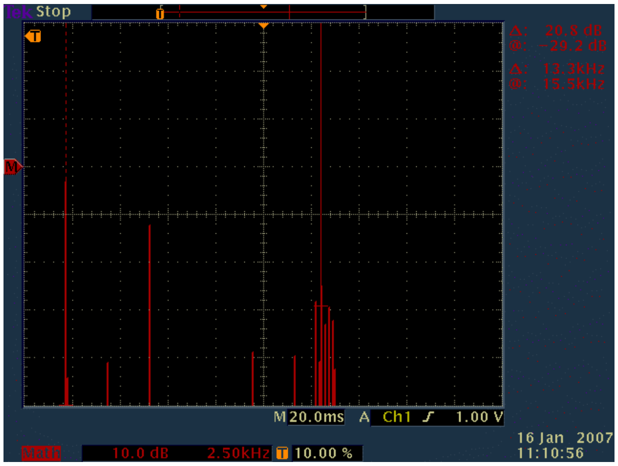

Figure - 4

The cluster of energy that appears around 15kHz are aliasing components. These were caused by the 2kHz clipped signal from a conventional audio processor, as the hard limited signal was routed to the codec. This is proof that all peak limiting for coded audio must employ a limiting means that is void of THD content. Clipped waveforms are exceedingly high in THD. This is why the use of look-ahead limiting is the preferred limiting mechanism for coders. This style of limiter yields very low THD, and will not alias the system.

For reference purposes, *Figure – 5* is the same signal, prior to the codec. Notice how the odd harmonics line up as would be expected from a clipped waveform. The added strange content that appears around 15kHz in *Figure – 4* is what exaggerates coding artifacts when conventional style processing is applied to coded audio.

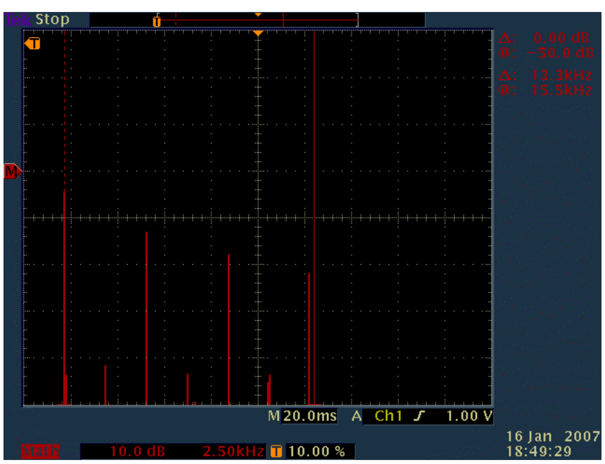

Figure - 5

CONCLUSIONS

Research, testing, development, and hopefully sound reasoning offered here, now explain why coded audio performs as it does. Various signal processing and conditioning means can be used to bring to life coded sound. The test results illustrated here reveal that conventional compressors and limiters exaggerate artifacts. While signal processing, conditioning and peak limiting is required for coded audio, the processing must employ methods that do not contribute additional distortion aspects, as this is what degrades clarity and quality at low bitrates, and sometimes even at moderate to higher rates.