WRITTEN REVIEW 2:
Design Specifications for a DSP Broadcast Mastering System

Ben Beverley 420170037

Digital Audio Systems, DESC9115, Semester 1 2013
Graduate Program in Audio and Acoustics
Faculty of Architecture, Design and Planning, The University of Sydney

INTRODUCTION

In January 2013 new broadcast audio delivery standards that had already been implemented in the USA and Europe came into effect in Australia. The new standards, outlined in the Free TV Australia Operational Practice OP-59 (2010), are based on the ITU-R BS.1770-3 (2012) recommendation. The ITU-R specifies design methods for audio level meters to ensure loudness standards are met and TV viewers experience a more even loudness of content across different programmes and commercials.

There has been a number of digital audio applications released to assist accurate loudness monitoring by audio professionals. These plugins and standalone applications can be time consuming, expensive and difficult to operate, particularly for non-audio professionals. It is common for video editors with limited knowledge of audio processing and metering to be required to deliver content direct to broadcast. Television networks too must implement “tech checks” on all delivered content, often rejecting deliveries based on inappropriate audio and loudness levels or even digital clipping.

This paper provides DSP design specifications for the implementation of an all-in-one final broadcast level assessment and adjustment application. The processor must be capable of accurately analysing and correcting multi-channel audio signals to meet level standards, whilst remaining as sonically transparent as possible. Some mathematical specifications are included; however for comprehensive details on the loudness metering, please refer to ITU-R BS.1770-3.

SIGNAL FLOW

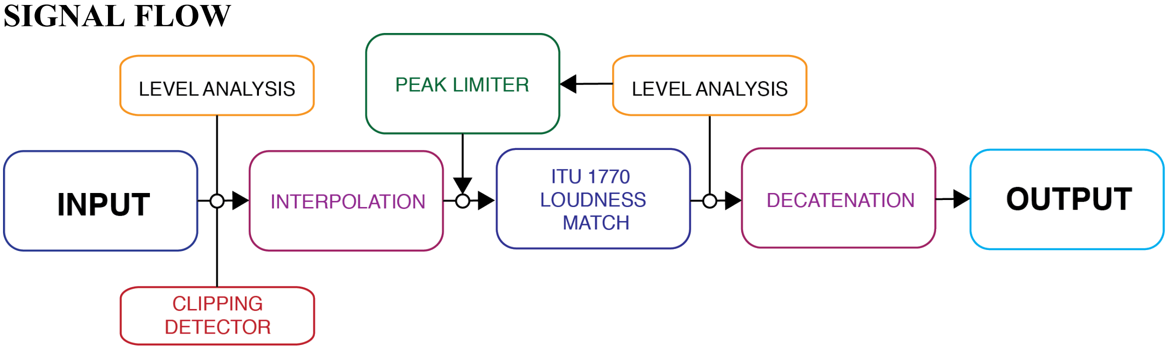


Figure 1. *Signal flow of the loudness correction processor*

The signal flow of this processor is relatively straightforward: after initial level analysis and clipping detection, the signal is interpolated (upsampled) for the true-peak measurement. It then passes into an iterative correction loop, employing a brickwall peak limiter to ensure the maximum peaks are below the desired amount

(-2 dBFS in Australia), whilst adjusting the overall signal level to reach the user-defined target LKFS value. Once the LKFS target and correct peak levels are reached, the signal is decatenated (downsampled) and output.

TRUE-PEAK LEVELS

ITU-R BS.1770-3 specifies the use of true-peak meters over peak-sample meters, which are the standard for many digital audio workstations. The importance of this specification can be seen in Table 1, which shows the maximum peaks of a stereo audio file. The peak-sample meter indicates that the audio levels, although very close to 0dBFS, are still below it and thus free of clipping. The true-peak meter however reveals that the signal in fact peaks above the 0 dBFS level and clipping may result in some D/A converters.

Meter Type	Channel 1 (Left)	Channel 2 (Right)
Peak-sample (dBFS)	-0.1000	-0.0994
True-peak (dB TP)	0.0655	0.1261

Table 1. *Peak sample meter vs. true-peak meter showing that clipping has already occurred without the peak sample meter registering any*

This is because true-peak metering more closely estimates the actual peaks of a signal as though it were in the continuous-time domain, whereas peak-sample meters only measure the value of the highest discrete-time domain sample (Lund, 2006). The simplest way of dealing with this error inside the discrete domain is to upsample the signal using the DSP process; interpolation. This process involves inserting zeros into the signal and applying a lowpass anti-aliasing filter; **Figure 2** shows an example of this process. It can be seen that the pulse-train of the interpolated signal appears more continuous than the original signal. The information in each can in fact be converted into identical continuous domain signals, yet the maximum value of the inter-sample peaks are not necessarily the same.

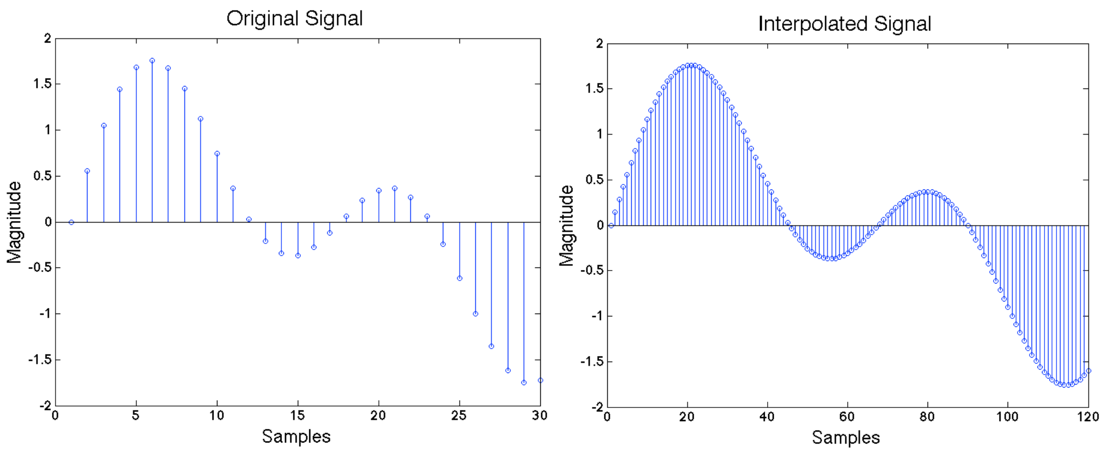


Figure 2. *Original signal vs. interpolated signal; upsampled by a factor of 4*

The accuracy obtained by oversampling a signal for true-peak detection is determined by the following equation:

$$20.\log(\cos(\pi . f_{norm} / n)) \tag{1}$$

where n is the over-sampling ratio, and

f_{norm} is the maximum normalised frequency (the ratio of the highest frequency to be metered to the sampling frequency) (ITU-R BS.1770-3)

This equation shows that the higher the frequency content of a transient, the larger the potential under-read. It is calculated that for a 4x over-sampling rate, the under-read maximum at $f_{norm} = 0.45$ is merely 0.554 dB. In contrast, the under-read maximum at the original sample rate is -2.31 dB.

ITU-R BS.1770-3 provides guidelines for the accurate estimation of true-peaks in a PCM digital audio signal. The algorithm is a four-stage process that involves attenuation, oversampling, a low-pass filter and then peak analysis of the absolute values.

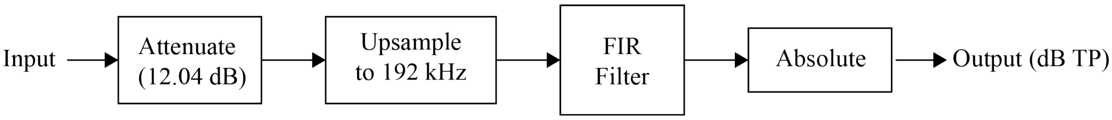


Figure 3. *Block diagram showing the ITU-R BS.1770-3 recommended process for accurate true-peak level measurement*

Attenuation is first performed to provide headroom for the subsequent signal processing. The -12.04 dB amount allows for a 2-bit shift; a step that is unnecessary if the calculations are performed in floating-point such as a Matlab function. Upsampling is then performed by inserting zeros into the signal between samples to bring the sample rate up 192 kHz (a 4x factor of the broadcast standard 48 kHz).

A finite impulse response (FIR) low-pass interpolating filter is then applied to prevent aliasing from the upsampling process. The filter must be made of a large amount of taps to ensure a very low passband ripple, achieve extreme stop-band attenuation and a narrow transition band. The 48 taps specified in ITU-R BS.1770-3 would not be suitable for this application given that there is audio passing through the processor and not merely analysis happening. A minimum 128 taps should suffice, although actual testing would need to be performed. The response of both a 48-order filter and 128-order filter, as seen in Figure 4, displays this necessity. Notice how the 48-order filter has a rippling at the top of the signal (the passband), a slower roll-off at the stop-band (in this case at 20 kHz), and less attenuation (approximately -26 dB vs. -50 dB).

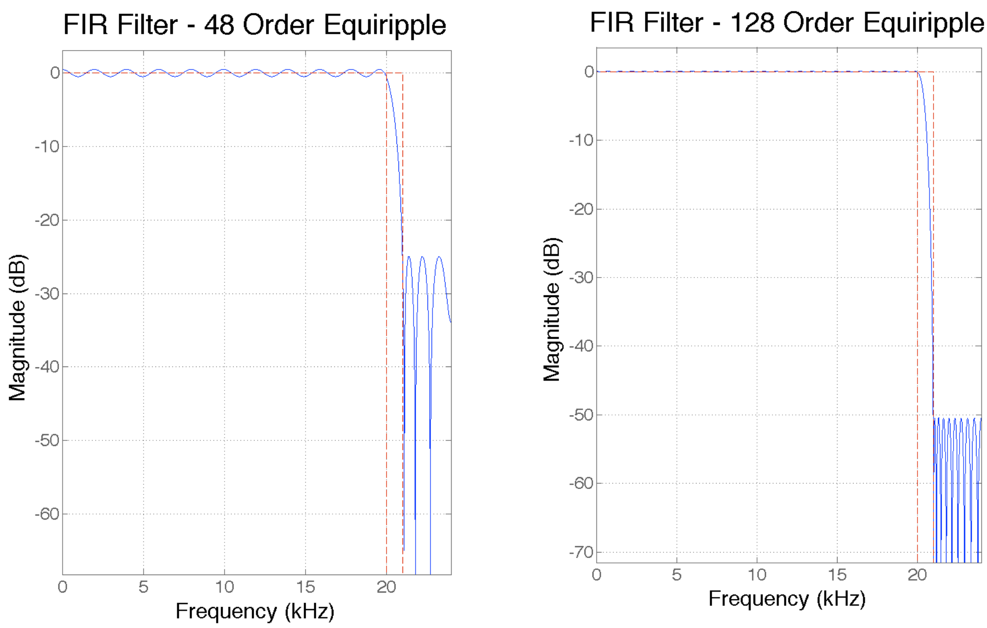


Figure 4. *Magnitude responses for a 48-order equiripple FIR filter vs. 128-order equiripple FIR filter*

The final stage is the analysis of the absolute value of the peaks, which is done by inverting the negative value samples to make the signal unipolar. The output of the peak meter is a more accurate estimation of the true-peaks of the audio signal and is labelled dB TP (decibels true-peak).

ITU-R BS.1770-3 LOUDNESS MATCH

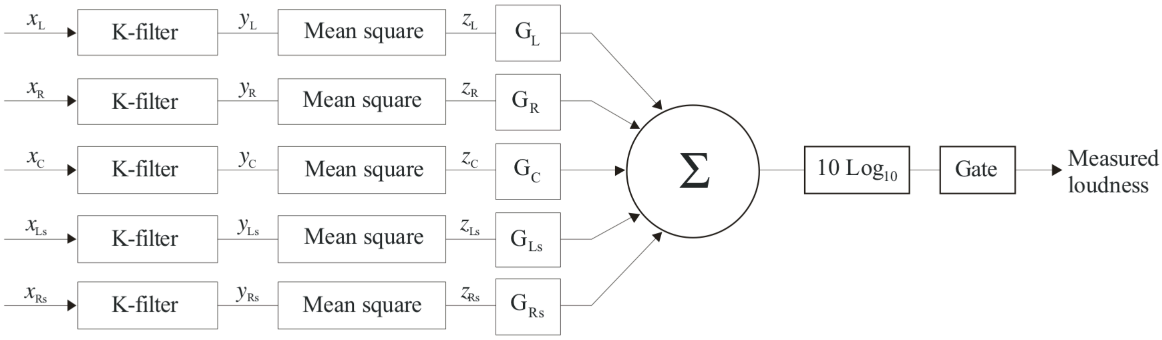


Figure 5. *Simplified block diagram of the multichannel loudness algorithm (ITU-R BS.1770-3, 2012)*

The algorithms specified in ITU-R BS.1770-3 to measure loudness levels (dB LKFS) are based on the results of extensive subjective psychoacoustic listening tests. The frequency response of human hearing is first modelled in the K-filter (shown in **Figure** . This FIR filter consists of two stages of filtering: the first is a +4 dB shelving filter, which boosts high frequencies to roughly simulate the increased hearing sensitivity of the human ear seen in loudness contours; whilst the second stage is a second order low-pass filter to negatively weight the lower frequencies.

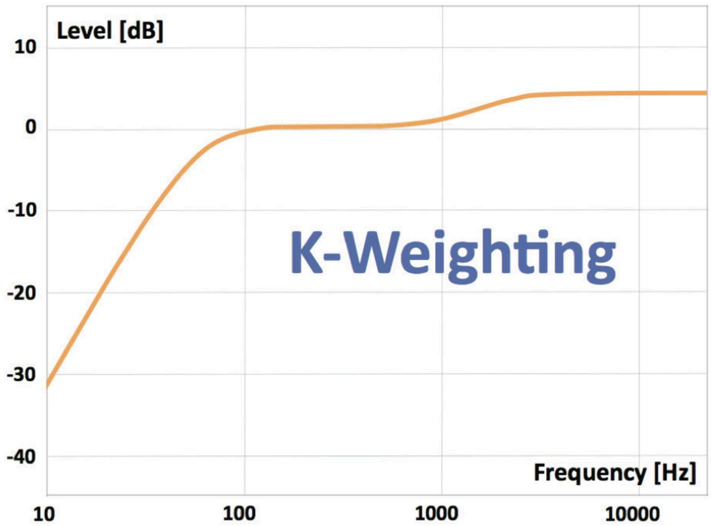


Figure 6. *K-Weighting filter curve to model human hearing (EBU TECH 3343)*

The level of the signal is then measured by finding the mean square (z_i) over the measurement interval T:

$$z_i = \frac{1}{T}\int_0^T y_i^2 \, dt \tag{2}$$

where y_i is the K-weighted signal, i is the set of input channels (L,R,C,Ls,Rs) and dt is the measurement block size.

The loudness (L_K) is then calculated:

$$L_K = -0.691 + 10\log_{10}\sum_i G_i \cdot z_i \tag{3}$$

where G_i are the weighting coefficients for each channel.

The channel weighting coefficients allow for the fact that sounds arriving from behind a listener may be perceived to be louder than sounds arriving from the front. Both surround channels have a weighting of G = 1.41 (~+1.5 dB) to the LKFS values, which means that the final loudness level reading will be higher when there is more content in the surround channels.

Calculation of the mean square (z_{ij}) is done in gating blocks; sets of contiguous audio samples. The specified duration of the gating block is T_g = 400 ms (to the nearest sample). Once the mean square values has been calculated for all of the gating blocks, the relative threshold (Γ_r) is determined by:

$$\Gamma_r = -0.691 + 10\log_{10}\sum_i G_i\left(\frac{1}{|J_g|}\cdot\sum_{J_g} z_{ij}\right) - 10LKFS \tag{4}$$

where $J_g = \{j : l_j > \Gamma_a\}$, and Γ_a is the absolute threshold, -70 LKFS.

Finally the single value gated loudness is determined:

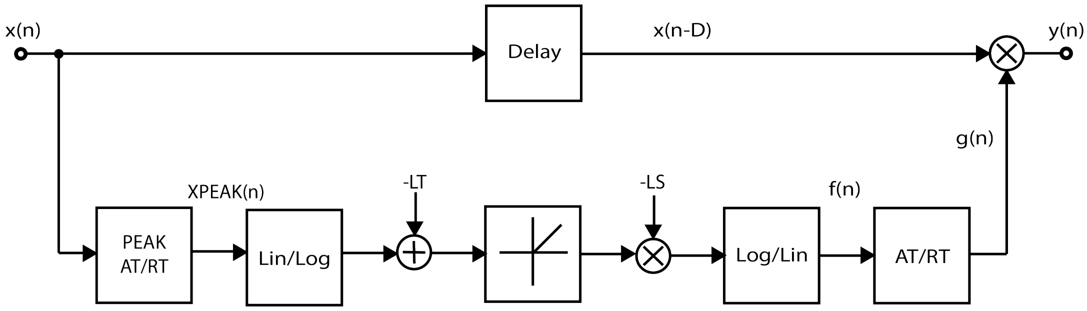
$$L_{KG} = -0.691 + 10\log_{10}\sum_i G_i\left(\frac{1}{|J_g|}\cdot\sum_{J_g} z_{ij}\right) LKFS \tag{5}$$

PEAK LIMITER

The peak limiter in this signal processing application is implemented in iteration with the loudness matching processor to ensure both the correct maximum peaks and overall loudness criteria are met. In other words: if the limiter runs then the signal is looped back into the LKFS matching algorithm again. The loop will continue until both the peak level target and LKFS target criteria are met.

Since the purpose of this processor is to ensure satisfactory broadcast levels are attained, the limiter design must be unforgiving of peaks above the threshold value with a compression ratio of infinity-to-one (i.e. A "brickwall" limiter). Although both feedback and feedforward limiters can be made to function equivalently when the compression ratio is constant (Abel & Berners, 2003), the simplicity and versatility of the feedforward design will leave further opportunity for refinement and development if more intelligently transparent gain management options are to be explored (see Further Research section later).

The DSP algorithms underlying the limiter can be understood by breaking down the processes in the following feedforward limiter block diagram:



AT = Attack Time
RT = Release Time
LT = Limiter Threshold
LS = Limiter Slope

Figure 7. *Block diagram of the feedforward limiter*

The input signal *x(n)* enters the system and is split into a delayed component; *Delay* and a side chain path. The *Delay* signal is delayed to account for the time it takes for the side chain path to be calculated, *x(n-D)*.

The side chain path first measures the peaks of the input signal and determines whether the processor is in attack or release mode. The signal *XPEAK(n)* is then moved into the logarithmic domain for the static processing. The limiter threshold (*LT*) is applied using addition (note: in the linear domain this process is in fact multiplication (THAT Corporation 2009)).

Calculation of the control parameter *f(n)* in the logarithmic domain *F* in dB is performed by the line equation:

$$F_L = -LS(X - LT) + CS(CT - LT) \tag{6}$$

The slope factor *S* is calculated by:

$$S = 1 - \frac{1}{R} \tag{7}$$

and *R* is the compression factor:

$$R = \frac{\Delta L_I}{\Delta L_O} = \frac{1}{1 - S} \tag{8}$$

where ΔL_I is the fraction of input level change and ΔL_O is the fraction of output level change. Typically the *S* value for a limiter is equal to 1 and the *R* value is ∞.

The output factor *f(n)* of the static function is used as the input signal to the dynamic filter, creating the weighting variable *g(n)*, the gain factor. This gain factor is multiplied by the delayed input signal *x(n – D)* to produce the system output *y(n)*.

$$y(n) = x(n - D) \cdot g(n) \tag{9}$$

A brickwall limiter design requires fast *AT* and *RT* values to ensure the threshold is never exceeded and the limiter remains relatively transparent to short transients. Mathematically the calculations of the attack time and release time parameters are:

$$AT = 1 - e^{-2.2T/t_{AT}} \qquad RT = 1 - e^{-2.2T/t_{RT}} \tag{10}$$

where t is the time parameter in seconds and T is the sampling period.

FURTHER RESEARCH

The first ITU-R BS.1770 publication (1770-1) was found to unfairly weight the more dynamic programmes and genres with longer quiet sections. Investigations by the EBU P/PLOUD group found that a simple yet effective gate, which excluded levels below a certain threshold, would counter this problem (Grimm et al., 2010). The second revision of the ITU-R BS.1770 standard (1770-2) then introduced the specification of such a gate. Travaglini (2013) argues that although this is a step in the right direction, it fails to take into account the “anchor element” of programmes, which is normally the dialogue. A more intelligent loudness meter based on voice-detection would assist in ensuring uniform programme loudness not only in television broadcast, but across other mediums also.

A DSP process that can be used for dialogue detection is cross-correlation. Kotti et al. (2008) describe how dialogue is a repetitive, non-random pattern. Speech has typical pulse lengths and silent periods between utterances, which can be quantified to derive a signal typical of dialogue. This signal can be cross-correlated with any other signal to determine how closely the content matches. Significantly large values of the cross-correlation function indicate the presence of dialogue. Furthermore, the discrete-time Fourier transform of the cross-correlation, which is known as the cross-power spectral density, can also be used to detect dialogue. This frequency-domain process shows which frequencies are strongly related to the spectral characteristics of dialogue (Kotti et al., 2008).

Dolby’s LM100 Broadcast Loudness Meter contains a proprietary dialogue level analysis algorithm called *Dialogue Intelligence* (Dolby Laboratories, 2013) which measures the perceived loudness of dialogue perhaps in a similar DSP method as mentioned above. Incorporating such a process into the DSP application outlined in this paper could prove very useful, but without an internationally standardised method for dialogue level measurement, it is currently not necessary.

Travaglini et al. (2012) propose further revisions to the ITU-R BS.1770-3 processing with a loudness algorithm called HELM (High Efficiency Loudness Level). HELM differs from the current standard in that it factors in the LFE channel; weights the spatial channels differently (based on HRTF findings related to how we perceive sound source directions); and incorporates a -7 recursive gating threshold instead of the current -10.

Pestana et al. (2013) too argue the need for further revisions to ITU-R BS.1770-3, having discovered a number of more suitable settings as a result of recent subjective listening tests. They suggest that the +4 dB shelving filter used to simulate hearing sensitivity in the mid-high frequencies, trades peak gain for a broader bandwidth and should probably be set higher. They also experimented with various different time constants for the windowing value to the gating block in an attempt to rectify the current issue whereby the algorithm underestimates the loudness of percussive material with limited high-range spectral bandwidth (e.g. hi-hats, shakers, tambourines).

The limiter specified in this paper could also be developed further, particularly if the effect were to be implemented for real-time DAW processing. A more transparentsounding peak limiter is possible through the manipulation of a variable release time parameter. The Sonnox Oxford Limiter, for example, boasts very transparent gain reduction up to about 12 dB through the use of a variable release time that fluctuates up to about 10 seconds to ensure gain changes are less audible (Inglis, 2005).

CONCLUSIONS

The logical area for further development of the DSP system described in this paper is in the limiter processing. It is currently an effective brickwall, which serves the purpose of ensuring maximum peaks are tamed; however if a more sonically transparent DSP algorithm were implemented, the system could extend beyond merely providing a fast 'tech-check' application and become a valuable DAW real-time level management tool. A large degree of testing and optimisation to minimise processing latency would be required.

The first ITU-R BS. 1770 loudness standard (1770-1) introduced in 2010 has already seen two revisions, as its usefulness and applicability has been tested through industry practice and further psychometric research. It seems likely that further revisions will result, however, the emphasis must continue to be on algorithms that are simple and DSP efficient to implement. As ideal as a dialogue detection algorithm seems, if the processing load is too high, it seems unreasonable to expect the broadcast industry to embrace the specification and is therefore unlikely to be adopted by the ITU-R anytime soon.

REFERENCES

Abel, J.S. & Berners, D.P. 2003, 'On Peak-Detecting and RMS Feedback and Feedforward Compressors', *Presented at the 115th Convention of the Audio Engineering Society,* New York, USA.

Dolby Laboratories, 2013, Dolby Broadcast Loudness Meter LM100. [ONLINE] Available at: http://www.dolby.com/us/en/professional/hardware/broadcast/test-and-measurement/lm100.html [Accessed 23 May 13]

European Broadcasting Union 2011, *EBU – TECH 3343: Practical guidelines for Production and Implementation in accordance with EBU R 128,* European Broadcasting Union, Geneva.

Free TV Australia 2010, Operational Practice OP-59: Measurement and Management of Loudness in Soundtracks for Television Broadcasting, Issue 1, Sydney.

Grimm, E., Skovenborg, E., Spikofski, G., 2010, 'Determining an Optimal Gated Loudness Measurement for TV Sound Normalization', *Presented at the 128th Convention of the Audio Engineering Society,* London, UK.

Inglis, S. 2005, 'On Test: Sony Oxford Limiter: Mastering Limiter Plug-in For Pro Tools', *Sound on Sound*, December 2005, pp112-3.

International Telecommunication Union 2012, *Recommendation ITU-R BS.1770-3: Requirements for loudness and true-peak indicating meters*, International Telecommunication Union, Geneva.

Kotti, M., Ververidis, D., Evangelopoulos, G., Panagakis, I., Kotropoulos, C., Maragos, P., Pitas, I. 2008, 'Audio-Assisted Movie Dialogue Detection', *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, pp1618-27

Lund, T., 2006, 'Stop Counting Samples', *Presented at the 121st Convention of the Audio Engineering Society,* San Francisco, USA.

Pestana, P., Reiss, J., Barbosa, A. 2013, 'Loudness measurement of multitrack audio content using modifications of ITU-R BS.1770', *Presented at the 134th Convention of the Audio Engineering Society,* Rome, Italy.

THAT Corporation. 2009. *The Mathematics of Log-Based Dynamic Processors*, viewed April 15 2013, http://www.thatcorp.com/datashts/dn01A.pdf

Travaglini, A., Alemanno, A., Uncini, A. 2012, 'HELM: High Efficiency Loudness Model for Broadcast Content', *Presented at the 132nd Convention of the Audio Engineering Society,* Budapest, Hungary.

Travaglini, A. 2013, 'Comparative analysis of different loudness meters based on voice detection and gating', *Presented at the 134th Convention of the Audio Engineering Society,* Rome, Italy.

Zölzer, U, 2002. DAFX - Digital Audio Effects. 1st ed. West Sussex, England: John Wiley & Sons, Ltd.