RECOMMENDATION ITU-R BS.1770

Algorithms to measure audio programme loudness and true-peak audio level

(Question ITU-R 2/6)

(2006)

Scope

This Recommendation specifies audio measurement algorithms for the purpose of determining subjective programme loudness, and true-peak signal level.

The ITU Radiocommunication Assembly,

considering

a) that modern digital sound transmission techniques offer an extremely wide dynamic range;

b) that modern digital sound production and transmission techniques provide a mixture of mono, stereo and multichannel formats and that sound programmes are produced in all of these formats;

c) that listeners desire the subjective loudness of audio programmes to be uniform for different sources and programme types;

d) that many methods are available for measurement of audio levels but that existing measurement methods employed in programme production do not provide indication of subjective loudness;

e) that, for the purpose of programme exchange, it is essential to have a single recommended algorithm for objective estimation of subjective loudness;

f) that future complex algorithms based on psychoacoustic models may provide improved objective measures of loudness for a wide variety of audio programmes;

g) that digital media overload abruptly, and thus even momentary overload should be avoided,

considering further

h) that peak signal levels may increase due to commonly applied processes such as filtering or bit-rate reduction;

j) that existing metering technologies do not reflect the true-peak level contained in a digital signal since the true-peak value may occur in between samples;

k) that the state of digital signal processing makes it practical to implement an algorithm that closely estimates the true-peak level of a signal;

l) that use of a true-peak indicating algorithm will allow accurate indication of the headroom between the peak level of a digital audio signal and the clipping level, *recommends*

1 that when an objective measure of the loudness of an audio channel or programme is required to facilitate programme delivery and exchange, the algorithm specified in Annex 1 should be used;

2 that methods employed in programme production and post-production to indicate programme loudness may be based on the algorithm specified in Annex 1;

3 that when an indication of true-peak level of a digital audio signal is required, the measurement method should be based on the guidelines shown in Annex 2, or on a method that gives similar or superior results,

NOTE 1 – Users should be aware that measured loudness is an estimation of subjective loudness and involves some degree of discrepancy depending on listeners, audio material and listening conditions.

further recommends

1 that further work should be conducted to extend the algorithm specified in Annex 1 to provide indication of short-term loudness;

2 that consideration should be given to the possible need to update this Recommendation in the event that new loudness algorithms are shown to provide performance that is significantly improved over the algorithm specified in Annex 1.

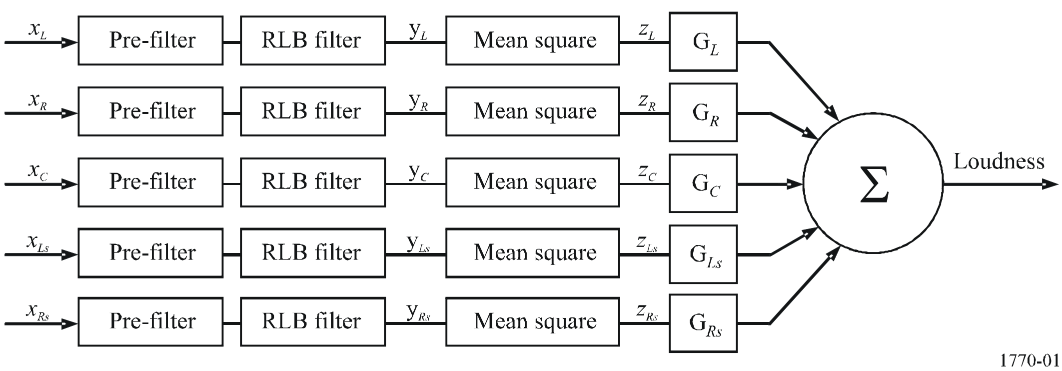
Annex 1

Specification of the objective multichannel loudness measurement algorithm

This Annex specifies the multichannel loudness measurement algorithm. Figure 1 shows a block diagram of the various components of the algorithm. Labels are provided at different points along the signal flow path to aid in the description of the algorithm. The block diagram shows inputs for five main channels (left, centre, right, left surround and right surround); this allows monitoring of programmes containing from one to five channels. For a programme that has less than five channels some inputs would not be used. The low frequency effects (LFE) channel is not included in the measurement.

FIGURE 1

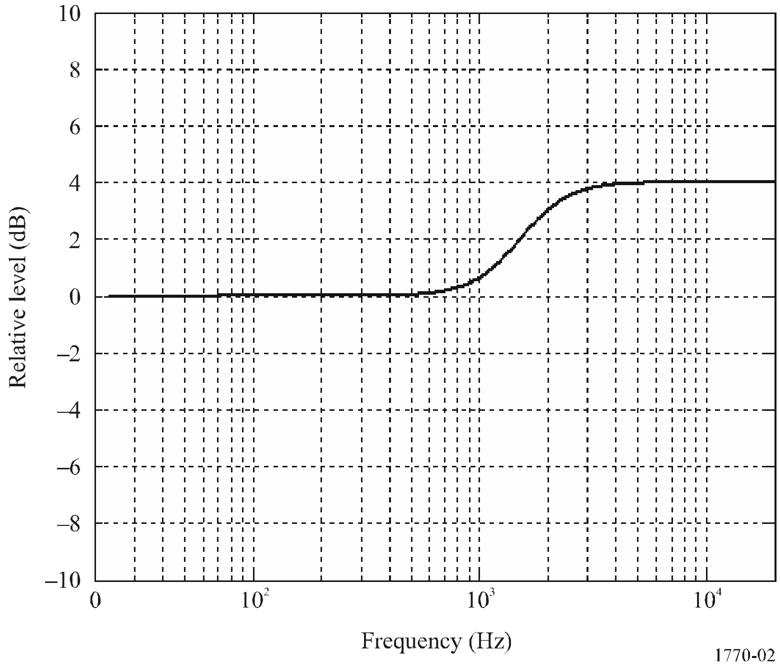
Block diagram of multichannel loudness algorithm



The first stage of the algorithm applies a pre-filtering of the signal prior to the *Leq*(RLB) measure as shown in Fig. 2. The pre-filtering accounts for the acoustic effects of the head, where the head is modelled as a rigid sphere.

FIGURE 2

Response of the pre-filter used to account for the acoustic effects of the head



The pre-filter is defined by the filter shown in Fig. 3 with the coefficients specified in Table 1.

FIGURE 3

Signal flow diagram as a 2nd order filter

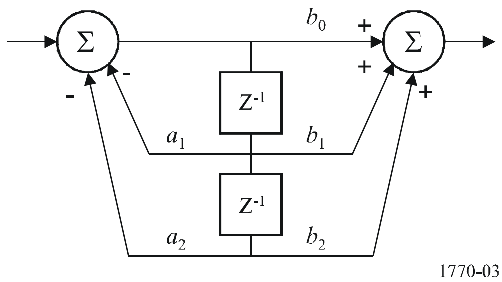


TABLE 1

Filter coefficients for the pre-filter to model a spherical head

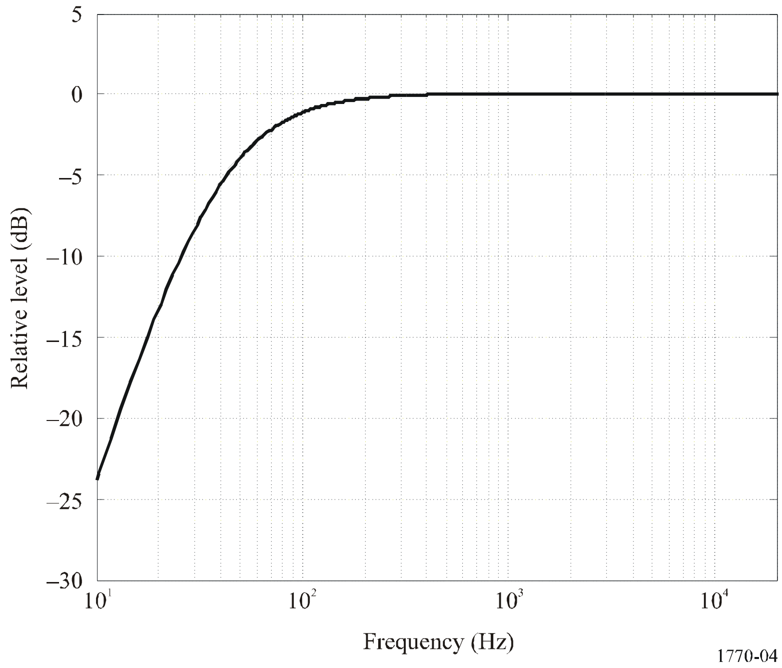
		b_0	1.53512485958697
a_1	−1.69065929318241	b_1	−2.69169618940638
a_2	0.73248077421585	b_2	1.19839281085285

These filter coefficients are for a sampling rate of 48 kHz. Implementations at other sampling rates will require different coefficient values, which should be chosen to provide the same frequency response that the specified filter provides at 48 kHz. The values of these coefficients may need to be quantized due to the internal precision of the available hardware. Tests have shown that the performance of the algorithm is not sensitive to small variations in these coefficients.

The second stage of the algorithm applies the RLB weighting curve, which consists of a simple high-pass filter as shown in Fig. 4.

FIGURE 4

RLB weighting curve



The RLB weighting curve is specified as a 2nd order filter as shown in Fig. 3, with the coefficients specified in Table 2.

TABLE 2

Filter coefficients for the RLB weighting curve

		b_0	1.0
a_1	−1.99004745483398	b_1	−2.0
a_2	0.99007225036621	b_2	1.0

These filter coefficients are for a sampling rate of 48 kHz. Implementations at other sampling rates will require different coefficient values, which should be chosen to provide the same frequency response that the specified filter provides at 48 kHz.

With the pre-filter and the RLB filtering applied, the mean-square energy in the measurement interval T is then measured as:

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

where y_i is the input signal filtered by both the pre-filter to model the head effects, and the RLB weighting curve. ($i = L, R, C, Ls, Rs, N$ where N is the number of channels).

Once the weighted mean-square level, z_i, has been computed for each channel, the final step is to sum the N channels as follows:

$$\text{Loudness} = -0.691 + 10 \log_{10} \sum_i^N G_i \cdot z_i \qquad \text{dB} \tag{2}$$

If a 0 dBfs 1 kHz sine wave is input to the left, centre, or right channel input, the indicated loudness will equal –3.00 dB.

The weighting coefficients for the different channels are given in Table 3.

TABLE 3

Weightings for the individual audio channels

Channel	Weighting, G_i
Left (G_L)	1.0 (0 dB)
Right (G_R)	1.0 (0 dB)
Centre (G_C)	1.0 (0 dB)
Left surround (G_{Ls})	1.41 (~ +1.5 dB)
Right surround (G_{Rs})	1.41 (~ +1.5 dB)

It should be noted that while this algorithm has been shown to be effective for use on audio programmes that are typical of broadcast content, the algorithm is not, in general, suitable for use to estimate the subjective loudness for pure tones.

Appendix 1
to Annex 1

Description and development of the multichannel measurement algorithm

This Appendix describes a newly developed algorithm for objectively measuring the perceived loudness of audio signals. The algorithm can be used to accurately measure the loudness of mono, stereo and multichannel signals. A key benefit of the proposed algorithm is its simplicity, allowing it to be implemented at very low cost. This Appendix also describes the results of formal subjective tests conducted to form a subjective database that was used to evaluate the performance of the algorithm.

1 Introduction

There are many applications where it is necessary to measure and control the perceived loudness of audio signals. Examples of this include television and radio broadcast applications where the nature and content of the audio material changes frequently. In these applications the audio content can continually switch between music, speech and sound effects, or some combination of these. Such changes in the content of the programme material can result in significant changes in subjective loudness. Moreover, various forms of dynamics processing are frequently applied to the signals, which can have a significant effect on the perceived loudness of the signal. Of course, the matter of subjective loudness is also of great importance to the music industry where dynamics processing is commonly used to maximize the perceived loudness of a recording.

There has been an ongoing effort within Radiocommunication Working Party 6P in recent years to identify an objective means of measuring the perceived loudness of typical programme material for broadcast applications. The first phase of ITU-R's effort examined objective monophonic loudness algorithms exclusively, and a weighted mean-square measure, *Leq*(RLB), was shown to provide the best performance for monophonic signals [Soulodre, 2004].

It is well appreciated that a loudness meter that can operate on mono, stereo, and multichannel signals is required for broadcast applications. The present document proposes a new loudness measurement algorithm that successfully operates on mono, stereo, and multichannel audio signals. The proposed algorithm is based on a straightforward extension of the *Leq*(RLB) algorithm. Moreover, the new multichannel algorithm retains the very low computational complexity of the monophonic *Leq*(RLB) algorithm.

2 Background

In the first phase of the ITU-R study a subjective test method was developed to examine loudness perception of typical monophonic programme materials [Soulodre, 2004]. Subjective tests were conducted at five sites around the world to create a subjective database for evaluating the performance of potential loudness measurement algorithms. Subjects matched the loudness of various monophonic audio sequences to a reference sequence. The audio sequences were taken from actual broadcast material (television and radio).

In conjunction with these tests, a total of ten commercially developed monophonic loudness meters/algorithms were submitted by seven different proponents for evaluation at the Audio Perception Lab of the Communications Research Centre, Canada.

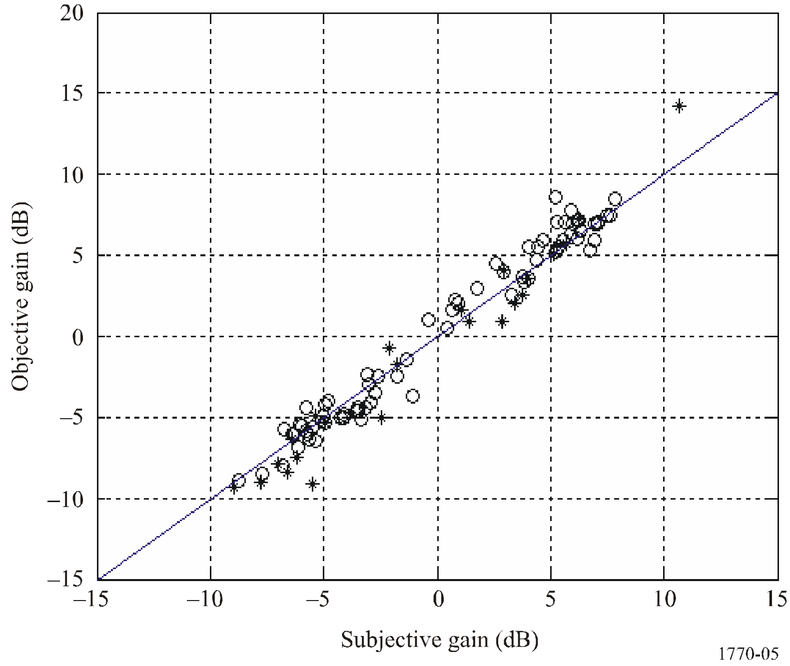
In addition, Soulodre contributed two additional basic loudness algorithms to serve as a performance baseline [Soulodre, 2004]. These two objective measures consisted of a simple frequency weighting function, followed by a mean-square measurement block. One of the two measures, *Leq*(RLB), uses a high-pass frequency weighting curve referred to as the revised low-frequency B-curve (RLB).

The other measure, *Leq*, is simply an unweighted mean-square measure.

Figure 5 shows the results of the initial ITU-R study for the *Leq*(RLB) loudness meter. The horizontal axis indicates the relative subjective loudness derived from the subjective database, while the vertical axis indicates the loudness predicted by the *Leq*(RLB) measure. Each point on the graph represents the result for one of the audio test sequences in the test. The open circles represent speech-based audio sequences, while the stars are non-speech-based sequences. It can be seen that the data points are tightly clustered around the diagonal, indicating the very good performance of the *Leq*(RLB) meter.

FIGURE 5

Monophonic *Leq*(RLB) loudness meter versus subjective results (*r* = 0.982)



Leq(RLB) was found to provide the best performance of all of the meters evaluated (although within statistical significance some of the psychoacoustic-based meters performed as well). *Leq* was found to perform almost as well as RLB. These findings suggest that for typical monophonic broadcast material, a simple energy-based loudness measure is similarly robust compared to more complex measures that may include detailed perceptual models.

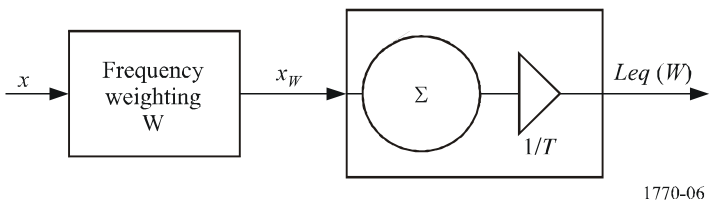
3 Design of the *Leq*(RLB) algorithm

The *Leq*(RLB) loudness algorithm was specifically designed to be very simple. A block diagram of the *Leq*(RLB) algorithm is shown in Fig. 6. It consists of a high-pass filter followed by a means to average the energy over time. The output of the filter goes to a processing block that sums the energy and computes the average over time.

The purpose of the filter is to provide some perceptually relevant weighting of the spectral content of the signal. One advantage of using this basic structure for the loudness measures is that all of the processing can be done with simple time-domain blocks having very low computational requirements.

FIGURE 6

Block diagram of the simple energy-based loudness measures



The *Leq*(RLB) algorithm shown in Fig. 6 is simply a frequency-weighted version of an *Equivalent Sound Level* (*Leq*) measure. *Leq* is defined as follows:

$$Leq(W) = 10 \log_{10}\left[\frac{1}{T}\int_0^T \frac{x_W^2}{x_{Ref}^2}\,\mathrm{d}t\right] \qquad \text{dB} \qquad (3)$$

where:

x_W: signal at the output of the weighting filter

x_{Ref}: some reference level

T: length of the audio sequence.

The symbol W in $Leq(W)$ represents the frequency weighting, which in this case was the revised low-frequency B-curve (RLB).

4 Subjective tests

In order to evaluate potential multichannel loudness measures it was necessary to conduct formal subjective tests in order to create a subjective database. Potential loudness measurement algorithms could then be evaluated in their ability to predict the results of the subjective tests. The database provided perceived loudness ratings for a broad variety of mono, stereo, and multichannel programme materials. The programme materials used in the tests were taken from actual television and radio broadcasts from around the world, as well as from CDs and DVDs. The sequences included music, television and movie dramas, sporting events, news broadcasts, sound effects and advertisements. Included in the sequences were speech segments in several languages.

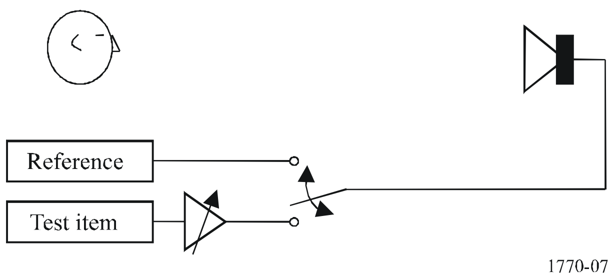
4.1 Subjective test set-up

The subjective tests consisted of a loudness-matching task. Subjects listened to a broad range of typical programme material and adjusted the level of each test item until its perceived loudness matched that of a reference signal (see Fig. 7).

The reference signal was always reproduced at a level of 60 dBA, a level found by Benjamin to be a typical listening level for television viewing in actual homes [Benjamin, 2004].

FIGURE 7

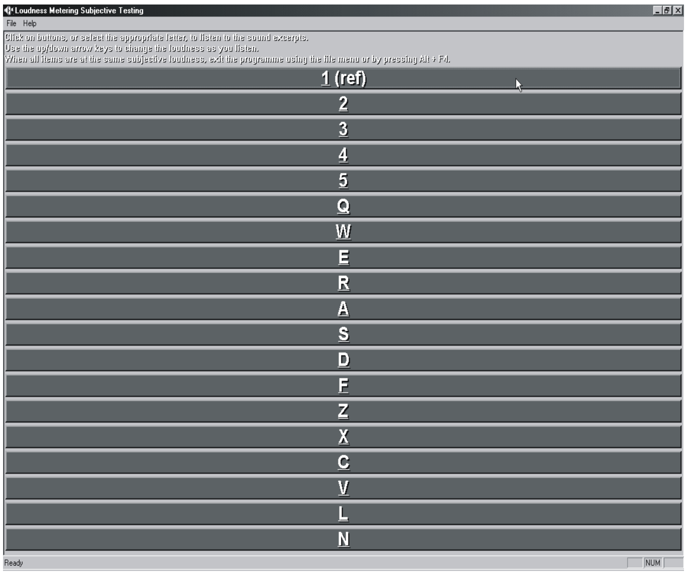
Subjective test methodology



A software-based multichannel subjective test system, developed and contributed by the Australian Broadcasting Corporation, allowed the listener to switch instantly back and forth between test items and adjust the level (loudness) of each item. A screen-shot of the test software is shown in Fig. 8. The level of the test items could be adjusted in 0.25 dB steps. Selecting the button labelled “1” accessed the reference signal. The level of the reference signal was held fixed.

FIGURE 8

User interface of subjective test system



1770-08

Using the computer keyboard, the subject selected a given test item and adjusted its level until its loudness matched the reference signal. Subjects could instantly switch between any of the test items by selecting the appropriate key. The sequences played continuously (looped) during the tests. The software recorded the gain settings for each test item as set by the subject. Therefore, the subjective tests produced a set of gain values (decibels) required to match the loudness of each test sequence with the reference sequence. This allowed the relative loudness of each test item to be determined directly.

Prior to conducting the formal blind tests, each subject underwent a training session in which they became acquainted with the test software and their task in the experiment. Since many of the test items contained a mixture of speech and other sounds (i.e. music, background noises, etc.), the subjects were specifically instructed to match the loudness of the overall signal, not just the speech component of the signals.

During the formal blind tests the order in which the test items were presented to each subject was randomized. Thus, no two subjects were presented with the test items in the same order. This was done to eliminate any possible bias due to order effects.

4.2 The subjective database

The subjective database used to evaluate the performance of the proposed algorithm actually consisted of three separate datasets. The datasets were created from three independent subjective tests conducted over the course of a few years.

The first dataset consisted of the results from the original ITU-R study where subjects matched the perceived loudness of 96 monophonic audio sequences. For this dataset, subjective tests were carried out at five separate sites around the world providing a total of 97 listeners. A three-member panel made up of Radiocommunication WP 6P SRG3 members selected the test sequences as well as the reference item. The reference signal in this experiment consisted of English female speech. The sequences were played back through a single loudspeaker placed directly in front of the listener.

Following the original ITU-R monophonic study, some of the algorithm proponents speculated that the range and type of signals used in the subjective tests was not sufficiently broad. They further speculated that it was for this reason that the simple *Leq*(RLB) energy-based algorithm outperformed all of the other algorithms.

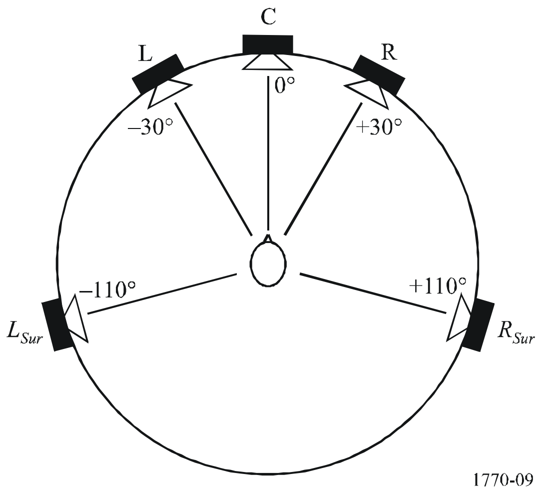
To address this concern, proponents were asked to submit new audio sequences for a further round of subjective tests. They were encouraged to contribute monophonic sequences that they felt would be more challenging to the *Leq*(RLB) algorithm. Only two of the meter proponents contributed new sequences.

Using these new sequences, formal subjective tests were conducted at the Audio Perception Lab of the Communications Research Center, Canada. A total of 20 subjects provided loudness ratings for 96 monophonic sequences. The tests used the same subjective methodology used to create the first dataset, and the same reference signal was also used. The results of these tests formed the second dataset of the subjective database.

The third dataset consisted of loudness ratings for 144 audio sequences. The test sequences consisted of 48 monophonic items, 48 stereo items, and 48 multichannel items. Moreover, one half of the monophonic items were played back via the centre channel (mono), whereas the other half of the monophonic items were played back via the left and right loudspeakers (dual mono). This was done to account for the two different manners in which one might listen to a monophonic signal. For this test, the reference signal consisted of English female speech with stereo ambience and low-level background music. A total of 20 subjects participated in this test which used the loudspeaker configuration specified in Recommendation ITU-R BS.775, and depicted in Fig. 9.

FIGURE 9

Loudspeaker configuration used for the third dataset



The first two datasets were limited to monophonic test sequences and so imaging was not a factor. In the third dataset, which also included stereo and multichannel sequences, imaging was an important consideration that needed to be addressed. It was felt that it was likely that the imaging and ambience within a sequence could have a significant effect on the perceived loudness of the sequence. Therefore, stereo and multichannel sequences were chosen to include a broad range of imaging styles (e.g. centre pan vs. hard left/right, sources in front vs. sources all around) and varying amounts of ambience (e.g. dry vs. reverberant).

The fact that subjects had to simultaneously match the loudness of mono, dual mono, stereo, and multichannel signals meant that this test was inherently more difficult than the previous datasets which were limited to mono signals. This difficulty was furthered by the various imaging styles and varying amounts of ambience. There was some concern that, as a result of these factors, the subjects could be overwhelmed by the task. Fortunately, preliminary tests suggested that the task was manageable, and indeed the 20 subjects were able to provide consistent results.

5 Design of the multichannel loudness algorithm

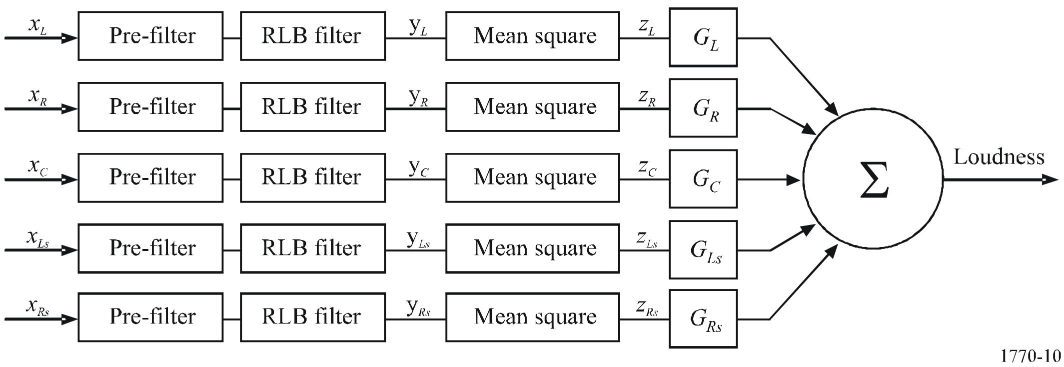
As stated earlier, the *Leq*(RLB) algorithm was designed to operate on monophonic signals, and an earlier study has shown that it is quite successful for this task. The design of a multichannel loudness algorithm brings about several additional challenges. A key requirement for a successful multichannel algorithm is that it must also work well for mono, dual mono, and stereo signals. That is, these formats must be viewed as special cases of a multichannel signal (albeit very common cases).

In the present study we assume that the multichannel signals conform to the standard Recommendation ITU-R BS.775 5.1 channel configuration. No effort is made to account for the LFE channel.

In the multichannel loudness meter, the loudness of each of the individual audio channels is measured independently by a monophonic *Leq*(RLB) algorithm, as shown in Fig. 10. However, a pre-filtering is applied to each channel prior to the *Leq*(RLB) measure.

FIGURE 10

Block diagram of proposed multichannel loudness meter



The purpose of the pre-filter is to account for the acoustic effects that the head has on incoming signals. Here, the head is modelled as a rigid sphere. The same pre-filter is applied to each channel. The resulting loudness values are then weighted (G_i) according to the angle of arrival of the signal, and then summed (in the linear domain) to provide a composite loudness measure. The weightings are used to allow for the fact sounds arriving from behind a listener may be perceived to be louder than sounds arriving from in front of the listener.

A key benefit of the proposed multichannel loudness algorithm is its simplicity. The algorithm is made up entirely of very basic signal processing blocks that can easily be implemented in the time-domain on inexpensive hardware. Another key benefit of the algorithm is its scalability. Since the processing applied to each channel is identical, it is very straightforward to implement a meter that can accommodate any number of channels from 1 to N. Moreover, since the contributions of the individual channels are summed as loudness values, rather than at the signal level, the algorithm does not depend on inter-channel phase or correlation. This makes the proposed loudness measure far more generic and robust.

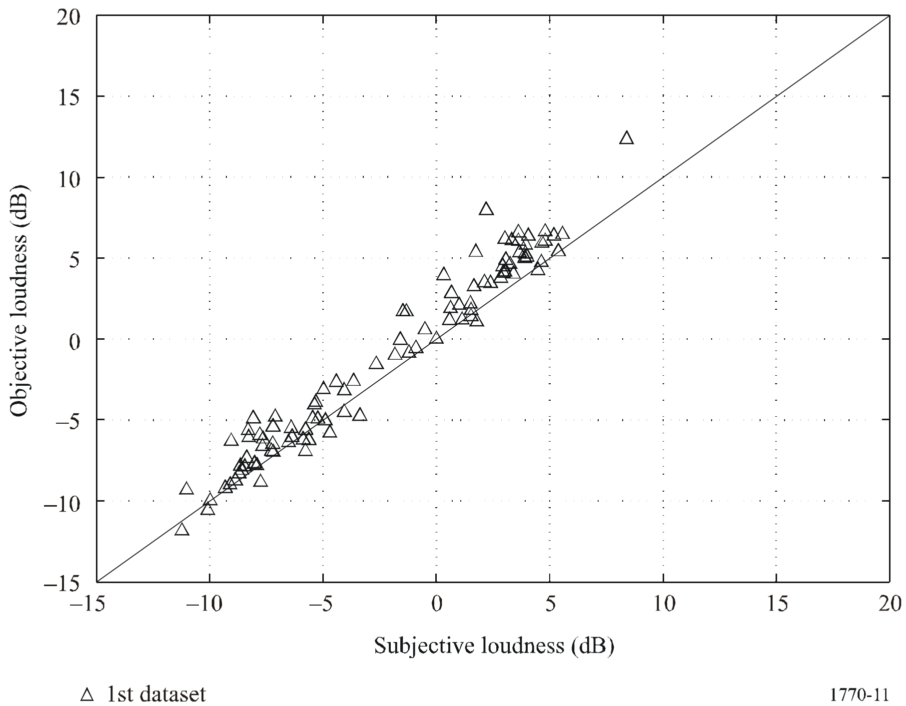
6 Evaluation of the multichannel algorithm

The 336 audio sequences used in the three datasets were processed through the proposed multichannel algorithm and the predicted loudness ratings were recorded. As a result of this process, the overall performance of the algorithm could be evaluated based on the agreement between the predicted ratings and the actual subjective ratings obtained in the formal subjective tests.

Figures 11, 12 and 13 plot the performance of the proposed loudness meter for the three datasets. In each Figure the horizontal axis provides the subjective loudness of each audio sequence in the dataset. The vertical axis indicates the objective loudness predicted by the proposed loudness meter. Each point on the graph represents the result for an individual audio sequence. It should be noted that a perfect objective algorithm would result in all data points falling on the diagonal line having a slope of 1 and passing through the origin (as shown in the Figures).

FIGURE 11

Results for the first (monophonic) dataset (*r* = 0.979)



It can be seen from Fig. 11 that the proposed multichannel loudness algorithm performs very well at predicting the results from the first (monophonic) dataset. The correlation between the subjective loudness ratings and the objective loudness measure is $r = 0.979$.

As seen in Fig. 12, the correlation between the subjective loudness ratings and the objective loudness measure for the second dataset is also very good ($r = 0.985$). It is interesting to note that about one half of the sequences in this dataset were music.

FIGURE 12

Results for the second (monophonic) dataset (*r* = 0.985)

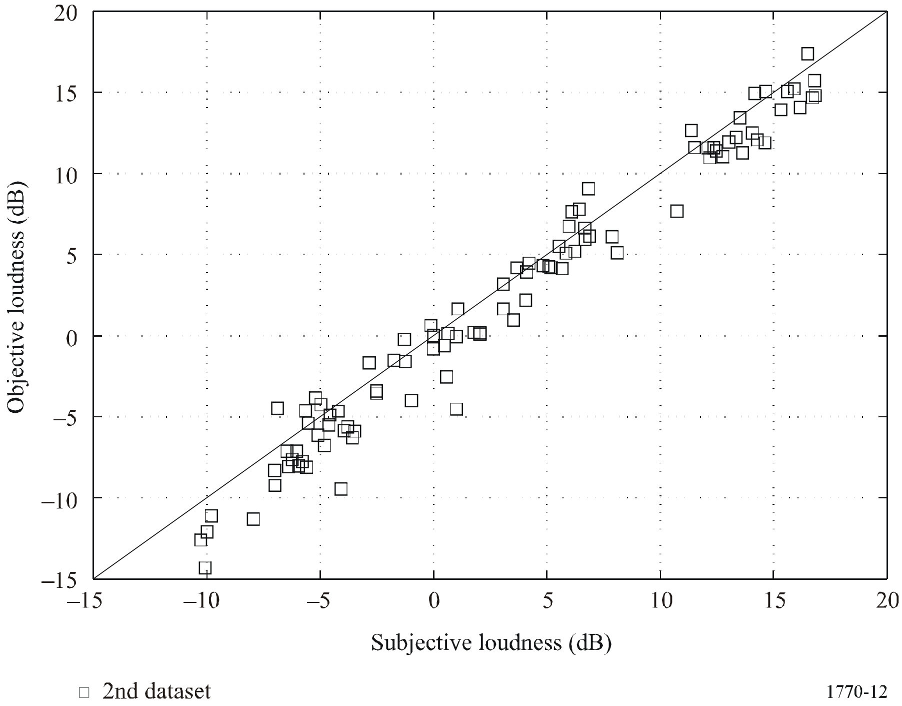


FIGURE 13

Results for the third (mono, stereo and multichannel) dataset (*r* = 0.980)

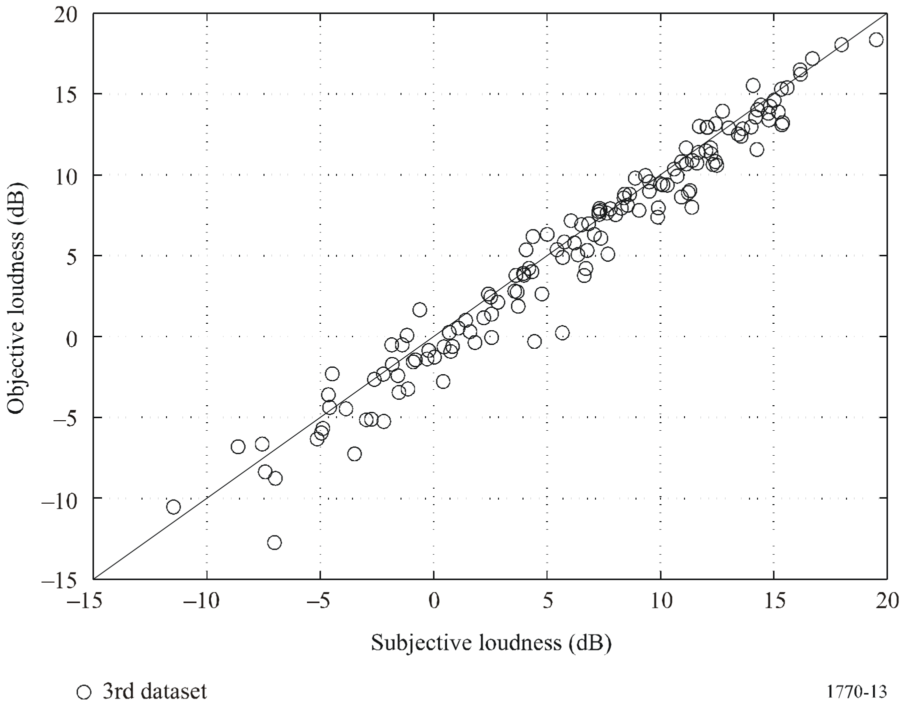
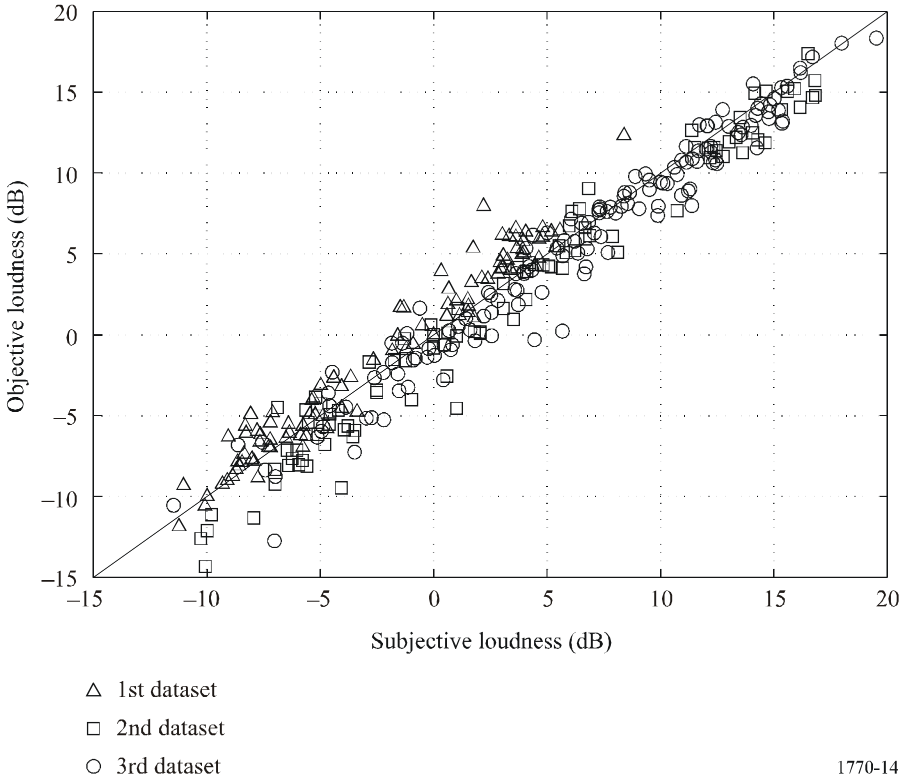


Figure 13 shows the results for the third dataset, which included mono, dual mono, stereo and multichannel signals. The multi-channel results included in Figs. 13 and 14 are for the specified algorithm, but with the surround channel weightings set to 4 dB (original proposal) instead of 1.5 dB (final specification). It has been verified that the change from 4.0 dB to 1.5 dB does not have any significant effect on the results. Once again, the performance of the algorithm is very good, with a correlation of $r = 0.980$.

It is useful to examine the performance of the algorithm for all of the 336 audio sequences that made up the subjective database. Therefore, Fig. 14 combines the results from the three datasets. It can be seen that the performance is very good across the entire subjective database, with an overall correlation of $r = 0.977$.

FIGURE 14

Combined results for all three datasets ($r = 0.977$)



The results of this evaluation indicate that the multichannel loudness measurement algorithm, based on the *Leq*(RLB) loudness measure, performs very well over the 336 sequences of the subjective database. The subjective database provided a broad range of programme material including music, television and movie dramas, sporting events, news broadcasts, sound effects, and advertisements. Also included in the sequences were speech segments in several languages. Moreover, the results demonstrate that the proposed loudness meter works well on mono, dual mono, stereo, as well as multichannel signals.

References

SOULODRE, G.A. [May 2004] Evaluation of Objective Loudness Meters, 116th Convention of the Audio Engineering Society, Berlin, Preprint 6161.

BENJAMIN, E. [October, 2004] Preferred Listening Levels and Acceptance Windows for Dialog Reproduction in the Domestic Environment, 117th Convention of the Audio Engineering Society, San Francisco, Preprint 6233.

Annex 2

Guidelines for accurate measurement of "true-peak" level

This Annex describes an algorithm for estimation of true-peak level within a single channel linear PCM digital audio signal. The discussion that follows presumes a 48 kHz sample rate. True-peak level is the maximum (positive or negative) value of the signal waveform in the continuous time domain; this value may be higher than the largest sample value in the 48 kHz time-sampled domain. The algorithm provides an estimate for the signal as it is, and, optionally, as it would be in the event that some downstream equipment were to remove the DC component of the signal. Optional mild high frequency pre-emphasis in the peak measurement signal path can enable the algorithm to report a higher peak level for high-frequency signals than is actually the case. The purpose for this is that the phase shifts of subsequent signal processing stages (such as Nyquist filters) could cause growth of high frequency signal peaks, and in some applications this feature could be useful to provide further protection from downstream clipping.

1 Summary

The stages of processing are:

1 Attenuate: 12.04 dB attenuation

2 4 × over-sampling

3 Emphasis: Pre-emphasis shelving filter, zero at 14.1 kHz, pole at 20 kHz (optional)

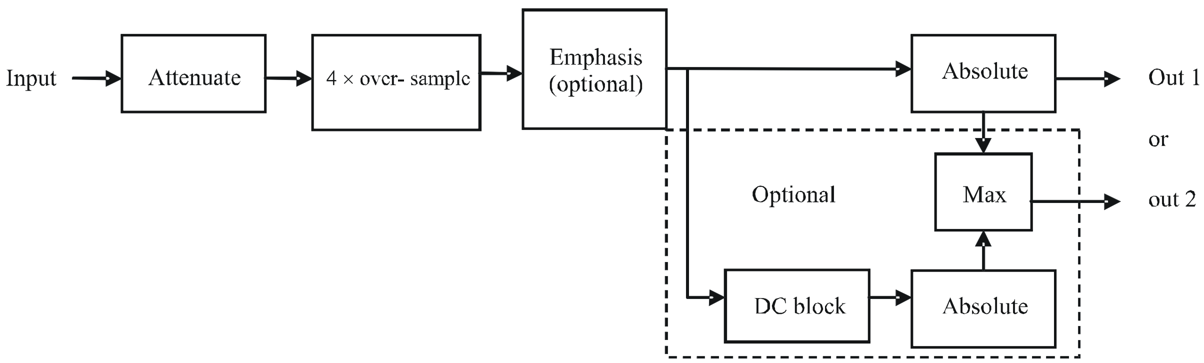
4 DC block (optional)

5 Absolute: Absolute value

6 Max: Highest value detection (optional, included if DC block is included).

Detection of absolute value both before and after the DC block allows estimation of the peak level of the signal at the current point of measurement, as well as estimation of the peak level if at some downstream device the DC component of the signal is removed.

2 Block diagram



1770-15

3 Detailed description

The first step consists of imposing an attenuation of 12.04 dB (2-bit shift). The purpose of this step is to provide for headroom for the subsequent signal processing employing integer arithmetic. This step is not necessary if the calculations are performed in floating point.

The 4 × over-sampling filter increases the sampling rate of the signal from 48 kHz to 192 kHz. This higher sample rate version of the signal more accurately indicates the actual waveform that is represented within the signal. Higher sampling rates and over-sampling ratios are preferred (see Appendix 1 to this Annex). Incoming signals that are at higher sampling rates require proportionately less over-sampling (e.g. for an incoming signal at 96 kHz sample rate a 2 × over-sampling would be sufficient.)

The optional pre-emphasis shelving filter makes the algorithm indicate a higher peak level for the highest frequency signal components. This may be done out of consideration that it is more difficult to measure and control the peak values of the highest frequency signal components due to the dispersion (phase-shift) effects that occur in the numerous Nyquist filters that occur frequently throughout a broadcast signal chain.

The optional DC blocking filter provides coverage for the case where the signal is highly asymmetric, or contains some DC offset. Besides measuring the peak value of the current signal (including the asymmetry and/or DC offset), inclusion of this optional section enables measurement of the signal as it would be if some downstream piece of equipment were to implement a DC blocking filter.

The absolute value of the samples is taken by inverting the negative value samples; at this point the signal is unipolar, with negative values replaced by positive values of the same magnitude. Output 1 is the stream of output values if the optional DC block is not implemented.

If the optional DC block is implemented, the "MAX" block selects the larger of each sample out of the two signal paths; in this case the output is taken from Output 2.

Subsequent system blocks (not shown or specified in this document) can compare the output sample values to the nominal 100% peak signal level (1/4 of full scale if 12 dB of attenuation had been applied at the input), yielding an estimation of the true-peak level with respect to digital full scale.

Appendix 1*
to Annex 2

Considerations for accurate peak metering of digital audio signals

What is the problem?

Peak meters in digital audio systems often register "peak-sample" rather than "true-peak".

A peak-sample meter usually works by comparing the absolute (rectified) value of each incoming sample with the meter's current reading; if the new sample is larger it replaces the current reading; if not, the current reading is multiplied by a constant slightly less than unity to produce a logarithmic decay. Such meters are ubiquitous because they are simple to implement, but they do not always register the true-peak value of the audio signal.

---

* NOTE – The following informative text was contributed by AES Standards Working Group SC-02-01 through the Radiocommunication WP 6J Rapporteur on loudness metering.

So using a peak-sample meter where accurate metering of programme peaks is important can lead to problems. Unfortunately, most digital peak meters are peak-sample meters, although this is not usually obvious to the operator.

The problem occurs because the actual peak values of a sampled signal usually occur between the samples rather than precisely at a sampling instant, and as such are not correctly registered by the peak-sample meter.

This results in several familiar peak-sample meter anomalies:

– *Inconsistent peak readings*: It is often noticed that repeatedly playing an analog recording into a digital system with a peak-sample meter produces quite different readings of programme peaks on each play. Similarly, if a digital recording is repeatedly played through a sample-rate converter before metering, registered peaks are likewise different on each play. This is because the sample instants can fall upon different parts of the true signal on each play.

– *Unexpected overloads*: Since sampled signals may contain overloads even when they have no samples at, or even close to, digital full scale, overload indication by a peak-sample meter is unreliable. Overloads may cause clipping in subsequent processes, such as within particular *D*/*A* converters or during sample-rate conversion, even though they were not previously registered by the peak-sample meter (and were even inaudible when monitored at that point).

– *Under-reading and beating of metered tones*: Pure tones (such as line-up tones) close to integer factors of the sampling frequency may under-read or may produce a constantly varying reading even if the amplitude of the tone is constant.

How bad can the problem be?

In general, the higher the frequency of the peak-sample metered signal, the worse the potential error.

For continuous pure tones it is easy to demonstrate, for example, a 3 dB under-read for an unfortunately-phased tone at a quarter of the sampling frequency. The under-read for a tone at half the sampling frequency could be almost infinite; however most digital audio signals do not contain significant energy at this frequency (because it is largely excluded by anti-aliasing filters at the point of *D*/*A* conversion and because "real" sounds are not usually dominated by continuous high frequencies).

Continuous tones which are not close to low-integer factors of the sampling frequency do not under-read on peak-sample meters because the beat frequency (the difference between $n.f_{tone}$ and f_s) is high compared to the reciprocal of the decay rate of the meter. In other words, the sampling instant is close enough to the true-peak of the tone often enough that the meter does not under-read.

However, for individual transients, under-reads are not concealed by that mechanism, so the higher the frequency content of the transient, the larger the potential under-read. It is normal in "real" sound for transients to occur with significant high frequency content, and under-reading of these can commonly be several dBs.

Because real sounds generally have a spectrum which falls off towards higher frequencies, and because this does not change with increasing sampling frequency, peak-sample meter under-read is less severe at higher original sampling frequencies.

What is the solution?

In order to meter the true-peak value of a sampled signal it is necessary to "over-sample" (or "up-sample") the signal, essentially recreating the original signal between the existing samples, and thus increasing the sampling frequency of the signal. This proposal sounds dubious: how can we recreate information which appears already to have been lost? In fact, sampling theory shows that we can do it, because we know that the sampled signal contains no frequencies above half of the original sampling frequency.

What over-sampling ratio is necessary? We need to answer a couple of questions to find out:

– What is the maximum acceptable under-read error?

– What is the ratio of the highest frequency to be metered to the sampling frequency (the maximum "normalized frequency")?

If we know these criteria, it is possible to calculate the over-sampling ratio we need (even without considering yet the detail of the over-sampling implementation) by a straightforward "graph-paper" method. We can simply consider what under-read will result from a pair of samples at the over-sampled rate occurring symmetrically either side of the peak of a sinusoid at our maximum normalized frequency. This is the "worst case" under-read.

So for: over-sampling ratio, n

maximum normalized frequency, f_{norm}

sampling frequency, f_s

we can see that:

the sampling period at the over-sampled rate is $1/n.f_s$

the period of the maximum normalized frequency is $1/f_{norm}.f_s$

so:

the maximum under-read (dB) is $20.\log(\cos(2.\pi.f_{norm}.f_s/n.f_s.2))$

(2 in denominator since we can miss a peak by a maximum of half the over-sampling period)

or:

maximum under-read (in dB) = $20.\log(\cos(\pi.f_{norm}/n))$

This equation was used to construct the following Table, which probably covers the range of interest:

Over-sampling ratio	Under-read (dB) maximum f_{norm} = 0.45	Under-read (dB) maximum f_{norm} = 0.5
4	0.554	0.688
8	0.136	0.169
10	0.087	0.108
12	0.060	0.075
14	0.044	0.055
16	0.034	0.042
32	0.008	0.010

How should a true-peak meter be implemented?

The over-sampling operation is performed by inserting zero-value samples between the original samples in order to generate a data stream at the desired over-sampled rate, and then applying a low-pass "interpolation" filter to exclude frequencies above the desired maximum f_{norm}. If we now operate the peak-sample algorithm on the over-sampled signal, we have a true-peak meter with the desired maximum under-read.

It is interesting to consider the implementation of such an over-sampler. It is usual to implement such the low-pass filter as a symmetrical FIR. Where such filters are used to pass high-quality audio, e.g. in (old-fashioned) over-sampling *D*/*A* converters or in sample-rate converters, it is necessary to calculate a large number of "taps" in order to maintain very low passband ripple, and to achieve extreme stop-band attenuation and a narrow transition band. A long word-length must also be maintained to preserve dynamic range and minimize distortion.

However, since we are not going to listen to the output of our over-sampler, but only use it to display a reading or drive a bar graph, we probably do not have the same precision requirements. So long as the passband ripple, coupled with addition of spurious components from the stop-band, does not degrade the reading accuracy beyond our target, we are satisfied. This reduces the required number of taps considerably, although we may still need to achieve a narrow transition band depending on our maximum normalized frequency target. Similarly the word-length may only need to be sufficient to guarantee our target accuracy down to the bottom of the bar graph, unless accurate numerical output is required to low amplitudes.

So it may be that an appropriate over-sampler (possibly for many channels) could be comfortably implemented in an ordinary low-cost DSP or FPGA, or perhaps in an even more modest processor. On the other hand, over-sampling meters have been implemented using high-precision over-sampling chips intended for *D*/*A* converter use. Whilst this is rather wasteful of silicon and power, the devices are low-cost and readily available.

The simplest way to determine the required number of taps and the tap coefficients for a particular meter specification is to use a recursive FIR filter design programme such as Remez or Meteor.

It may also be a requirement in a peak-meter to exclude the effect of any input DC, since audio meters have traditionally been DC blocked. On the other hand, if we are interested in the true-peak signal value for the purposes of overload elimination, then DC content must be maintained and metered. If required, exclusion of DC can be achieved with low computation power by inclusion of a low-order IIR high-pass filter at the meter's input.

It is sometimes required to meter peak signal amplitude after the application of some type of weighting filter in order to emphasize the effects of certain parts of the frequency band. Implementation is dependent on the nature of the particular weighting filter.