Dolby® Dialogue Intelligence™ Reference Code User's Guide

Issue 1

Confidential Information

Dolby Laboratories Licensing Corporation

Corporate Headquarters

Dolby Laboratories, Inc.
Dolby Laboratories Licensing Corporation
100 Potrero Avenue
San Francisco, CA 94103-4813 USA
Telephone 415-558-0200
Fax 415-863-1373
www.Dolby.com

European Licensing Liaison Office

Dolby Laboratories, Inc.
Wootton Bassett
Wiltshire SN4 8QJ England
Telephone 44-1793-842100
Fax 44-1793-842101

Asia

Dolby Japan K.K.
NBF Higashi-Ginza Square 3F
13–14 Tsukiji 1-Chome, Chuo-ku
Tokyo 104-0045 Japan
Telephone 81-3-3524-7300
Fax 81-3-3524-7389
www.Dolby.co.jp

Issue 1
S11/24282

Table of Contents

List of Figures

List of Tables

Introduction

This document explains how to use the Dolby® Dialogue Intelligence™ reference code. Dolby Laboratories created Dialogue Intelligence to identify the parts of a program that contain dialogue.

Note: Dialogue Intelligence and the Dialogue Intelligence logo are trademarks of Dolby Laboratories. No right to use these trademarks, other Dolby trademarks, or the Dolby trade name is included in this license. Companies who wish to use Dolby trademarks should contact a Dolby Laboratories account manager to obtain the appropriate trademark license.

Accurate loudness estimation is an important element of any broadcast chain. Having an accurate estimate of loudness allows a broadcaster to regulate program loudness, thereby minimizing the annoyance to viewers from shifting loudness levels among channels, programs, advertisements, and other aspects of broadcasting where loudness differences can be detected.

Loudness estimation has been a long-standing challenge for the broadcast industry. Many of the loudness metrics used, such as peak levels and quasi–peak program meters, do not reflect program loudness as perceived by a human listener.

The introduction of ITU-R BS.1770-1 [1] did much to improve the field of loudness measurement. BS.1770-1 specifies a K-weighting filter and an algorithm that allows for the accurate estimation of perceived program loudness. BS.1770-1, however, does not specify which segments of a program should be included when estimating the loudness of a program. For example, consider a program that contains periods of silence. A human listener might disregard these silent periods when assessing the loudness of the program. Conversely, BS.1770-1 includes these periods, thereby producing a lower and less accurate result.

Dolby's approach to this problem is to measure loudness only on the segments of a program that contain dialogue (speech gating). This reflects the fact that:

- Content creators typically set dialogue at a fixed level, and mix other content around the dialogue.
- Viewers typically adjust their television volume control according to the audibility/intelligibility of dialogue.

The use of dialogue as an anchor element is widely recognized in the broadcast industry. As ITU-R BS.1864 states, "one program element that is of concern to the audience in programs that are predominantly dialogue is the loudness of dialogue, and that this should desirably be uniform in internationally exchanged programs [3]." Similarly, ATSC recommended practice A/85 defines an anchor element as a "perceptual loudness reference point," and states that this anchor element is typically dialogue [4].

The value of dialogue as an anchor element has been proven by the track record of Dolby's professional loudness products—such as the LM100 Broadcast Loudness Meter and DP600 Program Optimizer—that have been used to measure and correct the loudness of hundreds of thousands of hours' worth of content.

Dialogue Intelligence is the core piece of technology that facilitates speech gating. Dialogue Intelligence analyzes a program and identifies the segments that contain dialogue. This allows the loudness measurement algorithm to exclude the nondialogue segments from the loudness calculation.

Recently, an alternative gating method, level gating, has emerged from ITU-R BS.1770-2 [2]. Level gating makes no attempt to identify segments containing dialogue. Instead, it uses histogramming techniques to produce a loudness estimate.

Level gating has been shown to be reasonably successful at estimating the loudness of short-form content that may be heavily compressed (for example, advertisements). However, level gating and speech gating can produce significantly different results when applied to long-form content.

ITU-R BS.1864 allows for the selection of a gating method that is appropriate to the content being measured. That gating method might be level gating (BS.1770-2) or speech gating (Dialogue Intelligence).

This reference code provides a reference Implementation of Dialogue Intelligence. Additionally, a conformance test is provided so adopters can confirm the performance of their Implementation.

Operation

This chapter describes the algorithmic operation of Dialogue Intelligence™.

2.1 Overview

The input to Dialogue Intelligence is a single channel of uncompressed audio at a sample rate of 32, 44.1, or 48 kHz. The output from Dialogue Intelligence is a single binary decision variable that indicates whether the current feature frame contains speech.

Dialogue Intelligence is composed of three stages:

- Sample-rate conversion to 16 kHz
- Feature extraction
- Boost classifier

The feature extraction and boost classifier stages operate at a sample rate of 16 kHz. It is the responsibility of the sample-rate converter to ensure that the input sample rate is converted to 16 kHz, and to disregard any audio above the Nyquist frequency (8 kHz). The sample-rate converter operates on a fixed input/output frame size, and therefore requires a delay line to buffer input samples.

The feature extraction stage accepts 16 kHz audio as an input, and generates a feature vector as its output. The feature vector is an array of seven observations, corresponding to seven features that are calculated by Dialogue Intelligence. These seven features are:

- Average squared L2-norm of weighted spectral flux (`SFV`)
- Skew of regressive line of best fit through estimated spectral power density (`AST`)
- Pause count (`PSC`)
- Skew coefficient of zero crossing rate (`ZCS`)
- Mean-to-median ratio of zero crossing rate (`ZCM`)
- Rhythmic measure (`RPM`)
- Long rhythmic measure (`LRM`)

The boost classifier accepts the feature vector as an input, and produces a binary decision variable (with values of `0x01` [speech] and `0x00` [other]) as an output. The boost classifier is based on a boosting machine learning algorithm that combines a set of weak learners (the individual features) into a single strong learner (the decision variable).

Dialogue Intelligence implements a speech/other discriminator that can be used for identifying specific segments of audio that contain speech. A block diagram is shown in Figure 2-1.

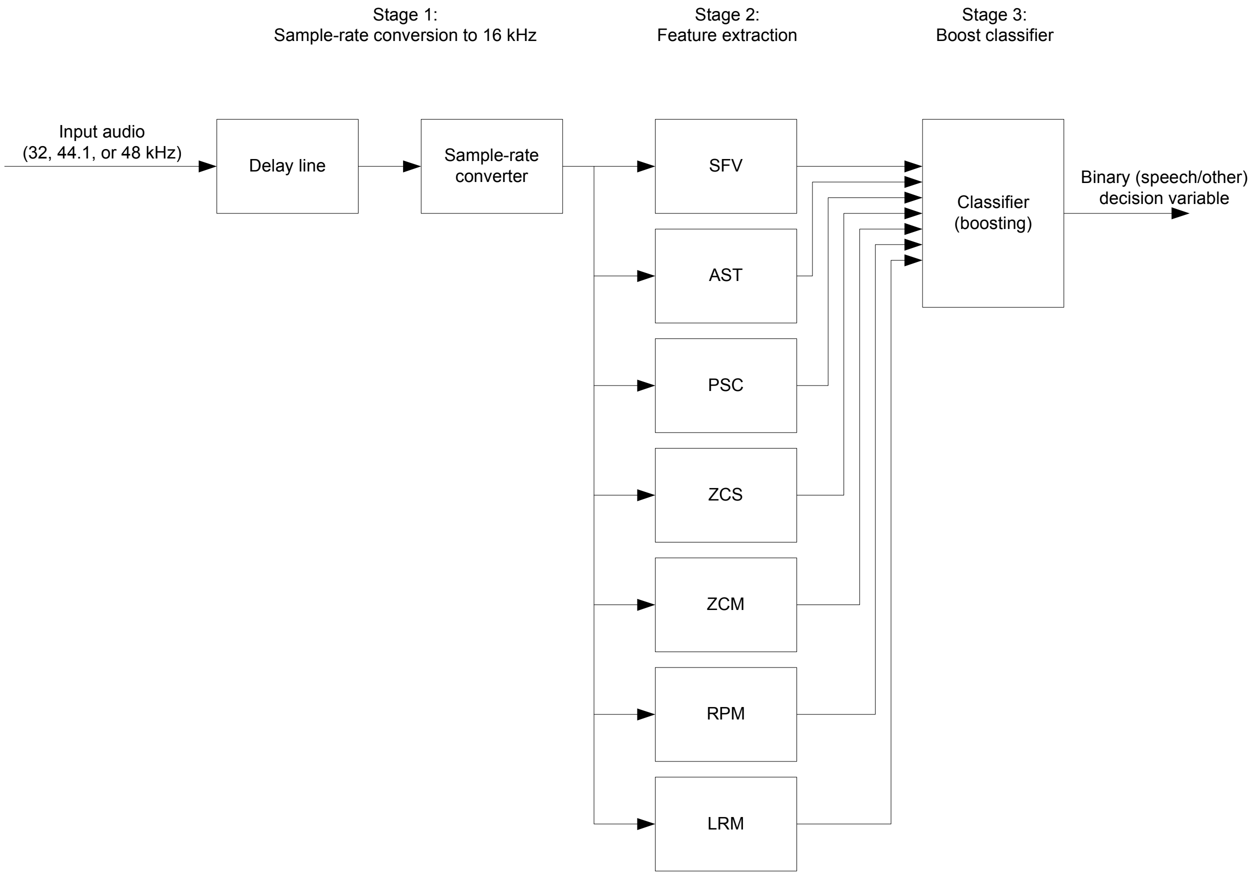

Figure 2-1 Dialogue Intelligence Block Diagram

2.2 Detailed Description

The reference code serves as the primary reference for how Dialogue Intelligence operates.

For an alternative view, Dialogue Intelligence is also described by [5]. However, please note the following corrections and updates since that document was published:

- Chapter 3.1: The frame size is 2,048 ms, not 2,057 ms.
- Chapter 3.1: A 75% overlap between successive feature frames has been introduced since publication. Therefore, the classifier output is updated every 512 ms (instead of every 2,048 ms).
- Chapter 3.1.1: The reference code for the "average squared L2-norm of weighted spectral flux" contains a known issue: audio samples are normalized, but this normalization is never compensated for in the subsequent calculations. While this behavior is unexpected, it was present in the training of Dialogue Intelligence and in implementations of the algorithm (for example, Dolby® LM100). This issue should be carried forward to future implementations of Dialogue Intelligence, as modifying the behavior may invalidate the classifier coefficients. The successful track record of the LM100 and other products utilizing Dialogue Intelligence suggests that this issue is not critical to overall performance.
- Chapter 3.1.2: The "skew of regressive line of best fit through estimated spectral power density" feature disregards blocks that are deemed to be quiet (determined by the sum of the absolute amplitudes).

- Chapter 3.1.6: The autocorrelation calculation is summed with scaled versions of the autocorrelation calculation from prior frames.
- Chapter 3.1.7: The long rhythmic measure feature no longer uses spectral weights. Instead it uses a technique similar to the rhythmic measure feature.
- Chapter 3.2: An accumulation stage has been added to the output of the classifier. The current boost result and the three prior boost results are accumulated. The sign of the sum is used to determine the speech classification.
- Chapter 3.2: The boosting coefficients have been updated since publication.
- Chapter 3: Frames that contain "low energy" are silenced to improve sensitivity performance.

2.3 Frame Sizes and Latency

Automated speech discrimination algorithms, such as Dialogue Intelligence, perform a complex task. Speech discrimination algorithms generally require a significant amount of input audio to analyze so that they may produce reliable outputs. This implies that such algorithms have an inherent latency. This chapter discussed the various frame sizes utilized by Dialogue Intelligence, and specifies the latency through Dialogue Intelligence.

Figure 2-2 illustrates the various frame sizes and update rates employed by Dialogue Intelligence.

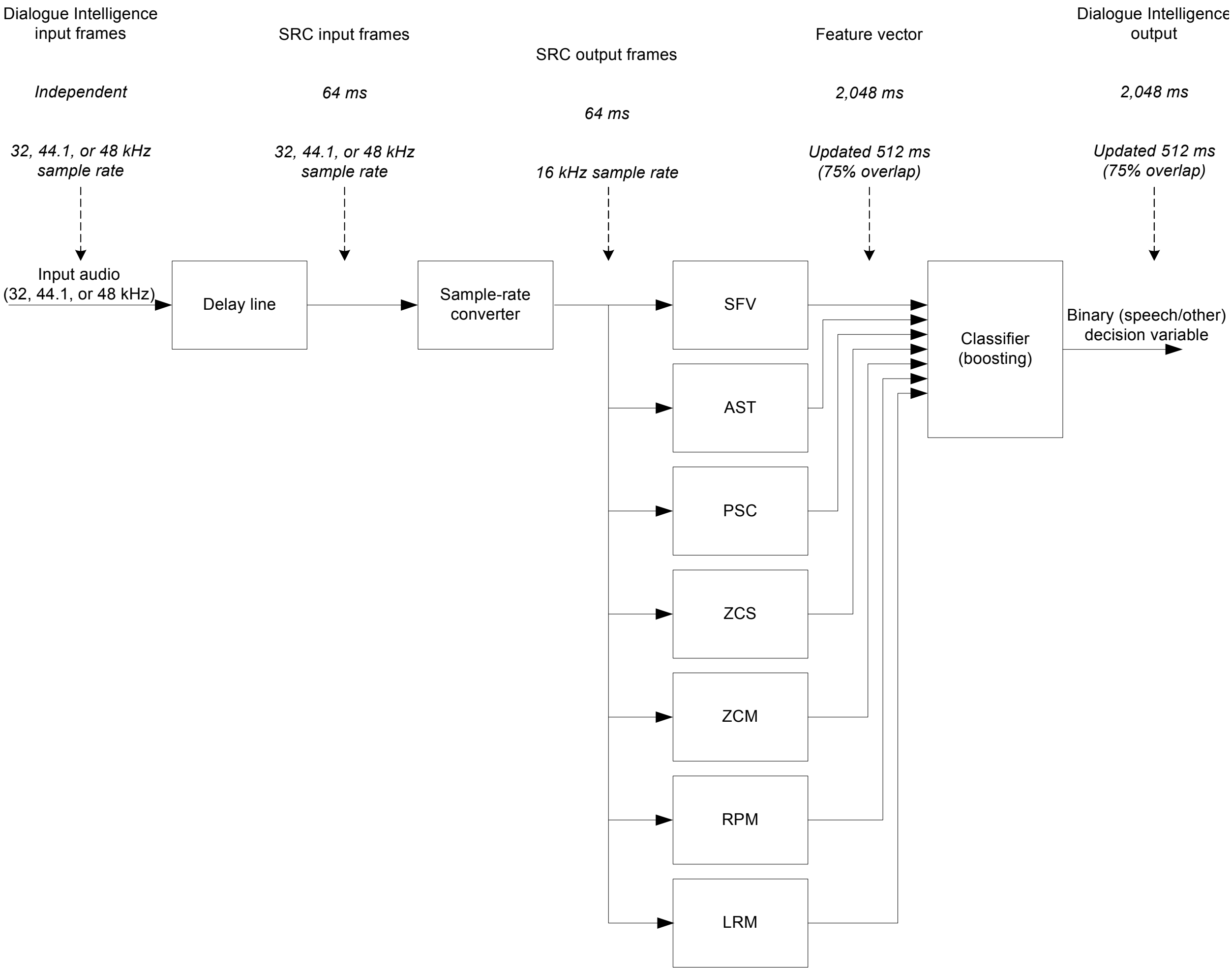

Figure 2-2 Dialogue Intelligence Sample Rates, Frame Sizes, and Update Rates

Dialogue Intelligence accepts any input frame size, and can operate with 32, 44.1, or 48 kHz inputs. As the core of Dialogue Intelligence operates at 16 kHz, the first two stages are a delay line and a sample-rate converter. The sample-rate converter operates on a fixed input/output frame size of 64 ms. The purpose of the delay line is to buffer samples for 64 ms before engaging the sample-rate converter. Additionally, the sample-rate converter has a group delay of 2 ms.

To avoid requiring large amounts of memory, each of the seven features decomposes its calculations into block processing and frame processing.

Block processing is a partial feature extraction on a small block of audio. The output of the block processing is buffered by the feature until it is time to perform frame processing. Each of the seven features uses an independent block size. These are detailed in Table 2-1

Table 2-1 Feature Block Size

Feature	Block Size (Samples)	Block Size (ms)	Blocks per Feature
SFV	1,024	64	32
AST	512	32	64
PSC	256	16	128
ZCS	256	16	128
ZCM	256	16	128
RPM	256	16	128
LRM	256	16	128

Frame processing is the calculation of a feature, representing 2,048 ms of audio, using the outputs from 2,048 ms of block processing with a 75% overlap. The features are updated every 512 ms.

Considering this, the overall latency of Dialogue Intelligence is ultimately determined by the buffering for the feature calculation (2,048 ms) plus the group delay of the sample-rate converter (negligible), resulting in an overall latency of 2,048 ms.

Note that, in practice, the latency can vary by ±512 ms due to the resolution of the Dialogue Intelligence outputs, and the accumulation operation on the output of the boosting algorithm.

Code Organization

This section describes the code aspects of the Dialogue Intelligence™ reference code.

3.1 Organization

The Dialogue Intelligence reference code is provided as C code, compliant to the ISO 9899:1990 standard (also referred to as ANSI C, or C90).

The native data types used by Dialogue Intelligence are specified in Table 3-1.

Table 3-1 Native Data Types

Native Type	Width	Description
`unsigned char`	8 bits	Used for memory manipulation
`int`	32 bits	
`float`	32 bits	IEEE 754 float

The supplied build system generates two components:

- `libdi`: A Dialogue Intelligence library
- `di-test`: A Dialogue Intelligence test application

The library requires certain system library functions, and therefore links against the C standard library as shown in Figure 3-1.

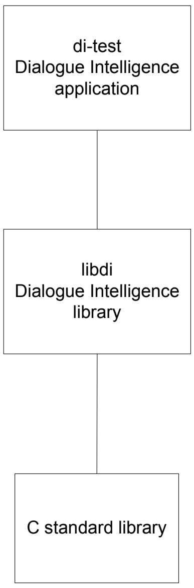

Figure 3-1 Dialogue Intelligence Reference Code Structure

Table 3-2 specifies the C standard library functions required by Dialogue Intelligence.

Table 3-2 Required Standard C Library Functions

Function	Computation Description
`float fabs(float x);`	The absolute value of a floating-point number
`float log10f(float x);`	The base-ten logarithm of a floating-point number
`float log(float x);`	The natural logarithm of a floating-point number
`float sqrt(float x);`	The square root of a floating-point number

Table 3-3 describes the contents of the Dialogue Intelligence reference code, by directory.

Table 3-3 Directory Contents

Directory	Description
doc	Dialogue Intelligence documentation
frontend	Source code for the test application
include	Dialogue Intelligence header files
make	Build systems for building Dialogue Intelligence and test application
src	Source code for Dialogue Intelligence
test	Conformance test materials

3.2 Test Application

The test application `di-test` is a command-line executable that extracts audio from an input PCM file, and then processes the audio using Dialogue Intelligence. The output from Dialogue Intelligence, a binary decision variable, is saved to an output file.

Note: To view the command-line switches, run the command `di-test -h`.

The application accounts for the latency of Dialogue Intelligence. As dialogue does not produce any classification outputs for the first 2,048 ms of input, no outputs are written during this period. Additionally, the application will append a 2,048 ms silent period to the PCM audio data, which allows the final classification results to be extracted from Dialogue Intelligence.

The test application is capable of running the Dialogue Intelligence conformance test specified in Chapter 4. If a reference file is included on the command line, the conformance test will be run.

The input PCM file is a binary file containing a single channel of PCM. The sample format is 16, 20, or 24 bit, and the sampling rate is 32, 44.1, or 48 kHz. Byte order is little endian. 20-bit data, if used, is stored in the top 20 bits of 24-bit words; the bottom 4 bits are set to zero.

The output and reference files are binary files, each containing an array of 8-bit values, one value per input sample. The values are `0x01` (speech) and `0x00` (other).

3.3 Guidance for Implementers

The Dialogue Intelligence reference code is provided as platform independent C code. It is expected that adopters may port this reference code to targets that have implementation constraints (for example, limited speed and limited precision). This chapter provides guidance for how these implementation constraints can be overcome when porting the Dialogue Intelligence reference code.

The three library functions that typically contribute the most to the Dialogue Intelligence computational complexity are the sample-rate converter (`SRC`), the fast Fourier transform (`FFT`) and the delay line (`DLY`). The `SRC` and `FFT` implementations are both platform independent. Replacing these with target-optimized versions may result in a significant speed up. Additionally, the `FFT` function is often called with real-only inputs. A real-input `FFT` may be developed to further reduce the computational complexity. Be cautious if reducing the order of the `SRC`, as performance degradation in the `SRC` may cause the compliance test to fail.

Many systems provide versions of memory management functions (`memset()`, `memcpy()`) that are highly optimized towards their memory architecture. Employing these system functions, especially within `DLY`, may significantly improve the speed of operation.

For guidance, the Dialogue Intelligence reference code has been profiled as running at 112 times faster than real time on a single core of a 32-bit PC, running 32-bit Microsoft® Windows® 7, with a clock speed of 2.93 GHz and 4 GB RAM.

The Dialogue Intelligence reference code is provided as floating-point code. Parties porting the Dialogue Intelligence reference code to fixed-point systems will need to determine the data precision at various points in the Dialogue Intelligence algorithm. The selection of data precision is left to implementers; however, Table 3-4 provides the precision used at key points in one known fixed-point Implementation. (Be aware that intermediate results, such as accumulators, use higher precision.) Implementers are free to select their own precision so long as the conformance test is passed.

Table 3-4 Data Precision in Known Fixed-Point Implementation

Data	Precision
Audio data	24 bit
Sample-rate converter coefficients	
Features	
Boost classifier sum	

Conformance Testing

This chapter provides information on conformance testing for Dialogue Intelligence™.

A single conformance test is defined for Dialogue Intelligence. Parties adopting Dialogue Intelligence are requested to self-certify the behavior of their Implementation by running the conformance test specified in this chapter.

4.1 Test Methodology

The intention of the conformance test is to ensure that the behavior of any implementation of Dialogue Intelligence matches the reference Implementation, with an accuracy of 97%.

The test methodology is illustrated in Figure 4-1. The first input to the conformance test is an audio (PCM) file named di_conf_in.pcm that contains a single channel of raw (binary) audio samples. The sampling rate is 48 kHz, and the sample resolution is 24 bit.

A test application, incorporating Dialogue Intelligence, accepts the PCM as input and passes it through Dialogue Intelligence. Dialogue Intelligence generates a sequence of speech classifications that the test application saves in an output file. The test application ensures that the invalid classifications returned from the first calls to Dialogue Intelligence are not saved in the output file. The test application will append a 2,048 ms silent period to the PCM data so that the final classifications can be extracted and saved.

The second input to the conformance test is the reference file di_conf_out.bin. This reference file contains an array of speech classifications that are the expected classifications from di_conf_in.pcm. The classifications are stored as 8-bit values: `0x01` (speech) and `0x00` (other). di_conf_out.bin contains one classification per sample in di_conf_in.pcm.

The conformance test compares the output of Dialogue Intelligence to the reference file di_conf_in.pcm. The output passes if at least 97% of the Dialogue Intelligence output classifications match the reference classifications.

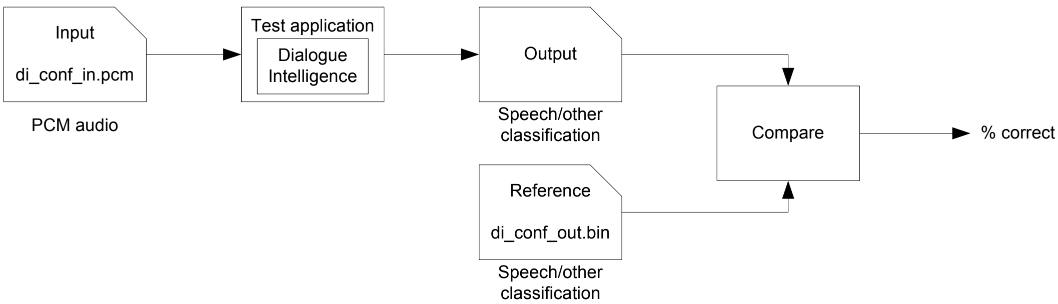

Figure 4-1 Test Methodology

4.2 Test Application Algorithm

This section provides an algorithmic description of the test procedure. An Implementation is provided with the di-test.exe application included with the reference code.

Create the following initialized variables:

- `MATCHES` = 0: The number of matched classifications.
- `TOTAL` = 0: The total number of classifications.
- `ZEROS` = 0: The number of zeros needed to be passed in at end of test.
- `FRAME_SIZE` = `INPUT_FRAME_SIZE`: Value range is 1 to 19,200; the default is 1,024.

Also create the following uninitialized variables:

- `FLUSH_FRAME_SIZE`: Holds the required frame size when flushing Dialogue Intelligence to extract the final classifications
- `OUTPUT`: Current output value
- `REFERENCE`: Current reference value
- `PASS_RATE`: Percentage pass rate

Perform these steps to initialize Dialogue Intelligence and process audio frames:

1. Call `di_init()`.
2. Extract `FRAME_SIZE` contiguous audio samples (or as many as possible) from di_conf_in.pcm to form an input frame of audio samples.
3. Call `di_process()` to process the new audio frame. Assign return value to `OUTPUT`.
4. If `OUTPUT` = `INVALID`, increment `ZEROS` by `FRAME_SIZE`; otherwise, write the 8-bit value `OUTPUT` to the output file di_out.bin, repeating `FRAME_SIZE` times.
5. Check for the end of the input file di_conf.pcm. If not at the end of the file, return to step 2.

Perform this step to flush the final 2,048 ms of results from Dialogue Intelligence:

6. While `ZEROS` > 0:
   a. Set `FLUSH_FRAME_SIZE` to the smaller of `ZEROS` and `FRAME_SIZE`.
   b. Pass a frame of `FLUSH_FRAME_SIZE` zeros to Dialogue Intelligence via the `di_process()` function.
   c. Assign the return value to `OUTPUT`.
   d. Write the 8-bit value `OUTPUT` to the output file di_out.bin, repeating `FLUSH_FRAME_SIZE` times; and decrement `ZEROS` by `FLUSH_FRAME_SIZE`.

Perform these steps to compare the Dialogue Intelligence output against the reference file.

7. Extract one 8-bit classification value from the reference file di_conf_out.bin, and assign to the variable `REFERENCE`.
8. Extract one 8-bit classification value from the output file di_out.bin, and assign to the variable `OUTPUT`.
9. Increment `TOTAL` by 1.
10. Compare `REFERENCE` and `OUTPUT`. If `REFERENCE` equals `OUTPUT`, increment `MATCHES` by 1.
11. Check for the end of files. If not at end of di_conf_out.bin and not at end of di_out.bin, return to step 7.
12. Set `PASS_RATE` to `MATCHES` / `TOTAL` to calculate `PASS_RATE`.
13. Check the result. The result fails if `PASS_RATE` is less than 97%. The result also fails if not at end of di_conf_out.bin and not at end of di_out.bin.

Integration

This chapter provides guidance on how Dolby® Dialogue Intelligence™ can be integrated into a loudness metering or loudness correction product.

5.1 Dialogue Channels

The location of dialogue channels is an important consideration when integrating Dialogue Intelligence into a product. 5.1 programs are commonly mixed with dialogue content in the Center channel, whereas stereo programs often contain dialogue content in both Left and Right channels. There are no firm rules, however, about which channels should contain dialogue, and there are many counter examples to the norm.

Dolby's approach to this issue is to operate Dialogue Intelligence independently on the Center, Left, and Right channels (that is, the channels that normally contain dialogue). Running Dialogue Intelligence on each of these three channels produces three sets of speech/other flags.

This approach is simply adapted to content with fewer channels (for example, mono or stereo) by considering only the relevant channels (for example, by running Dialogue Intelligence on the Left and Right channels for stereo content).

5.2 Latency

As discussed in Section 2.3, Dialogue Intelligence has a latency of 2,048 ms. When Dialogue Intelligence is incorporated into a loudness meter, this latency must be accounted for so that speech gating is correctly aligned with power measurements. See Section 5.4 for a description of how this is achieved when integrating Dialogue Intelligence with ITU-R BS.1770-2.

5.3 Time Scales

Three time scales are commonly used in loudness measurement:

- Momentary: Used for driving meters with a short-time constant, sometimes referred to as bouncing meters
- Short term: Used for making short-term loudness estimates
- Integrated: Used for making long-term loudness estimates (that is, estimates for entire programs)

Level gating is normally applied to the integrated time scale. Similarly, speech gating is also applicable to the integrated time scale.

Unlike level gating, it is possible to create a short-term speech-gated loudness result. Dolby's experience is that a window length of ten seconds is appropriate when producing short-term speech-gated results.

Neither level gating nor speech gating should be applied to momentary time scales.

5.4 Integration with ITU-R BS.1770-2

Dialogue Intelligence is suitable for integration with ITU-R BS.1770-2.

Consider Figure 5-1, the block diagram of the multichannel loudness measurement algorithm from BS.1770-2. This scheme illustrates the algorithm used by BS.1770-2 to measure loudness. The final part of the measurement algorithm is a gate that is used for selecting content to be included in the measurement.

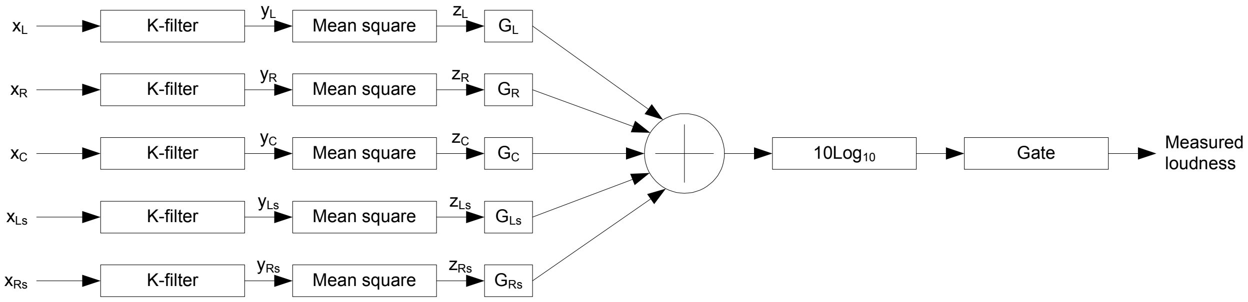

Figure 5-1 Block Diagram of Multichannel Loudness Measurement Algorithm

According to ITU-R BS.1864, which states that a user may select an appropriate gating method, the gate could be a level-based gate, as per BS.1770-2, or a speech gate driven by Dialogue Intelligence. Figure 5-2 illustrates how Dialogue Intelligence is integrated with BS.1770-2 to deliver 5.1 content. The Left, Right, and Center inputs are sent to three separate instances of Dialogue Intelligence. These three instances produce three independent speech/other outputs that are mapped to independent, linear channel gains of 1.0 or 0.0.

The five input channels shown in Figure 5-2 are passed through the same K-filter and mean square process as per BS.1770-2.

The output of the mean square process is split, and the bottom branch is subject to the same measurement algorithm from BS.1770-2 (that is, application of channel gains, summation, conversion to dB, and level gating), but with the addition of a 2,048 ms delay. The level-gating process is identical to that described in BS.1770-2.

The 2,048 ms delay is used to compensate for the latency of the Dialogue Intelligence algorithm. The delay allows all data to be correctly time aligned at key parts of the algorithm.

The Left, Right, and Center outputs from the mean square process are sent to the top branch and delayed by 2,048 ms. Following the delay, linear gains of either 0.0 or 1.0 are applied to each channel. The outputs of the gain stage are summed and converted to dB.

The effect of the gain stage is that when speech is not detected on any channel, all channels will be silenced. Conversely, when speech is detected, those channels that contained speech will be included in the loudness measurement.

Following conversion to dB, the loudness is input to a speech-gating process. The speech-gating process excludes frames that are below –70 LKFS and maintains the integrated (that is, infinite window length), speech-gated loudness estimate. The speech-gating process also accepts a global speech/other indication (equals speech if speech is detected on any channel) and tracks the percentage of a program that contains speech, as a percentage.

As shown in Figure 5-2, two different gating techniques (speech gating and level gating) can be run in parallel. The two gating techniques are not compatible, however, and the output from one should never be fed to the input of the other.

The adaptive gate selection process is responsible for selecting the most appropriate gating method for that piece of content. If a program contains a large amount of dialogue, speech gating is generally the most appropriate gating technique to apply. However, if a program contains limited dialogue, then level gating may be the most appropriate method.

The adaptive gate selection process accepts the speech-gated loudness and level-gated loudness as inputs. It also accepts the speech content percentage, as calculated by the speech-gating block, and a user-configurable threshold value. If the speech content is equal to or exceeds the threshold value, then the adaptive gate selection block will select the speech-gated loudness as its output. Conversely, if the speech content is less than the threshold, the level-gated loudness is selected as the output.

Finally, the adaptive gate selection process provides a gating indication, as an output. This affords users greater transparency, and therefore confidence, in the operation of the loudness meter.

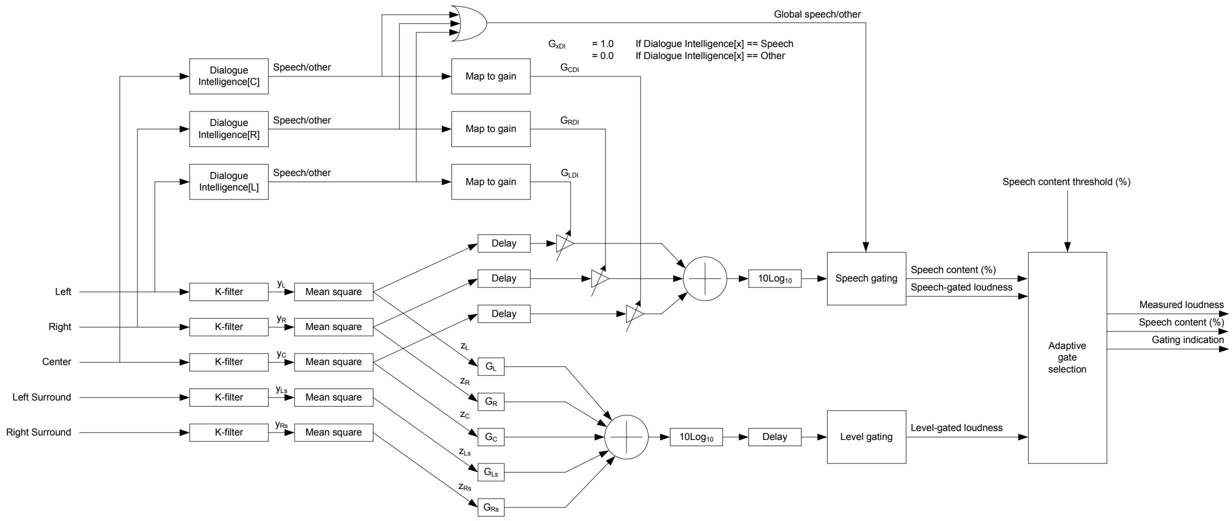

Figure 5-2 Dialogue Intelligence Integrated with BS.1770-2 (5.1 Content)

Figure 5-3 illustrates how Dialogue Intelligence is integrated with BS.1770-2 to deliver stereo content. The difference to note is that only one instance of Dialogue Intelligence is required. The Left and Right inputs are mixed, and the mix is sent to Dialogue Intelligence.

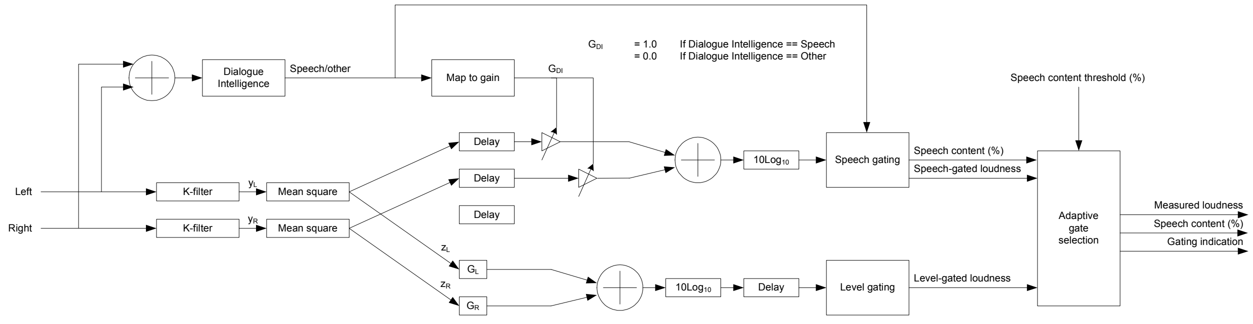

Figure 5-3 Dialogue Intelligence Integrated with BS.1770-2 (Stereo Content)

The Dialogue Intelligence reference code provides a conformance test only for Dialogue Intelligence. There is no conformance test that verifies the integration of Dialogue Intelligence with ITU-R BS.1770-2. However, a loudness meter that correctly integrates ITU-R BS.1770-2 with Dialogue Intelligence will measure the loudness of the mono audio file di_conf_in.pcm (the input file for the Dialogue Intelligence conformance test) as –24 LKFS.

References

[1] ITU Recommendation ITU-R BS.1770-1, *Algorithms to Measure Audio Program Loudness and True-Peak Audio Level*, 2007

[2] ITU Recommendation ITU-R BS.1770-2, *Algorithms to Measure Audio Program Loudness and True-Peak Audio Level,* 2011

[3] ITU Recommendation ITU-R BS.1864, *Operational Practices for Loudness in the International Exchange of Digital Television Programs,* 2010

[4] ATSC A/85:2011, *Recommended Practice: Techniques for Establishing and Maintaining Audio Loudness for Digital Television Document,* 2011

[5] Audio Engineering Society Convention Paper 6437, *Automated Speech/Other Discrimination for Loudness Monitoring,* M Vinton and C Robinson, May 2005