

MMA Technical Standards Board/ AMEI MIDI Committee

Letter of Agreement for Recommend Practice

Three Dimensional Sound Controllers (RP-049)

Originated By: MMA	Reference MMA Item #: 187r1/127	TSBB #: 33
Issue Date: Jul 23 2009	Approval Date: Jan 18 2009	Last Revised: _____
Related Items:		
Agreed to by MMA: _____	_____	Date: _____
<i>Signature</i>	<i>title</i>	
Agreed to by AMEI: _____	_____	Date: _____
<i>Signature</i>	<i>title</i>	

Source: Creative Technology, Ltd. / E-MU Systems (Ensoniq Corporation)

Abstract:

This specification adds three-dimensional (3D) sound controllers to the MIDI standard.

Background:

Three-dimensional sound is defined as audio that the listener perceives as emanating from arbitrary and dynamically changing locations in their surrounding space.

Three-dimensional sound has been widely used in producing and rendering compelling audio content for modern Interactive Audio systems, particularly video game audio on personal computers. The demand for this usage had led the Interactive Audio Special Interest Group (IASIG) to develop industry-standard guidelines in producing APIs and renderers specifically for 3D Interactive Audio.

Modern more economical processors now have the processing power that was previously only available in such large systems, so it now becomes reasonable and feasible to render such content in small embedded systems, such as stand-alone synthesizers or even mobile phones.

With the proliferation of multi-channel systems for home-cinema, video games and music, the need for multi-channel production systems is growing in order to offer to authors the tools to address these new playback configurations.

Also, modern Interactive Audio rendering systems have more processing power than ever before, which makes it feasible to tightly integrate the functionality of synthesis and interactive audio.

With these modern advances, the advancement of the MIDI standard from simple stereo rendering to true three-dimensional sound rendering becomes the next logical step in MIDI evolution.

This specification was initially a request by the membership to create a surround-sound controller. Subsequent discussions led to an early version which allowed that functionality along with complete elevation, azimuth and distance positioning. Some time later this version was produced, adding other controls involved in rendering compelling three-dimensional sound.

There are two schools of thought on how to position objects within a 3D space.

The first method, which is described in the IASIG I3D Level 1 document, states that the object's position is described in X, Y, and Z coordinates, using 32 bit values, which represents position in centimeters. This method makes it more difficult to simply position a sound from front to back from the listener's perspective, since discrete positions in X, Y, and Z coordinates must be used for each movement. Such an approach, characterized as "exocentric" is usually appropriate for games or virtual reality applications, where a 3D scene is rendered for a spectator moving freely in it.

The second method, more widely used in AES documents and papers, uses azimuth angle, elevation angle, and distance to describe the object, much like firing artillery. Here, moving an object around the head of the listener can be as easy as transmitting a 7 bit controller, giving the application full 360 degree positioning control, without sending elevation and distance parameters. This approach, characterized as "egocentric" is therefore appropriate for describing elements evolving relative to the spectator (here the listener), which is often how music is authored: where are the instruments? How loud do they sound at this location for the listener? ...

Because of the advantages cited in the above paragraph, and as discussed in the 1995 proposal, this document adopts the second method as the basis for positioning objects within a 3D space. In addition to azimuth angle, elevation angle and distance, other controllers are introduced to fully support in 3D the existing MIDI pan controller in a backwards compatible manner.

Readers should refer to the following specifications and / or standard publications for more information in the subject matter:

- The Complete MIDI 1.0 Specification
- Interactive Audio Special Interest Group 3D Level 1 Guideline

Publication Plan:

This document (or something similar) should be published on the MMA web site and the contents should be included in the "Channel Voice Messages" section of the next revision of the MIDI 1.0 Detailed Specification.

INTRODUCTION

This specification defines MIDI Registered Parameter Number (RPN) controllers corresponding to the parameters that would be involved in order to allow synthesizers to render MIDI in 3D, and to allow content authors to create compelling 3D MIDI sequences.

All controllers use 14-bit precision, and each controller is designed to offer high level control using general mapping to real world units, allowing the 3D MIDI synthesizer manufacturer the freedom to offer scalable quality of their rendering.

The specification accurately describes appropriate mathematical equations used for computing properties of the controllers' results on lower level parameters such as "gain".

The controllers defined herein are designed to complement standardized MIDI controllers, not to override them. This permits a 3D MIDI synthesizer rendering engine to treat all controllers independently, as they do today. Consequently, the 3D Sound controllers are designed to work as being relative to other similar parameters, which had either been established by MIDI control data or which had been configured in the sound preset data.

For example, parameters defined herein that contribute to "gain" (expressed in dB) should "add" to gain values as set by standard controllers, such as Master Volume and MIDI Controller #7, as well as gain parameters set in the sound preset data (if applicable), to produce a final gain value.

The specification is designed such that it may be made to work with any synthesis model of the manufacturer's choosing. It does not rely on specific synthesizer technology, such as Wavetable synthesis, or on any specific sound set, such as General MIDI, or any specific sound set data format, such as DLS.

The specification makes no assumptions regarding any aspect of the audio output format of the synthesizer, such as speaker layout or the output signal format. Having the three-dimensional sound controllers be agnostic of such details allows the same standard and content to be useful in any conceivable rendering system. Consequently, it becomes the responsibility (and the competitive advantage) of the rendering synthesizer to accept the 3D Sound control data, and render the corresponding audio in the most compelling manner possible, using whatever speaker layout or output CODEC is available.

3D SOUND CONTROLLERS DEFINITION

The preamble defining a 3D Sound Controller is as follows.

B<n> 65 3D

<n> MIDI Channel

The Registered Parameter Number MSB for all controllers described herein is the value 61 (0x3D). Conversely, a “3D Sound Controller” is hereby defined as a Registered Parameter Number controller whose MSB is the value 61 (0x3D).

All 128 Registered Parameter Number LSB values corresponding to 3D Sound Controllers are either defined here, or are to be reserved for 3D Sound Controllers that may be defined in the future.

GENERAL 3D SOUND CONTROLLERS PARAMETER FORMAT

The General Parameter Format for all 3D Sound Controllers is as follows.

B<n> 64 <Param> 06 <Data MSB> 26 <Data LSB>

<n> MIDI Channel
 <Param> 3D Sound Parameter
 <Data MSB> 3D Sound Parameter Value MSB Contribution
 <Data LSB> 3D Sound Parameter Value LSB Contribution

In each controller, both the MSB and LSB contributions are mandatory. Once the LSB contribution is received, the Synthesizer should combine that value with the previously stored MSB contribution for the given parameter, and apply that to the synthesis model.

Parameter Descriptions are offered in the following format:

<MSB/LSB> Data Significance:

Type	MIDI value	RW Value
Min	<00/00>	{s} {value} {unit}
Max	<7F/7F>	{s} {value} {unit}
Step	<00/01>	{s} {value} {unit}
Default	<VV/vv>	{s} {value} {unit}
Except	<XX/xx>	{s} {value} {unit}

<MSB/LSB> represents the Data Entry MSB and LSB respectively.

{value} represents a real world value

{unit} represents the real world unit to which that value applies.

{s} is one of the following:

	Positive	(Value is positive)
-	Negative	(Value is negative)
~	Approximate	(Value listed is approximate)

‘Min’ represents the value of the minimum RPN value, which is always <00/00>

‘Max’ represents the value of the maximum RPN value, which is always <7F/7F>

‘Step’ represents the value of each individual RPN value step, which is always <00/01>.

‘Default’ represents both the MIDI value and the real world value that should be applied to the synthesizer in the Reset All Controller or Power On conditions. Note that different controllers will have different MIDI values in these conditions, which is why they are showed here as <VV/vv>. The values of VV and vv are specified with each controller.

‘Except’ represents a particular MIDI value or range of values that have “exceptional” behavior. Note different controllers will have different MIDI values in these conditions, which is why they are showed here as <XX/xx>.

The values of *XX* and *xx*, as well as the behavior of the exception itself, are specified with each controller.

AZIMUTH ANGLE PARAMETER CONTROLLER

Registered Parameter Number LSB Data Value 0 is used to control Azimuth Angle.

B<n> 64 00 06 <Data MSB> 26 <Data LSB>

<n> MIDI Channel
 <Data MSB> Azimuth Value MSB Contribution
 <Data LSB> Azimuth Value LSB Contribution

<MSB/LSB> Data Significance:

Type	MIDI value	RW Value
Min	<00/00>	-180.00 degrees
Max	<7F/7F>	~179.98 degrees (= 180 - 360/16384 degrees)
Step	<00/01>	~ 0.02 degrees (= 360/16384 degrees)
Default	<40/00>	0.00 degrees

The azimuth is given in the horizontal plane. The default value of 0 (MIDI: <40/00>, decimal: 8192) is in front of the listening position, 90 degrees (MIDI: <60/00>, dec.: 12288) is on the right, -90 degrees (MIDI: <20/00>, dec.: 4096) on the left, and -180 degrees (MIDI <00/00>, dec. 0) behind the listening position.

The actual angle value *a* in degrees is given by the following formula where *d* is the decimal controller value that ranges from 0 to 16383:

$$a = -180 + d \times \frac{360}{16384}$$

ELEVATION ANGLE PARAMETER CONTROLLER

Registered Parameter Number LSB Data Value 1 is used to control Elevation Angle.

B<n> 64 01 06 <Data MSB> 26 <Data LSB>

<n> MIDI Channel
 <Data MSB> Elevation Value MSB Contribution
 <Data LSB> Elevation Value LSB Contribution

<MSB/LSB> Data Significance:

Type	MIDI value	RW Value
Min	<00/00>	-180.00 degrees
Max	<7F/7F>	~179.98 degrees (= 180 - 360/16384 degrees)
Step	<00/01>	~ 0.02 degrees (= 360/16384 degrees)
Default	<40/00>	0.00 degrees

The elevation is given in the vertical plane containing the apparent position of the source (see Technical Note 1, figure 1). The default value of 0 places the sound in the horizontal plane. An elevation of 90 degrees is above the listening position, -90 degrees is under it. Elevation values are coded in [-180, 180] (as opposed to [-90, 90]) in order to facilitate fly-by type trajectories, such as front-to-back and back-to-front movements that don't require an azimuth change. Also this choice allows handling the MIDI bytes for elevation angle in the same manner as the azimuth angle.

GAIN PARAMETER CONTROLLER

Registered Parameter Number LSB Data Value 2 is used to control Gain.

B<n> 64 02 06 <Data MSB> 26 <Data LSB>

<n> MIDI Channel
 <Data MSB> Gain Value MSB Contribution
 <Data LSB> Gain Value LSB Contribution

<MSB/LSB> Data Significance:

Type	MIDI value	RW Value
Min	<00/01>	- 163.82 dB
Max	<7F/7F>	0.00 dB
Step	<00/01>	0.01 dB (1 mB)
Default	<7F/7F>	0.00 dB
Except	<00/00>	- ∞ dB

The gain parameter control offers the MIDI content author a way to control gain using mB (milliBels), as an alternative to the standard MIDI CC#7/11, which offers gain through a mapping curve. This parameter proves to be convenient for computational engines that are biased toward values in real world units.

Note it is important that Maximum be exactly 0 dB.

DISTANCE RATIO PARAMETER CONTROLLER

Registered Parameter Number LSB Data Value 3 is used to control Distance Ratio.

B<n> 64 03 06 <Data MSB> 26 <Data LSB>

<n> MIDI Channel
 <Data MSB> Distance Ratio Value MSB Contribution
 <Data LSB> Distance Ratio Value LSB Contribution

<MSB/LSB> Data Significance:

Type	MIDI value	RW Value
Min	<00/00>	0.00
Max	<7F/7E>	1 - 1/16384
Step	<00/01>	~ 0.000061 (= 1/16384)
Default	<00/10>	~ 0.001 (= 16/16384)
Except	<7F/7F>	1.00

This parameter controls the ratio of the current distance that an object is away from the listener to the maximum distance (see next controller description) that an object may be away from the listener.

Note this parameter can also be interpreted as a distance of up to one kilometer, expressed in steps of 6.1 centimeters, if all other distance based attenuation parameters are kept at their default (reset-all-controller) value.

See Technical Note 2 at the end of this document for more details on this controller.

MAXIMUM DISTANCE PARAMETER CONTROLLER

Registered Parameter Number LSB Data Value 4 is used to control Maximum Distance.

B<n> 64 04 06 <Data MSB> 26 <Data LSB>

<n> MIDI Channel
 <Data MSB> Maximum Distance Value MSB Contribution
 <Data LSB> Maximum Distance Value LSB Contribution

<MSB/LSB> Data Significance:

Type	MIDI value	RW Value
Min	<00/00>	0.00 distance units
Max	<7F/7E>	1000 - 1000/16384 distance units
Step	<00/01>	~ 0.06 distance units (= 1000/16384)
Default	<7F/7F>	1000.00 distance units
Except	<7F/7F>	1000.00 distance units

This parameter controls the maximum distance that an object may be away from the listener. See Technical Note 2 at the end of this document for more details on this controller, and on the distance model in general.

GAIN AT MAXIMUM DISTANCE PARAMETER CONTROLLER

Registered Parameter Number LSB Data Value 5 is used to control Gain at Maximum Distance.

B<n> 64 05 06 <Data MSB> 26 <Data LSB>

<n>	MIDI Channel
<Data MSB>	Gain at Max Distance Value MSB Contribution
<Data LSB>	Gain at Max Distance Value LSB Contribution

<MSB/LSB> Data Significance:

Type	MIDI value	RW Value
Min	<00/00>	- 163.83 dB (= -16384 + 1 mB)
Max	<7F/7F>	0.00 dB
Step	<00/01>	0.01 dB
Default	<51/0F>	- 60.00 dB

This parameter controls the gain at the maximum distance that an object may be away from the listener. See Technical Note 2 at the end of this document for more details on this controller and on the distance model in general.

Note it is important that Maximum be exactly 0 dB.

REFERENCE DISTANCE RATIO PARAMETER CONTROLLER

Registered Parameter Number LSB Data Value 6 is used to control Reference Distance Ratio.

B<n> 64 06 06 <Data MSB> 26 <Data LSB>

<n>	MIDI Channel
<Data MSB>	Reference Distance Ratio MSB Contribution
<Data LSB>	Reference Distance Ratio LSB Contribution

<MSB/LSB> Data Significance:

Type	MIDI value	RW Value
Min	<00/00>	~-0.000061 (= 1/16384)
Max	<7F/7F>	1.0 (= (1+16383)/16384)
Step	<00/01>	~ 0.000061 (= 1/16384)
Default	<00/10>	0.001

This parameter controls the ratio of the distance below which no distance-based attenuation is applied to the maximum possible distance that an object may be away from the listener (as set by the maximum_distance controller). The actual ratio value r is given by the following formula where d is the decimal controller value:

$$r = \frac{1+d}{16384}$$

See Technical Note 2 at the end of this document for more details on this controller.

PAN SPREAD ANGLE PARAMETER CONTROLLER

Registered Parameter Number LSB Data Value 7 is used to control Pan Spread Angle.

B<n> 64 07 06 <Data MSB> 26 <Data LSB>

<n> MIDI Channel
 <Data MSB> Pan Spread Value MSB Contribution
 <Data LSB> Pan Spread Value LSB Contribution

<MSB/LSB> Data Significance:

Type	MIDI value	RW Value
-----	-----	-----
Min	<00/00>	-180.00 degrees
Max	<7F/7F>	~179.98 degrees (= 180 - 360/16384 degrees)
Step	<00/01>	~ 0.02 degrees (= 360/16384 degrees)
Default	<4A/55>	~ 30.00 degrees

The pan spread angle is half the angle of the sector of the stereo field.

See Technical Note 1 at the end of this document for more details on this controller.

See the Azimuth controller definition for the MIDI to real-world value transformation formula.

ROLL ANGLE PARAMETER CONTROLLER

Registered Parameter Number LSB Data Value 8 is used to control Roll Angle.

B<n> 64 08 06 <Data MSB> 26 <Data LSB>

<n> MIDI Channel
 <Data MSB> Roll Value MSB Contribution
 <Data LSB> Roll Value LSB Contribution

<MSB/LSB> Data Significance:

Type	MIDI value	RW Value
-----	-----	-----
Min	<00/00>	-180.00 degrees
Max	<7F/7F>	~179.98 degrees (= 180 - 360/16384 degrees)
Step	<00/01>	~ 0.02 degrees (= 360/16384 degrees)
Default	<40/00>	0.00 degrees

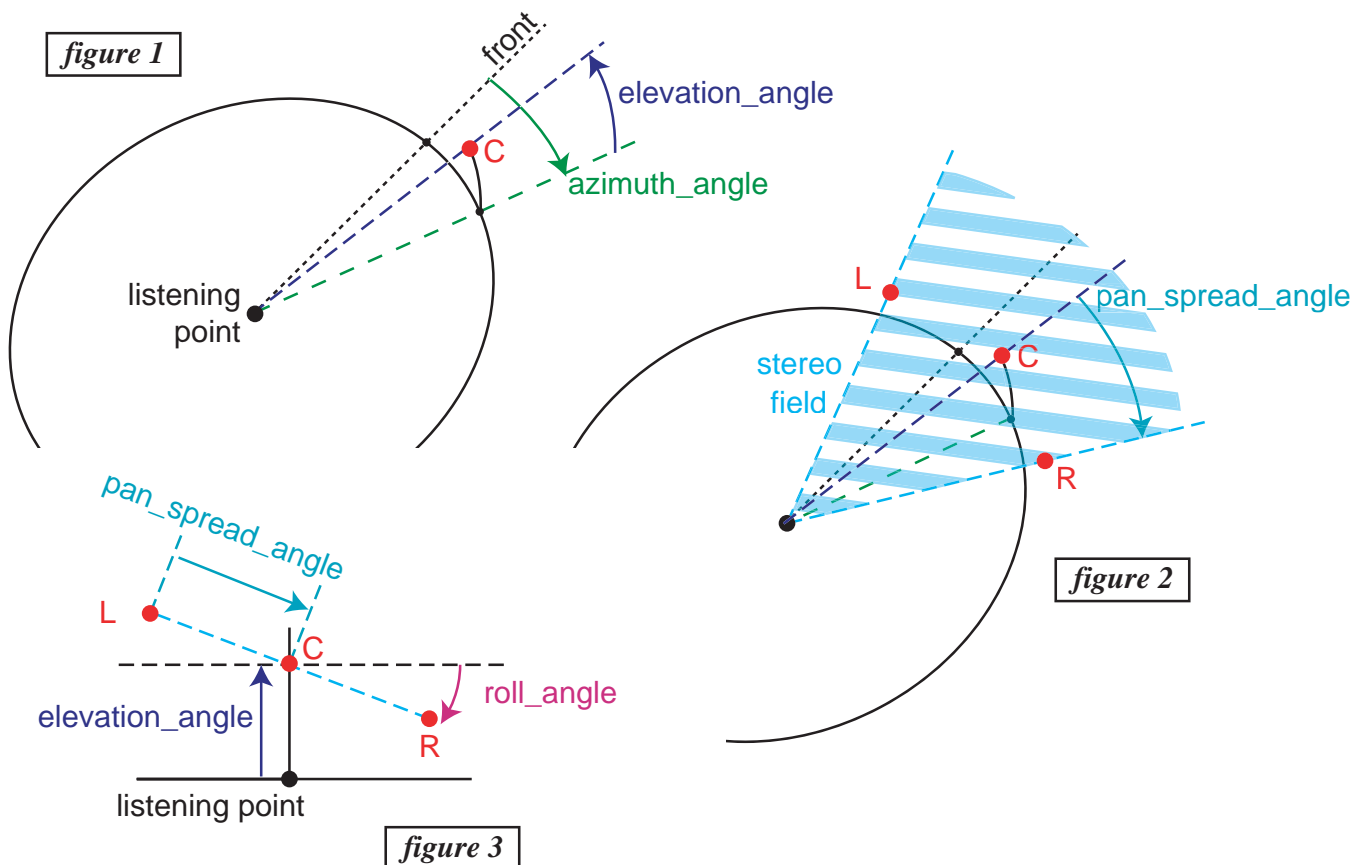
The roll angle is the rotation angle of the stereo field around its median axis. See Technical Note 1 at the end of this document for more details on this controller. See the Azimuth controller definition for the MIDI to real-world value transformation formula.

Technical Notes

The MIDI Pan event is used to set the pan position in the stereo field. This stereo field defines a sector in the horizontal plane originating at the listener's position, facing the listener. The 3D MIDI Controllers other than Gain, permit control of the stereo field by specifying its center and left / right extremities. The technical notes below will elaborate on the usage of the 3D Sound Controllers defined above.

1/ Controlling the stereo field in 3D:

(In the figures below, the stereo field is represented in white and blue stripes, its edges with dashed blue lines, and the left, center, right positions in the stereo field situated at a reference distance of the listener are represented by red dots, respectively labeled L, C, R)



The two controllers *pan_spread_angle* and *roll_angle* allow modifications to the stereo field, the sector in space defined by the following properties:

- Azimuth is defined as the angle between the front of the listener, and the direction from the listener to the center of the stereo field (see figure 1).
- Elevation measures the angle between the horizontal plane and the direction from the listener to the center of the stereo field, measured in the vertical plane (see figures 1 and 3).
- The angular width of the stereo field corresponds to twice the *pan_spread_angle*, as shown in figures 2 and 3 below. Since the *pan_spread_angle* is within $[-180, 180)$, the stereo field can range from a single line in space to a whole plane ("all around the listener").
- The stereo field can be rotated through the *roll_angle* controller. The rotation is made around the vector going from the listening point (the origin) to the center of the stereo field (see figure 3, showing the stereo field as seen from the listening point, looking in the direction of its median).

With the default values of 30 and 0 respectively for the *pan_spread_angle* and *roll_angle*, when azimuth and elevation are at their default values (0 degrees), the L and R positions (thus at azimuths of -30 and 30) coincide with the front speaker layout commonly used in the industry.

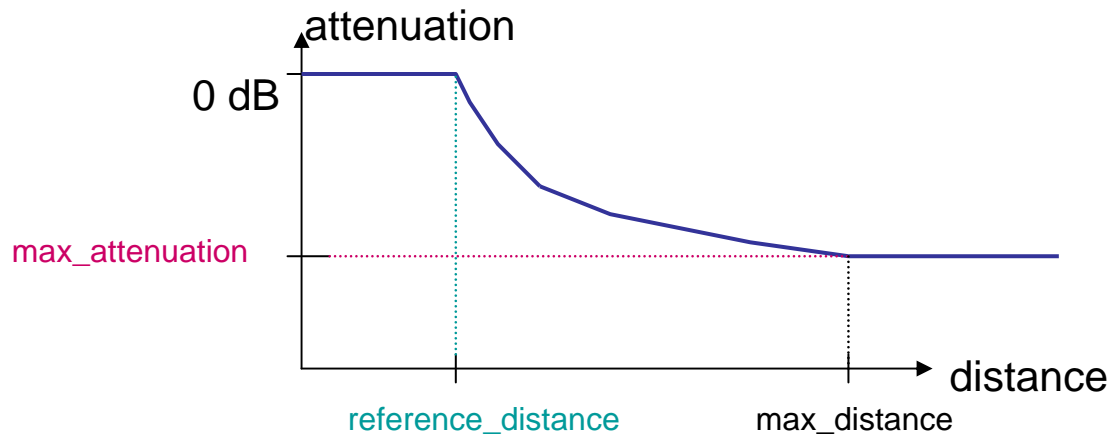
Here are a few examples showing some combinations of pan spread and roll angles:

- *pan_spread_angle* = 30, *roll_angle* = 0 : default situation with a stereo field 60 degrees wide, and L at azimuth -30 (*pan*=0), C at azimuth 0 (*pan*=64) and R at azimuth 30 (*pan*=127).
- *pan_spread_angle* = 30, *roll_angle* = -180 : reverse stereo field compared to a) with L at azimuth 30 and R at azimuth -30.
- *pan_spread_angle* = -180, *roll_angle* = -180 : the stereo field is 360 degrees, L and R coincide behind the listener, and C is at azimuth 0 (*pan*=64).
- *pan_spread_angle* = 0 : the stereo field collapses to the position defined by (azimuth, elevation).

2/ Distance-based attenuation:

This section describes an attenuation model based on the distance between the origin of the spherical coordinate system (the listening position), and the point in space associated with the MIDI channel. As shown in the figure below representing the attenuation according to the distance, this model relies on the following 3 parameters:

1. `max_distance`: the distance at which no additional distance based attenuation is applied when the sound moves further away
2. `reference_distance`: the distance beyond which distance based attenuation is applied, and below which no distance based attenuation is applied
3. `max_attenuation`: the maximum distance based attenuation applied to the sound. It is applied when the sound is at `max_distance`.



The attenuation curve applied when the distance is between `reference_distance` and `max_distance` is defined in this proposal by the model chosen by the IA-SIG for the 3D Audio Rendering and Evaluation guidelines Level 2 (I3DL2). It defines attenuation in dB given by the following formula:

$$attenuation = 20 \log_{10} \left(\frac{reference_distance}{reference_distance + ROF \cdot (distance - reference_distance)} \right)$$

where ROF, the roll off Factor, is a scaling factor that is used to scale the distances beyond the reference distance. This model is also used in OpenAL and Microsoft's DirectSound3D, and is therefore implemented by PC soundcard manufacturers in their implementation of the OpenAL or DirectSound APIs.

With this attenuation curve, a value of 1.0 for the roll off factor (which is the default value in DirectSound) leads for instance to an attenuation of 60 dB when a sound, whose reference distance is 1 meter, is 1000 meters away. This also results in an attenuation of 6 dB for each doubling of the distance. In order to use this attenuation model given the parameters of the proposed model, one would simply need to use a roll off factor given by:

$$ROF = \frac{reference_distance \cdot 10^{\frac{-max_attenuation}{20}} - reference_distance}{max_distance - reference_distance}$$

Based on this distance attenuation model a set of four parameters is proposed for 3D MIDI to encode the description of the attenuation characteristics of a channel, along with the distance it is to be rendered at. The parameters are the following:

1. `maximum_distance`
2. `gain_at_maximum_distance`
3. `distance_ratio`
4. `reference_distance_ratio`

The first parameter, `maximum_distance`, is expressed in units of distance (can be meters) and defines the point where an attenuation of `gain_at_maximum_distance` is applied. In order to provide guaranteed precision for the range of distance in which distance based attenuation is applied, the distance of the source is expressed by the `distance_ratio` parameter as a ratio (between 0 and 1) of the `maximum_distance` parameter. Therefore the actual distance value (as used in the preceding example formulas) is defined by:

$$\text{distance} = \text{distance_ratio} * \text{maximum_distance}$$

The same principle applies to `reference_distance_ratio`, where the actual reference distance is defined by:

$$\text{reference distance} = \text{reference_distance_ratio} * \text{maximum_distance}$$

Here are examples for values of those parameters:

- The buzz of a fly would typically not be heard beyond 10 meters away (`maximum_distance = 10`, `gain_at_maximum_distance = -163.83`) but would sound significantly louder a few centimeters away from your ear (`reference_distance_ratio = 0.01`, which means that between 0 and 10 cm, the fly sound is not attenuated). Here, given the `maximum_distance`, each step to express the distance of the fly with the `distance_ratio` parameter is about 0.6 millimeters.
- The engine of a car will be barely heard a kilometer away (`maximum_distance = 1000`, `gain_at_maximum_distance = -80`) and could be recorded about one meter away (`reference_distance_ratio = 0.001`). Here the `distance_ratio` offers a step of 6 centimeters.

Note that the computation of a roll off factor is only valid when `max_distance` and `reference_distance` are not equal, i.e. when `reference_distance_ratio` is less than 1.0. When `reference_distance_ratio=1.0`, there is no attenuation when `distance_ratio < 1.0`, and the attenuation jumps to `gain_at_maximum_distance` when `distance_ratio = 1.0`.

As a reference for the reader, here is another attenuation scheme found in the literature:

$$\text{attenuation} = 20 \log_{10} \left(\frac{\text{reference_distance}}{\text{distance}} \right)^\alpha$$

A default value of 1 for alpha causes the sound to drop by 6 dB per doubling of the distance, which is what is to expect for the simulation of a point sound source. With an alpha of ½, the attenuation is 3 dB per doubling of the distance, which fits the model of a lineic sound source (such as a river or the waves on the beach).

One can deduct the value of alpha given the parameters of the proposed model with:

$$\alpha = \frac{\text{max_attenuation}}{20 \log_{10} \left(\frac{\text{reference_distance}}{\text{max_distance}} \right)}$$

If the need arises, it can be envisioned that future extensions of 3D MIDI could support multiple distance based attenuation models, whose selection would be triggered by the MIDI content, but the default behavior would be the one defined in this specification.