

Musical Signal Processing with LabVIEW – Sound Spatialization and Reverberation

By:
Ed Doering

Musical Signal Processing with LabVIEW – Sound Spatialization and Reverberation

By:

Ed Doering

Online:

< <http://cnx.org/content/col10485/1.1/> >

C O N N E X I O N S

Rice University, Houston, Texas

This selection and arrangement of content as a collection is copyrighted by Ed Doering. It is licensed under the Creative Commons Attribution 2.0 license (<http://creativecommons.org/licenses/by/2.0/>).

Collection structure revised: November 7, 2007

PDF generated: February 4, 2011

For copyright and attribution information for the modules contained in this collection, see p. 17.

Table of Contents

1 Reverberation	1
2 Schroeder Reverberator	7
3 Localization Cues	13
Index	16
Attributions	17

Chapter 1

Reverberation¹

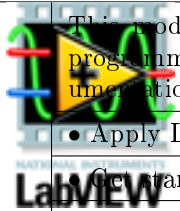
	This module refers to LabVIEW, a software development environment that features a graphical programming language. Please see the LabVIEW QuickStart Guide ² module for tutorials and documentation that will help you:
	<ul style="list-style-type: none">• Apply LabVIEW to Audio Signal Processing
	<ul style="list-style-type: none">• Obtain a fully-functional evaluation edition of LabVIEW

Table 1.1

1.1 Introduction

Reverberation is a property of concert halls that greatly adds to the enjoyment of a musical performance. The on-stage performer generates sound waves that propagate directly to the listener's ear. However, sound waves also bounce off the floor, walls, ceiling, and back wall of the stage, creating myriad copies of the direct sound that are time-delayed and reduced in intensity.

In this module, learn about the concept of reverberation in more detail and ways to emulate reverberation using a digital filter structure known as a **comb filter**.

The Figure 1.1 screencast video continues the discussion by visualizing the sound paths in a reverberant environment. The **impulse response** of the reverberant environment is also introduced. Understanding the desired impulse response is the first step toward emulating reverberation with a digital filter.

Image not finished

Figure 1.1: [video] Sound paths in a reverberant environment and impulse response model

¹This content is available online at <<http://cnx.org/content/m15471/1.2/>>.

²"NI LabVIEW Getting Started FAQ" <<http://cnx.org/content/m15428/latest/>>

1.2 Comb Filter Theory

The **comb filter** is a relatively simple digital filter structure based on a **delay line** and **feedback**. Watch the Figure 1.2 screencast video to learn how you can develop the comb filter structure by considering an idealized version of the impulse response of a reverberant environment.

Image not finished

Figure 1.2: [video] Development of the **comb filter** structure as a way to implement an idealized reverberant impulse response

The **difference equation** of the comb filter is required in order to implement the filter in LabVIEW. In general, a difference equation states the filter output $y(n)$ as a function of the past and present input values as well as past output values.

Take some time now to derive the comb filter difference equation as requested by the following exercise.

Exercise 1.1

(Solution on p. 5.)

Derive the difference equation of the comb filter structure shown at the end of the Figure 1.2 video.

1.3 Comb Filter Implementation

Once the difference equation is available, you can apply the coefficients of the equation to the LabVIEW "IIR" (infinite impulse response) digital filter. The Figure 1.3 screencast video walks through the complete process you need to convert the comb filter difference equation into a LabVIEW VI. The LabVIEW **MathScript node** creates the coefficient vectors. Once the comb filter is complete, its impulse response is explored for different values of delay line length and feedback gain.

Please follow along with the video and create your own version of the comb filter in LabVIEW. Refer to TripleDisplay³ to install the front-panel indicator used to view the signal spectrum.

Image not finished

Figure 1.3: [video] Implementing the comb filter in LabVIEW; exploration of the impulse response as a function of delay line length and feedback gain

1.4 Loop Time and Reverb Time

As you have learned in previous sections, the comb filter behavior is determined by the delay line length N and the feedback coefficient g . From a user's point of view, however, these two parameters are not very intuitive. Instead, the comb filter behavior is normally specified by **loop time** τ and **reverb time** denoted T_{60} . Reverb time may also be written as R_{T60} . Loop time indicates the amount of time necessary for a given sample to pass through the delay line, and is therefore the delay line length N times the sampling

³"[LabVIEW application] TripleDisplay" <<http://cnx.org/content/m15430/latest/>>

interval. The sampling interval is the reciprocal of sampling frequency, so the loop time may be expressed as in (1.1):

$$\tau = \frac{N}{f_S} \quad (1.1)$$

Reverb time indicates the amount of time required for the reverberant signal's intensity to drop by 60 dB (dB = decibels), effectively to silence. Recall from the Figure 1.2 video that the comb filter's impulse response looks like a series of decaying impulses spaced by a delay of N samples; this impulse response is plotted in Figure 1.4 with the independent axis measured in time rather than samples.

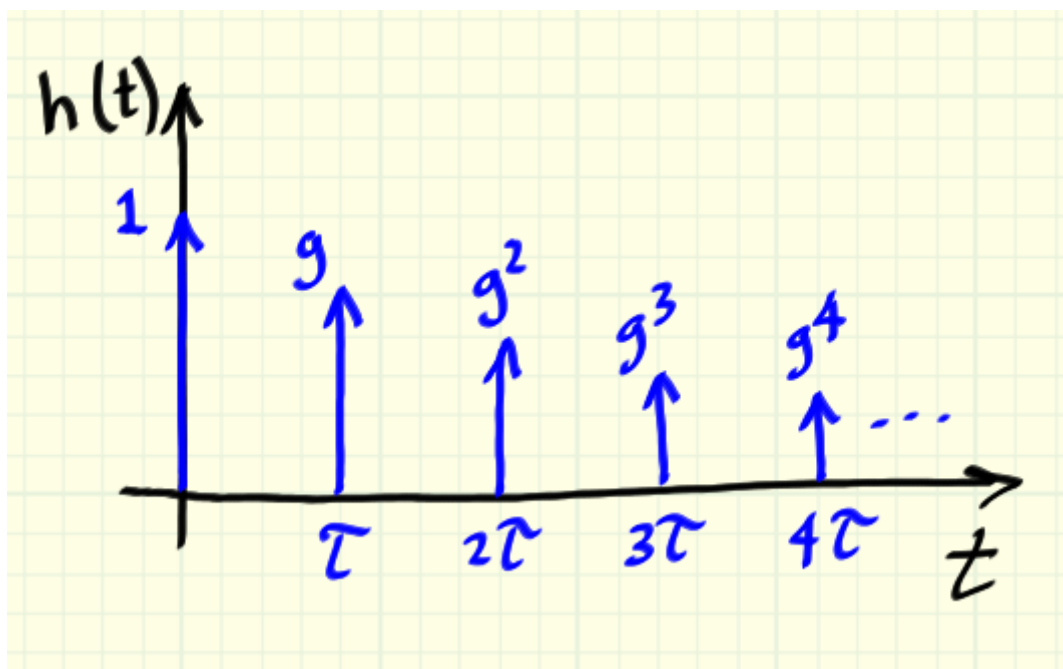


Figure 1.4: Comb filter impulse response

Take a few minutes to derive an equation for the comb filter feedback gain "g" as a function of the loop time and the reverb time. The following pair of exercises guide you through the derivation.

Exercise 1.2

(Solution on p. 5.)

Given the comb filter impulse response pictured in Figure 1.4, derive an equation for the reverb time T_{60} in terms of the loop time τ and the comb filter's feedback constant g . Hint: Recall the basic equation to express the ratio of two amplitudes in decibels, i.e., use the form with a factor of 20.

Exercise 1.3

(Solution on p. 5.)

Based on your previous derivation, develop an equation for the comb filter gain "g" in terms of the desired loop time and reverb time.

Exercise 1.4

(Solution on p. 5.)

To finish up, derive the equation for the comb filter delay "N" in terms of the desired loop time.

Now, return to your own comb filter VI and modify the front-panel controls and LabVIEW MathScript node to use loop time and reverb time as the primary user inputs. Experiment with your modified system to ensure that the spacing between impulses does indeed match the specified loop time, and that the impulse decay rate makes sense for the specified reverb time.

1.5 Comb Filter Implementation for Audio Signals

In this section, learn how to build a comb filter in LabVIEW that can process an audio signal, specifically, an impulse source. Follow along with the Figure 1.5 screencast video to create your own VI. The video includes an audio demonstration of the finished result. As a bonus, the video also explains where the "comb filter" gets its name.

Image not finished

Figure 1.5: [video] Building a LabVIEW VI of a comb filter that can process an audio signal

Next, learn how you can replace the impulse source with an audio .wav file. Speech makes a good test signal, and the Figure 1.6 screencast video shows how to modify your VI to use a .wav audio file as the signal source. The speech clip used as an example in the video is available here: [speech.wav](http://www.voiptroubleshooter.com/open_speech/speech.wav)⁴ (audio courtesy of the Open Speech Repository, www.voiptroubleshooter.com/open_speech⁵; the sentences are two of the many phonetically balanced **Harvard Sentences**, an important standard for the speech processing community).

Image not finished

Figure 1.6: [video] Modifying the LabVIEW VI of a comb filter to process a .wav audio signal

1.6 References

- Moore, F.R., "Elements of Computer Music," Prentice-Hall, 1990, ISBN 0-13-252552-6.
- Dodge, C., and T.A. Jerse, "Computer Music: Synthesis, Composition, and Performance," 2nd ed., Schirmer Books, 1997, ISBN 0-02-864682-7.

⁴<http://cnx.org/content/m15471/latest/speech.wav>

⁵http://www.voiptroubleshooter.com/open_speech

Solutions to Exercises in Chapter 1

Solution to Exercise 1.1 (p. 2)

Solution to Exercise 1.2 (p. 3)

$$T_{60} = \frac{-3\tau}{\log_{10} g}$$

Solution to Exercise 1.3 (p. 3)

$$g = 10^{-\frac{3\tau}{T_{60}}}$$

Solution to Exercise 1.4 (p. 3)

$$N = \tau f_S$$

Chapter 2

Schroeder Reverberator¹

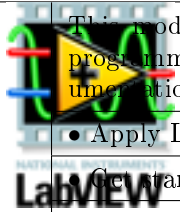
	This module refers to LabVIEW, a software development environment that features a graphical programming language. Please see the LabVIEW QuickStart Guide ² module for tutorials and documentation that will help you:
	• Apply LabVIEW to Audio Signal Processing
	• Get started with LabVIEW
	• Obtain a fully-functional evaluation edition of LabVIEW

Table 2.1

2.1 Introduction

Reverberation is a property of concert halls that greatly adds to the enjoyment of a musical performance. The on-stage performer generates sound waves that propagate directly to the listener's ear. However, sound waves also bounce off the floor, walls, ceiling, and back wall of the stage, creating myriad copies of the direct sound that are time-delayed and reduced in intensity.

In the prerequisite module Reverberation (Chapter 1), you learned how the **comb filter** structure can efficiently create replicas of a direct-path signal that are time delayed and reduced in intensity. However, the comb filter produces replicas that are time delayed by exactly the same amount, leading to the sensation of a pitched tone superimposed on the signal. Refer back to Reverberation (Chapter 1) to hear an audio demonstration of this effect. Put another way, the impulse response of the comb filter contains impulses with identical spacing in time, which is not realistic.

The **Schroeder reverberator** (see "References" section) uses a combination of comb filters and **all-pass filters** to produce an impulse response that more nearly resembles the random nature of a physical reverberant environment.

This module introduces you to the Schroeder reverberator and guides you through the implementation process in LabVIEW. As a preview of what can be achieved, watch the Figure 2.1 screencast video to see and hear a short demonstration of a LabVIEW VI that implements the Schroeder reverberator. The speech clip used in the video is available here: `speech.wav`³ (audio courtesy of the Open Speech Repository, www.voiptroubleshooter.com/open_speech⁴; the sentences are two of the many phonetically balanced **Harvard Sentences**, an important standard for the speech processing community).

¹This content is available online at <http://cnx.org/content/m15491/1.2/>.

²"NI LabVIEW Getting Started FAQ" <http://cnx.org/content/m15428/latest/>

³<http://cnx.org/content/m15491/latest/speech.wav>

⁴http://www.voiptroubleshooter.com/open_speech

Image not finished

Figure 2.1: [video] Demonstration of the **Schroeder reverberator** as implemented in LabVIEW

2.2 Structure of the Schroeder Reverberator

The Figure 2.2 screencast video presents the structure of the Schroeder reverberator and describes the rationale for its design.

Image not finished

Figure 2.2: [video] Structure of the **Schroeder reverberator** and rationale for its design

2.3 All-Pass Filter

The Schroeder reverberator uses **all-pass filters** to increase the pulse density produced by the parallel comb filters. You perhaps are familiar with the frequency response of an all-pass filter: its magnitude response is unity (flat) for all frequencies, and its phase response varies with frequency. For example, the all-pass filter is used to create a variable fractional delay as described in Karplus-Strong Plucked String Algorithm with Improved Pitch Accuracy⁵.

From an intuitive standpoint it would seem that a flat magnitude response in the frequency domain should correspond to an impulse response (time-domain) containing only a single delta function (recall that the impulse function and a constant-valued function constitute a Fourier transform pair). However, the all-pass filter's impulse response is actually a large negative impulse followed by a series of positive decaying impulses.

In this section, derive the transfer function and difference equation of the all-pass filter structure shown in Figure 2.3. Note that the structure is approximately of the same level of complexity as the comb filter: it contains a delay line of N samples and an extra gain element and summing element.

⁵"Karplus-Strong Plucked String Algorithm with Improved Pitch Accuracy" <<http://cnx.org/content/m15490/latest/>>

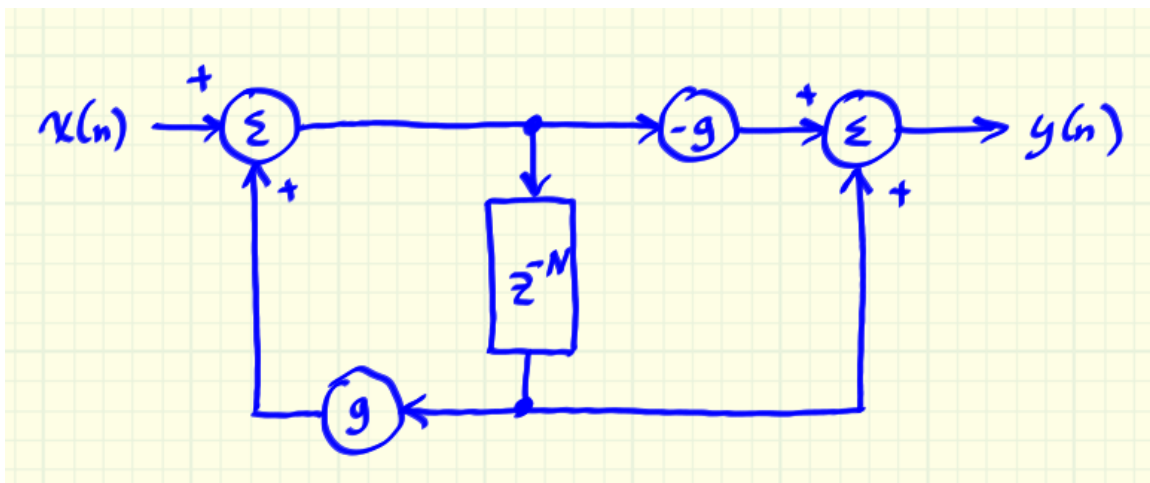


Figure 2.3: All-pass filter structure

Exercise 2.1

(Solution on p. 12.)

Determine the transfer function $H(z)$ for the all-pass filter structure of Figure 2.3.

Exercise 2.2

(Solution on p. 12.)

Based on your result for the previous exercise, write the difference equation for the all-pass filter.

2.4 All-Pass Filter Impulse Response

As described in the video of Figure 2.2, two all-pass filters are placed in cascade (series) with the summed output of the parallel comb filters. An understanding of the all-pass filter impulse response reveals why a cascade connection increases the pulse density of the comb filter in such a way as to emulate the effect of natural reverberation.

The Figure 2.4 screencast video derives the impulse response of the all-pass filter; the loop time and reverb time of the all-pass filter are also presented.

Image not finished

Figure 2.4: [video] Derivation of the all-pass filter impulse response

The Figure 2.5 screencast video demonstrates the sound of the all-pass filter impulse response compared to that of the comb filter. Moreover, the audible effect of increasing the comb filter pulse density with an all-pass filter is also demonstrated in the video.



Download the LabVIEW VI presented in the video: [apfdemo.zip](#)⁶ Refer to TripleDisplay⁷ to install the front-panel indicator required by the VI.

Image not finished

Figure 2.5: [video] Audio demonstration of the all-pass filter impulse response combined with comb filter

2.5 Project: Implement the Schroeder Reverberator in LabVIEW

As described in the Figure 2.2 screencast video, two all-pass filters are placed in cascade (series) with the summed output of four parallel comb filters. The video of Figure 2.4 explains how the all-pass filter "fattens up" each comb filter output impulse with high density pulses that rapidly decay to zero. Selecting mutually-prime numbers for the loop times ensures that the comb filter impulses do not overlap too soon, which further increases the effect of randomly-spaced impulses.

The table in Figure 2.6 lists the required reverb times (T_{60}) and loop times (τ) of the Schroeder reverberator. Note that the comb filters all use the same value (the desired overall reverb time).

Filter	T_{60} (ms)	τ (ms)
CF_1	T_{60}	29.7
CF_2	T_{60}	37.1
CF_3	T_{60}	41.1
CF_4	T_{60}	43.7
APF_1	5.0	96.83
APF_2	1.7	32.92

Figure 2.6: Reverb time and loop time values for the comb filters and all-pass filters of the Schroeder reverberator

The Figure 2.7 screencast video provides everything you need to know to build your own LabVIEW VI for the Schroeder reverberator.

⁶<http://cnx.org/content/m15491/latest/apfdemo.zip>

⁷"[LabVIEW application] TripleDisplay" <<http://cnx.org/content/m15430/latest/>>



Download the .wav reader subVI mentioned in the video : WavRead.vi⁸ .

Image not finished

Figure 2.7: [video] Suggested LabVIEW techniques to build your own Schroeder reverberator

2.6 References

- Schroeder, M.R., and B.F. Logan, "Colorless Artificial Reverberation," Journal of the Audio Engineering Society 9(3):192, 1961.
- Schroeder, M.R., "Natural Sounding Artificial Reverberation," Journal of the Audio Engineering Society 10(3):219-223, 1962.
- Moore, F.R., "Elements of Computer Music," Prentice-Hall, 1990, ISBN 0-13-252552-6.
- Dodge, C., and T.A. Jerse, "Computer Music: Synthesis, Composition, and Performance," 2nd ed., Schirmer Books, 1997, ISBN 0-02-864682-7.

⁸<http://cnx.org/content/m15491/latest/WavRead.vi>

Solutions to Exercises in Chapter 2

Solution to Exercise 2.1 (p. 9)

$$H(z) = \frac{-g + z^{-N}}{1 - gz^{-N}}$$

Solution to Exercise 2.2 (p. 9)

$$y(n) = -gx(n) + x(n - N) + gy(n - N)$$

Chapter 3

Localization Cues¹

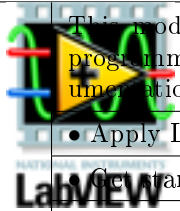
	This module refers to LabVIEW, a software development environment that features a graphical programming language. Please see the LabVIEW QuickStart Guide ² module for tutorials and documentation that will help you:
	<ul style="list-style-type: none">• Apply LabVIEW to Audio Signal Processing
	Get started with LabVIEW
	<ul style="list-style-type: none">• Obtain a fully-functional evaluation edition of LabVIEW

Table 3.1

3.1 Introduction

Our enjoyment of synthesized or recorded sound is greatly enhanced when we perceive the actual locations of the various musicians on stage. In a high quality stereo recording of a jazz trio, for example, you can tell that the drummer is perhaps located slightly to the left of center, the saxophonist is on stage right and the bass player is on stage left. If you have ever flipped on the "mono" (monaural) switch on your stereo amplifier, then you know that the resulting sound is comparatively unpleasant, especially when wearing headphones.

We live in a three-dimensional soundfield, and our ears continually experience slightly different versions of any given sound. These variations allow us to place (or **localize**) the sound source, even when we cannot see it.

In this module, learn about two **localization cues** called **interaural intensity difference** and **interaural timing difference**, abbreviated **IID** and **ITD**, respectively. Each cue relies on presenting a slightly different version of a sound to each ear.

3.2 Interaural Intensity Difference (IID)

The interaural intensity difference (IID) localization cue relies on the fact that an off-center source produces a higher intensity sound in one ear compared to the other. This technique is also called **intensity panning**, and is most effective when the listener is wearing headphones. When a multi-speaker arrangement is used in a room, a given sound propagates to both ears. Lower frequencies have longer acoustic wavelengths and are therefore less directional than higher frequency sounds. Consequently, the IID effect works better at higher frequencies above 1400 Hz (Dodge and Jerse, 1997).

¹This content is available online at <<http://cnx.org/content/m15475/1.2/>>.

²"NI LabVIEW Getting Started FAQ" <<http://cnx.org/content/m15428/latest/>>

The Figure 3.1 screencast video continues the discussion by deriving the equations needed to implement the IID effect using a two-speaker arrangement.

Image not finished

Figure 3.1: [video] Development of the equations needed to implement the **interaural intensity difference (IID)** localization cue

3.3 Interaural Timing Difference (ITD)

The interaural timing difference (ITD) localization cue relies on the fact that sound waves from an off-center source arrives at one ear slightly after the other ear. We can perceive this slight difference in time down to about 20 microseconds (Dodge and Jerse, 1997), and this difference helps us to place the sound source either to the left or right. The ITD cue works best in the lower frequency range of 270 to 500 Hz (Dodge and Jerse, 1997).

The Figure 3.2 screencast video continues the discussion by deriving the equation needed to implement the ITD effect using a two-speaker arrangement.

Image not finished

Figure 3.2: [video] Development of the equation needed to implement the **interaural timing difference (ITD)** localization cue

3.4 Project: Implement the IID and ITD Localization Cues in LabVIEW

You can easily experiment with both the IID and ITD localization cues in LabVIEW. The cues are probably easier to perceive when you choose a speech signal for your source.

The ITD cue requires a delay or time shift between the two stereo channels. The delay line can be constructed as a digital filter with the transfer function shown in :

$$H(z) = \frac{z^{-N}}{1} \quad (3.1)$$

Stated as a difference equation, the filter is $y(n) = x(n - N)$, which states that the output is the same as the input but delayed by N samples. The forward (or "b") coefficients are therefore zero for all but $b_N = 1$, and the reverse (or "a") coefficients are zero for all but $a_0 = 1$.



The Figure 3.3 screencast video provides LabVIEW coding tips to implement the equations and to generate a stereo audio signal.

Image not finished

Figure 3.3: [video] LabVIEW coding tips for the IID and ITD localization cues

3.5 References

- Dodge, C., and T.A. Jerse, "Computer Music: Synthesis, Composition, and Performance," 2nd ed., Schirmer Books, 1997, ISBN 0-02-864682-7

Index of Keywords and Terms

Keywords are listed by the section with that keyword (page numbers are in parentheses). Keywords do not necessarily appear in the text of the page. They are merely associated with that section. *Ex.* apples, § 1.1 (1) **Terms** are referenced by the page they appear on. *Ex.* apples, 1

- A** all-pass filter, § 2(7)
all-pass filters, 7, 8
audio, § 1(1)
- C** comb filter, § 1(1), 1, 2, 2, § 2(7), 7
- D** delay line, § 1(1), 2, § 3(13)
difference equation, 2
- E** echo, § 1(1)
- F** feedback, 2
- H** Harvard Sentences, 4, 7
- I** IID, 13
impulse response, § 1(1), 1, § 2(7)
- intensity panning, 13
- interaural intensity difference, § 3(13), 13, 14
- interaural timing difference, § 3(13), 13, 14
- ITD, 13
- L** LabVIEW, § 1(1), § 2(7), § 3(13)
localization cues, § 3(13), 13
localize, 13
loop time, § 1(1), 2
- M** MathScript node, 2
- R** reverb time, § 1(1), 2
reverberation, § 1(1), 1, § 2(7), 7
- S** Schroeder reverberator, 7, 8, 8

Attributions

Collection: *Musical Signal Processing with LabVIEW – Sound Spatialization and Reverberation*

Edited by: Ed Doering

URL: <http://cnx.org/content/col10485/1.1/>

License: <http://creativecommons.org/licenses/by/2.0/>

Module: "Reverberation"

By: Ed Doering

URL: <http://cnx.org/content/m15471/1.2/>

Pages: 1-5

Copyright: Ed Doering

License: <http://creativecommons.org/licenses/by/2.0/>

Module: "Schroeder Reverberator"

By: Ed Doering

URL: <http://cnx.org/content/m15491/1.2/>

Pages: 7-12

Copyright: Ed Doering

License: <http://creativecommons.org/licenses/by/2.0/>

Module: "Localization Cues"

By: Ed Doering

URL: <http://cnx.org/content/m15475/1.2/>

Pages: 13-15

Copyright: Ed Doering

License: <http://creativecommons.org/licenses/by/2.0/>

Musical Signal Processing with LabVIEW – Sound Spatialization and Reverberation

Reverberation is a property of concert halls that greatly adds to the enjoyment of a musical performance. Sound waves propagate directly from the stage to the listener, and also reflect from the floor, walls, ceiling, and back wall of the stage to create myriad copies of the direct sound that are time-delayed and reduced in intensity. Learn how to model a reverberant environment using comb filters and all-pass filters, and how to implement these digital filters in LabVIEW to create an audio signal processor that can add reverberation an audio signal. In addition, learn how to place a virtual sound source in a stereo sound field using interaural intensity difference (IID) and interaural timing difference (ITD) localization cues. This course is part of the series "Musical Signal Processing with LabVIEW."

About Connexions

Since 1999, Connexions has been pioneering a global system where anyone can create course materials and make them fully accessible and easily reusable free of charge. We are a Web-based authoring, teaching and learning environment open to anyone interested in education, including students, teachers, professors and lifelong learners. We connect ideas and facilitate educational communities.

Connexions's modular, interactive courses are in use worldwide by universities, community colleges, K-12 schools, distance learners, and lifelong learners. Connexions materials are in many languages, including English, Spanish, Chinese, Japanese, Italian, Vietnamese, French, Portuguese, and Thai. Connexions is part of an exciting new information distribution system that allows for **Print on Demand Books**. Connexions has partnered with innovative on-demand publisher QOOP to accelerate the delivery of printed course materials and textbooks into classrooms worldwide at lower prices than traditional academic publishers.