Outline

• The Reverb Problem
• Reverb Perception
• Early Reflections
• Late Reverb
• Schroeder Reverbs
• Feedback Delay Network (FDN) Reverberators
• Waveguide Reverberators
Reverberation Transfer Function

- Three sources
- One listener (two ears)
- Filters should include *pinnae filtering* (*spatialized* reflections)
- Filters change if *anything* in the room changes

In principle, this is an exact computational model.
Implementation

Let $h_{ij}(n) =$ impulse response from source $j$ to ear $i$. Then the output is given by six convolutions:

$$y_1(n) = (s_1 \ast h_{11})(n) + (s_2 \ast h_{12})(n) + (s_3 \ast h_{13})(n)$$

$$y_2(n) = (s_1 \ast h_{21})(n) + (s_2 \ast h_{22})(n) + (s_3 \ast h_{23})(n)$$

- For small $n$, filters $h_{ij}(n)$ are sparse
- Tapped Delay Line (TDL) a natural choice

Transfer-function matrix:

$$\begin{bmatrix} Y_1(z) \\ Y_2(z) \end{bmatrix} = \begin{bmatrix} H_{11}(z) & H_{12}(z) & H_{13}(z) \\ H_{21}(z) & H_{22}(z) & H_{23}(z) \end{bmatrix} \begin{bmatrix} S_1(\bar{z}) \\ S_2(\bar{z}) \\ S_3(\bar{z}) \end{bmatrix}$$
Complexity of Exact Reverberation

Reverberation time is typically defined as $t_{60}$, the time, in seconds, to decay by 60 dB.

**Example:**

- Let $t_{60} = 2$ seconds
- $f_s = 50$ kHz
- Each filter $h_{ij}$ requires 100,000 multiplies and additions per sample, or 5 billion multiply-adds per second.
- Three sources and two listening points (ears) $\Rightarrow$ 60 billion operations per second
  - 20 dedicated CPUs clocked at 3 Gigahertz
  - multiply and addition initiated each clock cycle
  - no wait-states for parallel input, output, and filter coefficient accesses
- FFT convolution is faster, if throughput delay is tolerable (and there are low-latency algorithms)

**Conclusion:** Exact implementation of point-to-point transfer functions is generally too expensive for real-time computation.
Possibility of a Physical Reverb Model

In a complete physical model of a room,

- sources and listeners can be moved without affecting the room simulation itself,
- spatialized (in 3D) stereo output signals can be extracted using a “virtual dummy head”

How expensive is a room physical model?

- Audio bandwidth = 20 kHz \(\approx\) 1/2 inch wavelength
- Spatial samples every 1/4 inch or less
- A 12’x19’x8’ room requires > 200 million grid points
- A lossless 3D finite difference model requires one multiply and 6 additions per grid point \(\Rightarrow\) 60 billion additions per second at \(f_s = 50\) kHz
- A 100’x50’x20’ concert hall requires more than 3 quadrillion operations per second

Conclusion: Fine-grained physical models are too expensive for real-time computation, especially for large halls.
Perceptual Aspects of Reverberation

Artificial reverberation is an unusually interesting signal processing problem:

- “Obvious” methods based on physical modeling or input-output modeling are too expensive
- We do not perceive the full complexity of reverberation
- What is important perceptually?
- How can we simulate only what is audible?
Perception of Echo Density and Mode Density

- For typical rooms
  - Echo density increases as \( t^2 \)
  - Mode density increases as \( f^2 \)
- Beyond some time, the echo density is so great that a stochastic process results
- Above some frequency, the mode density is so great that a random frequency response results
- There is no need to simulate many echoes per sample
- There is no need to implement more resonances than the ear can hear
Proof that Echo Density Grows as Time Squared

Consider a single spherical wave produced from a point source in a rectangular room.

- Tessellate 3D space with copies of the original room
- Count rooms intersected by spherical wavefront
Proof that Mode Density Grows as Freq. Squared

The resonant modes of a rectangular room are given by

\[ k^2(l, m, n) = k_x^2(l) + k_y^2(m) + k_z^2(n) \]

- \( k_x(l) = l\pi / L_x = l\)th harmonic of the fundamental standing wave in the \( x \)
- \( L_x = \) length of the room along \( x \)
- Similarly for \( y \) and \( z \)
- Mode frequencies map to a uniform 3D Cartesian grid indexed by \((l, m, n)\)
- Grid spacings are \( \pi / L_x, \pi / L_y, \) and \( \pi / L_z \) in \( x,y, \) and \( z, \) respectively.
- Spatial frequency \( k \) of mode \((l, m, n) = \) distance from the \((0,0,0)\) to \((l, m, n)\)
- Therefore, the number of room modes having a given spatial frequency grows as \( k^2 \)

\[ ^1 \text{For a tutorial on vector wavenumber, see Appendix E, section E.6.5, in the text: } \url{http://ccrma.stanford.edu/~jos/pasp/Vector_Wavenumber.html} \]
Early Reflections and Late Reverb

Based on limits of perception, the impulse response of a reverberant room can be divided into two segments

- *Early reflections* = relatively sparse first echoes
- *Late reverberation*—so densely populated with echoes that it is best to characterize the response statistically.

Similarly, the *frequency response* of a reverberant room can be divided into two segments.

- Low-frequency sparse distribution of resonant modes
- Modes packed so densely that they merge to form a *random frequency response* with regular statistical properties
Perceptual Metrics for Ideal Reverberation

Some desirable controls for an artificial reverberator include

- $t_{60}(f) =$ desired reverberation time at each frequency
- $G^2(f) =$ signal power gain at each frequency
- $C(f) =$ “clarity” = ratio of impulse-response energy in early reflections to that in the late reverb
- $\rho(f) =$ inter-aural correlation coefficient at left and right ears

Perceptual studies indicate that reverberation time $t_{60}(f)$ should be independently adjustable in at least three frequency bands.
Energy Decay Curve (EDC)

For measuring and defining reverberation time $t_{60}$, Schroeder introduced the so-called energy decay curve (EDC) which is the tail integral of the squared impulse response at time $t$:

$$\text{EDC}(t) \triangleq \int_{t}^{\infty} h^{2}(\tau)d\tau$$

- EDC($t$) = total signal energy remaining in the reverberator impulse response at time $t$
- EDC decays more smoothly than the impulse response itself
- Better than ordinary amplitude envelopes for estimating $t_{60}$
Energy Decay Relief (EDR)

The energy decay relief (EDR) generalizes the EDC to multiple frequency bands:

\[ \text{EDR}(t_n, f_k) \triangleq \sum_{m=n}^{M} |H(m, k)|^2 \]

where \( H(m, k) \) denotes bin \( k \) of the short-time Fourier transform (STFT) at time-frame \( m \), and \( M \) is the number of frames.

- FFT window length \( \approx 30 - 40 \) ms
- \( \text{EDR}(t_n, f_k) = \) total signal energy remaining at time \( t_n \) sec in frequency band centered at \( f_k \)
Energy Decay Relief (EDR) of a Violin Body
Impulse Response

- Energy summed over frequency within each “critical band of hearing” (Bark band)
- Violin body = “small box reverberator”
Reverb = Early Reflections + Late Reverb

- TDL taps may include lowpass filters (air absorption, lossy reflections)
- Several taps may be fed to late reverb unit, especially if it takes a while to reach full density
- Some or all early reflections can usually be worked into the delay lines of the late-reverberation simulation (transposed tapped delay line)
Early Reflections

The “early reflections” portion of the impulse response is defined as everything up to the point at which a statistical description of the late reverb becomes appropriate

- Often taken to be the first 100ms
- Better to test for *Gaussianness*
  - *Histogram* test for sample amplitudes in 10ms windows
  - *Exponential fit* ($t_{60}$ match) to EDC (Prony’s method, matrix pencil method)
  - *Crest factor* test (peak/rms)
- Typically implemented using *tapped delay lines* (TDL) (suggested by Schroeder in 1970 and implemented by Moorer in 1979)
- Early reflections should be *spatialized* (Kendall)
- Early reflections influence *spatial impression*
Late Reverberation

Desired Qualities:

1. a smooth (but not too smooth) decay, and
2. a smooth (but not too regular) frequency response.

• Exponential decay no problem
• Hard part is making it smooth
  – Must not have “flutter,” “beating,” or unnatural irregularities
  – Smooth decay generally results when the echo density is sufficiently high
  – Some short-term energy fluctuation is required for naturalness

• A smooth frequency response has no large “gaps” or “hills”
  – Generally provided when the mode density is sufficiently large
  – Modes should be spread out uniformly
  – Modes may not be too regularly spaced, since audible periodicity in the time-domain can result
• Moorer’s ideal late reverb: *exponentially decaying white noise*
  – Good smoothness in both time and frequency domains
  – High frequencies need to decay faster than low frequencies

• Schroeder’s rule of thumb for echo density in the late reverb is 1000 echoes per second or more

• For impulsive sounds, 10,000 echoes per second or more may be necessary for a smooth response
Typically, $g = 0.7$

Delay-line lengths $M_i$ mutually prime, and span successive orders of magnitude e.g., 1051, 337, 113

Allpass filters in series are allpass

Each allpass expands each nonzero input sample from the previous stage into an entire infinite allpass impulse response

Allpass sections may be called “impulse expanders”, “impulse diffusers” or simply “diffusers”

NOT a physical model of diffuse reflection, but single reflections are expanded into many reflections, which is qualitatively what is desired.
Why Allpass?

• Allpass filters do not occur in natural reverberation!

• “Colorless reverberation” is an idealization only possible in the “virtual world”

• **Perceptual factorization:**
  Coloration now orthogonal to decay time and echo density
Are Allpass Filters Really Colorless?

- Allpass impulse response only “colorless” when extremely short (less than 10 ms or so).
- Long allpass impulse responses sound like feedback comb-filters
- The difference between an allpass and feedback-comb-filter impulse response is *one echo*

\[
\begin{align*}
H(z) &= \frac{0.7 + z^{-7}}{1 + 0.7z^{-7}} \\
H(z) &= \frac{1}{1 + 0.7z^{-7}}
\end{align*}
\]

- Steady-state tones (sinusoids) really do see the same gain at every frequency in an allpass, while a comb filter has widely varying gains.
A Schroeder Reverberator called JCRev

Classic Schroeder reverberator JCRev.

JCRev was developed by John Chowning and others at CCRMA based on the ideas of Schroeder.

- Three Schroeder allpass sections:

\[
\text{AP}_N^g \triangleq \frac{g + z^{-N}}{1 + g z^{-N}}
\]

- Four feedforward comb-filters (STK uses FBCFs):

\[
\text{FFCF}_N^g \triangleq g + z^{-N}
\]
• Schroeder suggests a progression of delays close to

\[ M_i T \approx \frac{100 \text{ ms}}{3^i}, \quad i = 0, 1, 2, 3, 4. \]

• Comb filters impart distinctive coloration:
  • Early reflections
  • Room size
  • Could be one tapped delay line

• Usage: Instrument adds scaled output to RevIn

• Reverberator output RevOut goes to four delay lines
  • Four channels decorrelated
  • Imaging of reverberation between speakers avoided

• For stereo listening, Schroeder suggests a mixing matrix at the reverberator output, replacing the decorrelating delay lines

• A mixing matrix should produce maximally rich yet uncorrelated output signals

• JCRev is in the Synthesis Tool Kit (STK)
  • JCRev.cpp
  • JCRev.h.
• Four Schroeder “diffusion allpasses” in series

• Eight parallel Schroeder-Moorer lowpass-feedback-comb-filters:

\[ \text{LBCF}_{N}^{f,d} \triangleq \frac{1}{1 - f \frac{1-d}{1-d} z^{-1}} z^{-N} \]

• Second stereo channel: increase all 12 delay-line lengths by “stereo spread” (default = 23 samples)

• Used extensively in the free-software world
Freeverb Parameters

• $d$ ("damping") default:
  
  \[
  \text{damp} = \text{initialdamp} \ast \text{scaledamp} = 0.5 \cdot 0.4 = 0.2
  \]

• $f$ ("room size") default:
  
  \[
  \text{roomsize} = \text{initialroom} \ast \text{scaleroom} + \text{offsetroom} = 0.5 \cdot 0.28 + 0.7 = 0.84
  \]

• Feedback lowpass \((1 - d)/(1 - dz^{-1})\) causes reverberation time $t_{60}(\omega)$ to decrease with frequency $\omega$, which is natural

• $f$ mainly determines reverberation time at low-frequencies (where feedback lowpass has negligible effect)

• At very high frequencies, $t_{60}(\omega)$ is dominated by the diffusion allpass filters
T60 in Freeverb

- “Room size” $f$ sets low-frequency $t_{60}$
- “damping” $d$ controls how rapidly $t_{60}$ shortens as frequency increases
- Diffusion allpasses set lower bound on $t_{60}$

Interpreting “Room Size” Parameter

- Low-frequency reflection-coefficient for two plane-wave wall bounces
- Could be called liveness or reflectivity
- Changing roomsize normally requires changing delay-line lengths
Freeverb Allpass Approximation

Schroeder Diffusion Allpass

\[
\text{AP}_N^g \triangleq \frac{-g + z^{-N}}{1 - g z^{-N}}
\]

Freeverb implements

\[
\text{AP}_N^g \approx \frac{-1 + (1 + g) z^{-N}}{1 - g z^{-N}}
\]

- Each Freeverb “allpass” is more precisely a feedback comb-filter \(\text{FBCF}_N^g\) in series with a feedforward comb-filter \(\text{FFCF}_N^{-1,1+g}\), where

\[
\text{FBCF}_N^g \triangleq \frac{1}{1 - g z^{-N}}
\]

\[
\text{FFCF}_N^{-1,1+g} \triangleq -1 + (1 + g) z^{-N}.
\]

- A true allpass is obtained at \(g = (\sqrt{5} - 1)/2 \approx 0.618\) (reciprocal of “golden ratio”)

- Freeverb default is \(g = 0.5\)
FDN Late Reverberation

Jot (1991) FDN Reverberator for $N = 3$

- Generalized state-space model (unit delays replaced by arbitrary delays)
- Note direct path weighted by $d$
- The “tonal correction” filter $E(z)$ equalizes mode energy independent of reverberation time (perceptual orthogonalization)
- Gerzon 1971: “orthogonal matrix feedback reverb” cross-coupled feedback comb filters (see below)
Choice of Orthogonal Feedback Matrix $Q$

Late reverberation should resemble exponentially decaying noise. This suggests the following two-step procedure for reverberator design:

1. Set $t_{60} = \infty$ and make a good white-noise generator
2. Establish desired reverberation times in each frequency band by introducing losses

The white-noise generator is the lossless prototype reverberator.
Hadamard Feedback Matrix

A second-order Hadamard matrix:

\[
H_2 \triangleq \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix},
\]

Higher order Hadamard matrices defined by recursive embedding:

\[
H_4 \triangleq \frac{1}{\sqrt{2}} \begin{bmatrix} H_2 & H_2 \\ -H_2 & H_2 \end{bmatrix}.
\]

- Since \( H_3 \) does not exist, the FDN example figure above can be redrawn for \( N = 4 \), say, (instead of \( N = 3 \)), so that we can set \( Q = H_4 \).

- The Hadamard conjecture posits the existence of Hadamard matrices \( H_N \) of order \( N = 4k \) for all positive integers \( k \).

- “As of 2008, there are 13 multiples of 4 less than or equal to 2000 for which no Hadamard matrix of that order is known. They are: 668, 716, 892, 1004, 1132, 1244, 1388, 1436, 1676, 1772, 1916, 1948, 1964.”

Choice of Delay Lengths $M_i$

- Delay line lengths $M_i$ are typically *mutually prime* (Schroeder)
- For *sufficiently high mode density*, $\sum_i M_i$ must be sufficiently large.
  - No “ringing tones” in the late impulse response
  - No “flutter”
Mode Density Requirement

FDN order = sum of delay lengths:

\[ M \triangleq \sum_{i=1}^{N} M_i \]  
(FDN order)

- Order = number of poles
- All \( M \) poles are on the unit circle in the lossless prototype
- If uniformly distributed, mode density =

\[ \frac{M}{f_s} = MT \]  
modes per Hz

- Schroeder suggests 0.15 modes per Hz (when \( t_{60} = 1 \) second)
- Generalizing:

\[ M \geq 0.15 t_{60} f_s \]

- Example: For \( f_s = 50 \) kHz and \( t_{60} = 1 \) second, \( M \geq 7500 \)

- Note that \( M = t_{60} f_s \) is the length of the FIR filter giving a perceptually exact implementation. Thus, recursive filtering is about 7 times more efficient by this rule of thumb.
Choice of Loss Gains \( g_i \)

Jot (1991) FDN Reverberator for \( N = 3 \)

- To set the reverberation time \( t_{60} \), we need to move the poles of the lossless prototype slightly \textit{inside} the unit circle.

- The scaling coefficients \( g_i \) can accomplish this for \( 0 < g_i < 1 \).

- Since high-frequencies decay faster in propagation through air, we want to move the high-frequency poles farther in than low-frequency poles.

- Therefore, we need to generalize \( g_i \) above to \( G_i(z) \), with \( |G_i(e^{j\omega T})| \leq 1 \) imposed to ensure stability.
Damping Filter Design

The damping filter \( G_i(z) \) associated with the delay line of length \( M_i \) in the FDN can be written in principle as

\[
G_i(z) = G_T^{M_i}(z)L_i(z)
\]

where \( G_T(z) \) is the lowpass filter corresponding to one sample of wave propagation through air, and \( L_i(z) \) is a lowpass corresponding to absorbing/scattering boundary reflections along the (hypothetical) \( i \)th propagation path.

Define

\[
t_{60}(\omega) = \text{desired reverberation time at frequency } \omega
\]

\[
p_k = e^{j\omega_k T} = k\text{th pole of the lossless prototype}
\]

We can introduce frequency-independent damping with the (conformal map) substitution

\[
z^{-1} \leftarrow g z^{-1}
\]

- This \( z \)-plane mapping pulls all poles in the \( z \) plane from the unit circle to the circle of radius \( g \).
- Pole \( p_k = e^{j\omega_k T} \) moves to \( \tilde{p}_k = g e^{j\omega_k T} \).
Example

• Start with a pole at dc (digital integrator):

\[ H(z) = \frac{1}{1 - z^{-1}} \leftrightarrow [1, 1, 1, \ldots] \]

• Move it from radius 1 to radius 0.9 using \( z^{-1} \leftarrow 0.9z^{-1} \):

\[ H(z) = \frac{1}{1 - 0.9z^{-1}} \leftrightarrow [1, 0.9, 0.81, \ldots] \]
Frequency-Dependent Damping

For frequency-dependent damping, consider the mapping

\[ z^{-1} \leftarrow G(z) z^{-1} \]

where \( G(z) \) is a lowpass filter satisfying \( |G(e^{j\omega T})| \leq 1 \), \( \forall \omega \)

- Neglecting phase in the loss filter \( G(z) \), the substitution \( z^{-1} \leftarrow G(z) z^{-1} \) only affects the pole radius, not angle
- \( G(z) = \) per-sample filter in the propagation medium
- Schroeder (1961):
  
  The reverberation times of the individual modes must be equal or nearly equal so that different frequency components of the sound decay with equal rates ⇒
  
  - All pole radii in the reverberator should vary smoothly with frequency
  - Otherwise, late decay will be dominated by largest pole(s)
Lossy Mapping

Let’s in more detail look at the $z$-plane mapping

$$z^{-1} \leftarrow G(z) \, z^{-1}$$

- Pole $k$ contributes the term

$$H_k(z) = \frac{r_k}{1 - p_k z^{-1}} = r_k \cdot (1 + p_k z^{-1} + p_k^2 z^{-2} + \cdots)$$

to the partial fraction expansion of the transfer function

- This term maps to

$$\tilde{H}_k(z) = \frac{r_k}{1 - p_k[G(z) z^{-1}]} = r_k \cdot (1 + [G(z) p_k] z^{-1} + [G(z) p_k]^2 z^{-2} + \cdots)$$

- Thus, pole $k$ moves from $z = p_k = e^{j\omega_k T}$ to

$$\tilde{p}_k = R_k \, e^{j\omega_k T}$$

where

$$R_k = G \left( R_k \, e^{j\omega_k T} \right) \approx G \left( e^{j\omega_k T} \right)$$

which is a good approximation here since $R_k$ is nearly 1 for reverberators.
Example

• Start with a pole at dc (digital integrator):

\[ H(\hat{z}) = \frac{1}{1 - \hat{z}^{-1}} \leftrightarrow [1, 1, 1, \ldots] \]

• Move it from radius 1 to radius 0.9 using \( \hat{z}^{-1} \leftarrow 0.9 \hat{z}^{-1} \):

\[ H(z) = \frac{1}{1 - 0.9 z^{-1}} \leftrightarrow [1, 0.9, 0.81, \ldots] \]

• Now progress from radius 0.9 to 0.8 using

\[ \hat{z}^{-1} \leftarrow 0.9 \frac{1 + \alpha \hat{z}^{-1}}{1 + \alpha} z^{-1} \]

with \( 0.8 = (1 - \alpha)/(1 + \alpha) \)

\[ \Rightarrow \alpha = (1 - 0.8)/(1 + 0.8) = 1/0.9: \]

\[ H(z) = \frac{1}{1 - 0.9 \frac{1 + \alpha z^{-1}}{1 + \alpha} z^{-1}} = \frac{1}{1 - 0.9 \frac{0.9 + z^{-1}}{0.9 + 1} z^{-1}} \]

\[ = \frac{1}{1 - \frac{0.81}{1.9} z^{-1} + \frac{1}{1.9} z^{-2}} \]
**Desired Pole Radius**

Pole radius $R_k$ and $t_{60}$ are related by

$$R_k \frac{t_{60}(\omega_k)}{T} = 0.001$$

The ideal loss filter $G(z)$ therefore satisfies

$$|G(\omega)| \frac{t_{60}(\omega)}{T} = 0.001$$

The desired delay-line filters are therefore

$$G_i(z) = G^{M_i}(z)$$

$$\Rightarrow \quad |G_i(e^{j\omega T})| \frac{t_{60}(\omega)}{M_i T} = 0.001.$$ 

or

$$20 \log_{10} |G_i(e^{j\omega T})| = -60 \frac{M_i T}{t_{60}(\omega)}.$$

Now use `invfreqz` or `stmcb`, etc., in Matlab to design low-order filters $G_i(z)$ for each delay line.
First-Order Delay-Filter Design

Jot used first-order loss filters for each delay line:

\[ G_i(z) = g_i \frac{1 - a_i}{1 - a_i z^{-1}} \]

- \( g_i \) gives desired reverberation time at dc
- \( a_i \) sets reverberation time at high frequencies

Design formulas:

\[
\begin{align*}
g_i &= 10^{-3 M_i T / t_{60}(0)} \\
a_i &= \frac{\ln(10)}{4} \log_{10}(g_i) \left( 1 - \frac{1}{\alpha^2} \right)
\end{align*}
\]

where

\[
\alpha \triangleq \frac{t_{60}(\pi/T)}{t_{60}(0)}
\]
Let $h_k(n) = \text{impulse response of } k\text{th system pole}$. Then

$$E_k = \sum_{n=0}^{\infty} |h_k(n)|^2 = \text{total energy}$$

Thus, \textit{total energy is proportional to decay time}.

To compensate, Jot proposes a \textit{tonal correction filter} $E(z)$ for the late reverb (not the direct signal).

First-order case:

$$E(z) = \frac{1 - bz^{-1}}{1 - b}$$

where

$$b = \frac{1 - \alpha}{1 + \alpha}$$

and

$$\alpha \triangleq \frac{t_{60}(\pi/T)}{t_{60}(0)}$$

as before.
Zita-Rev1 Reverberator

- FDN+Schroeder reverberator
- Free open-source C++ for Linux by Fons Adriaensen
- Faust example zita_rev1.dsp

```
faust2firefox examples/zita_rev1.dsp
```

Feedback Delay Network + Schroeder Allpass Comb Filters:

- Allpass coefficients $\pm 0.6$
- Inspect Faust block diagram for delay-line lengths, etc.
Zita-Rev1 Damping Filters

FDN reverberators employ a damping filter for each delay line.

Zita-Rev1 three-band damping filter:

\[ H_d(z) = H_l(z)H_h(z) \]

where

\[ H_l(z) = g_m + (g_0 - g_m) \frac{1 - p_l}{2} \left( \frac{1 + z^{-1}}{1 - p_l z^{-1}} \right) = \text{low-shelf} \]

\[ H_h(z) = \frac{1 - p_h}{1 - p_h z^{-1}} = \text{low-pass} \]

\[ g_0 = \text{Desired gain at dc} \]

\[ g_m = \text{Desired gain across “middle frequencies”} \]

\[ p_l = \text{Low-shelf pole controlling low-to-mid crossover:} \]
\[ \Delta = \frac{1 - \pi f_1 T}{1 + \pi f_1 T} \]

\[ p_h = \text{Low-pass pole controlling high-frequency damping:} \]
\[ \text{Gives half middle-band } t_{60} \text{ at start of “high” band} \]
High-Frequency-Damping Lowpass

High-Frequency Damping Lowpass:

\[ H_h(z) = \frac{1 - p_h}{1 - p_h z^{-1}} \]

For \( t_{60} \) at “HF Damping” frequency \( f_h \) to be half of middle-band \( t_{60} \) (gain \( g_m \)), we require

\[ \left| H_h(e^{j2\pi f_h T}) \right| = \left| \frac{1 - p_h}{1 - p_h e^{-j2\pi f_h T}} \right| = g_m \]

Squaring and normalizing yields a quadratic equation:

\[ p_h^2 + b p_h + 1 = 0 \]

Solving for \( p_h \) using the quadratic formula yields

\[ p_h = -\frac{b}{2} - \sqrt{\left(\frac{b}{2}\right)^2 - 1}, \]

where

\[ -\frac{b}{2} = \frac{1 - g_m^2 \cos(2\pi f_h T)}{1 - g_m^2} > 1, \]

Discard unstable solution \( -b/2 + \sqrt{(b/2)^2 - 1} > 1 \)

To ensure \( |g_m| < 1 \), GUI keeps middle-band \( t_{60} \) finite
Rectilinear Digital Waveguide Mesh
A mesh of such waveguides in 2D or 3D can simulate waves traveling in any direction in the space.

Analogy: tennis racket = rectilinear mesh of strings = pseudo-membrane

Wavefronts are explicitly simulated in all directions

True diffuse field in late reverb

Spatialized reflections are “free”

Echo density grows naturally with time

Mode density grows naturally with frequency

Low-frequency modes very accurately simulated

High-frequency modes mistuned due to dispersion (can be corrected) (often not heard)

Multiply free almost everywhere

Coarse mesh captures most perceptual details
Reverb Resources on the Web

- Harmony Central article\(^2\) (with sound examples)
- William Gardner’s MIT Master’s thesis\(^3\)

\(^2\)http://www.harmony-central.com/Effects/Articles/Reverb/
\(^3\)http://www.harmony-central.com/Computer/Programming/virtual-acoustic-room.ps.gz