Simulating Reverberation

Spring 2011

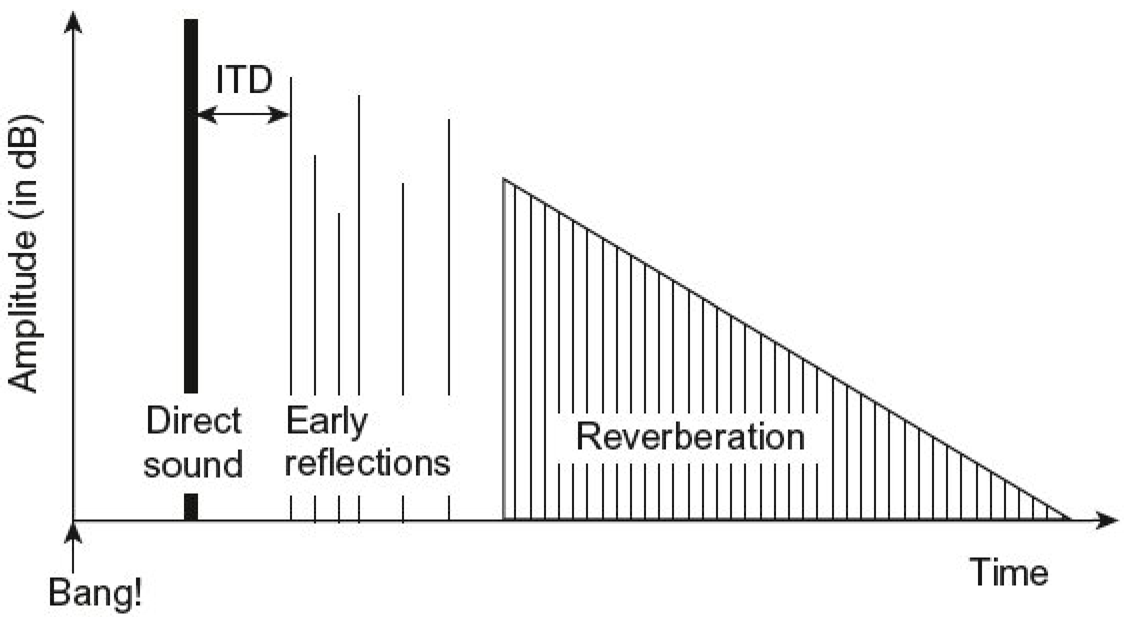
Some materials referenced from class materials by:
Prof. Clifford T. Mullis of the University of Colorado

Reverberation

- Reverberation is a natural effect due to sound reflections from surfaces in the physical space.
- Reverberation is achieved when the density of the reflections is high and increases over time while the magnitude of the reflections will typically present a logarithmic and “slow” decay over time.
- Each space presents different reverberation characteristics that will depend on its architectural characteristics as well as the properties of the materials.
- Different spaces can be modeled by simulating their reverberation characteristics. For example, a large opera hall vs a small voiceover studio.
- Every space has some amount of reverberation except for anechoic chambers which are typically artificially created.

Reverberation

- There are 3 main characteristics that describe the reverberation response:
  - The direct signal
  - Early reflections
  - Late reflections
- These can be reviewed by analizing the impulse response as follows:

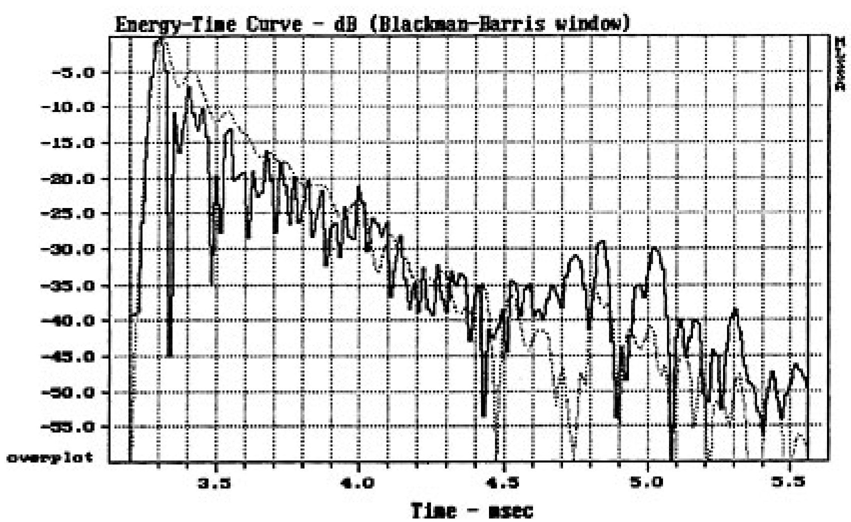


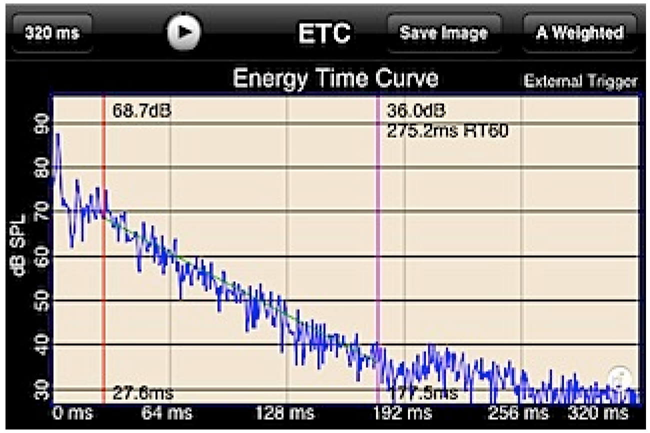
Source: EE Times, Acoustics and Psychoacoustics Applied - Part 1: Listening room design, David Howard and Jamie Angus

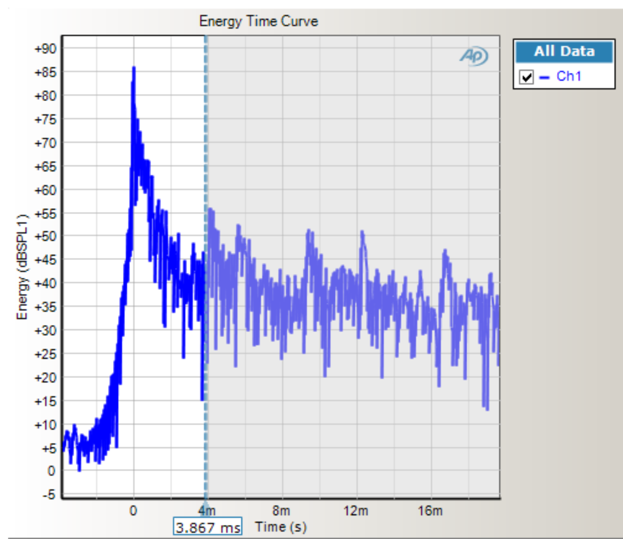
Reverberation

- The energy time curve (ETC) is a direct plot (in db) of how energy decays over time in a space.

- Here are a few real ETC graphs







Reverberation Parameters and Characteristics

- Reverberation Time (RT60): Describes the time required for the sound pressure to decrease 60dB from its initial value.

- Diffusion: Is the efficacy by which sound is spread evenly in an environment.

- Acoustic absorption: is the property of a material to absorb sound instead of reflecting it back to the space.

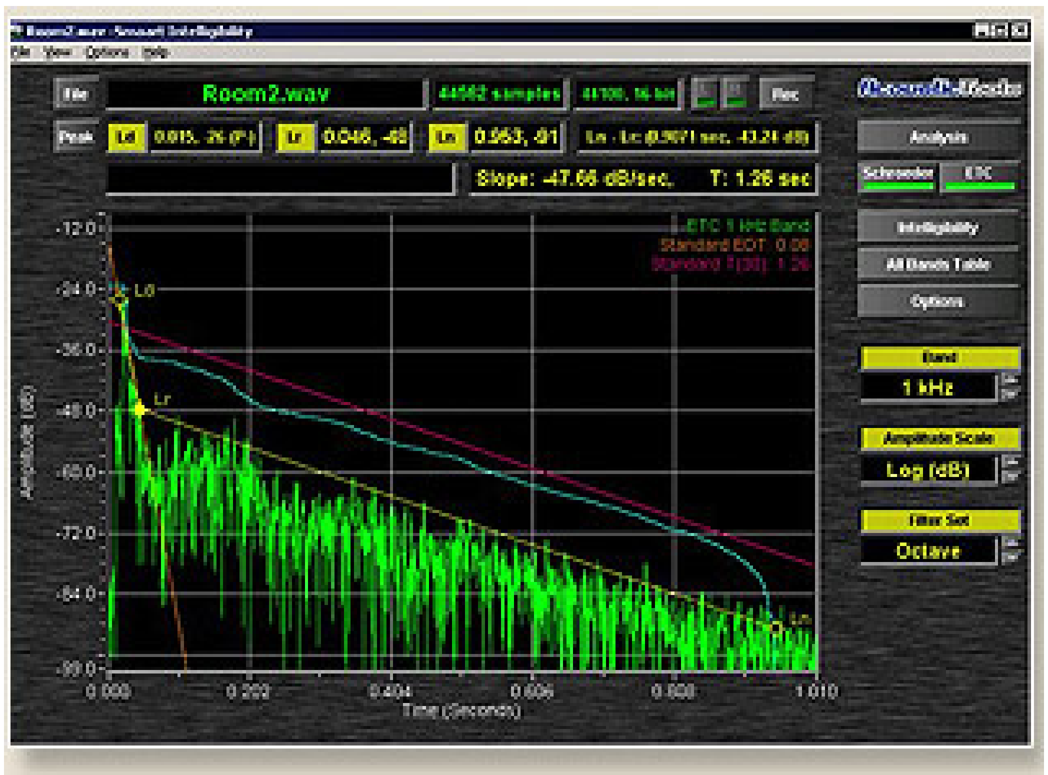
- Diffraction: Is the property of waves (of sound in this case) to get around an obstacle

Notice that all of the properties above are frequency dependent.

Reverberation

- The energy time curve (ETC) is a direct plot (in db) of how energy decays over time in a space.

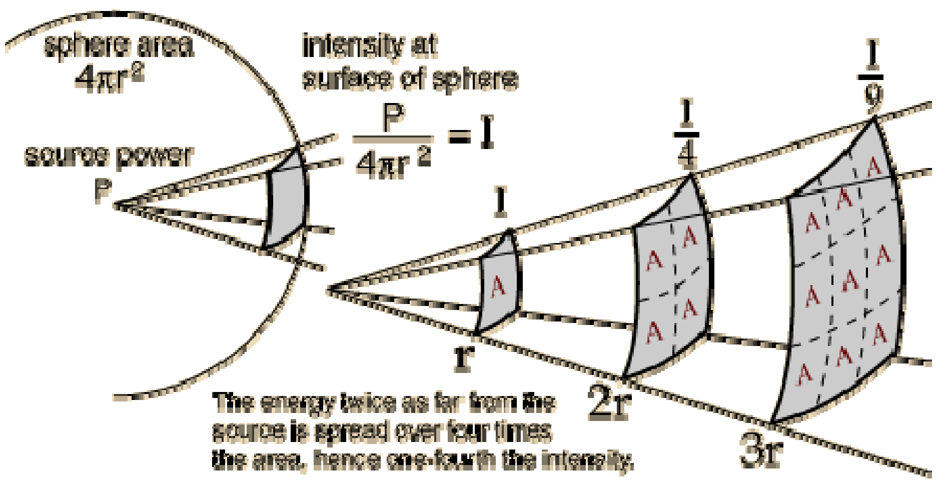
- Here is a real ETC graph

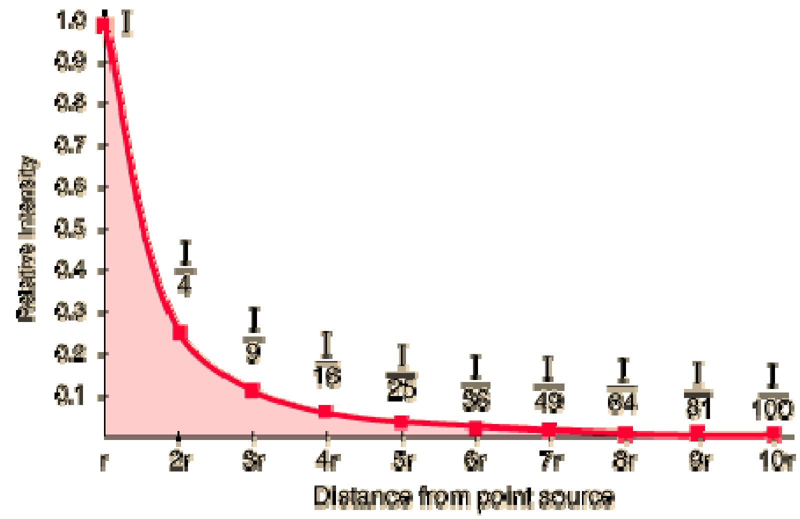


Source: EE Times, Acoustics and Psychoacoustics Applied - Part 1: Listening room design, David Howard and Jamie Angus

Side Note: Behavior of sound in space

- Inverse square law describes the decay in sound level as the signal gets further away from the source.





- Sound decays at 6db per doubling of distance. This is strictly only true when no reflections or reverberation are present (or when only the direct energy received from source to listener is considered) and only for a point source.
- Light from a point source also obeys the inverse square law.

Side Note: Absorption Coefficients

- Materials can be characterized by absorption coefficients that describe their efficiency to absorb (or reflect) sound. The absorption coefficient is the ratio of the absorbed energy over the incident energy.

- A value of 1.0 indicates that all sound is absorbed by the material while a value near 0 describes a highly reflective material.

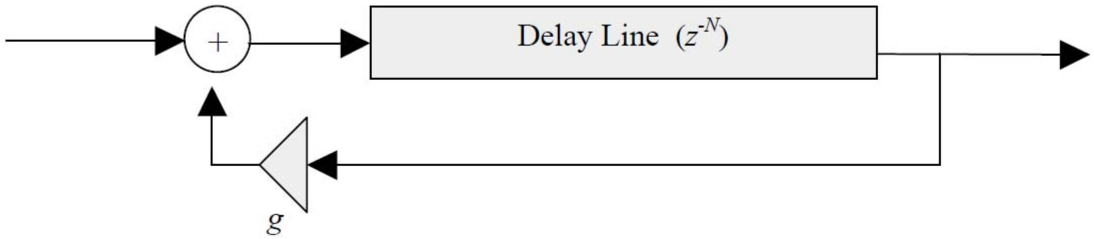
- Absorption coefficients are typically measured / specified for a set of frequencies or frequency bands

TABLE 15.1 Approximate typical absorption coefficients of various surfaces. Individual examples may vary considerably from these values.

Surface Treatment	Absorptivity at Frequency 125	250	500	1000	2000	4000
Acoustic tile, rigidly mounted	.2	.4	.7	.8	.6	.4
Acoustic tile, suspended in frames	.5	.7	.6	.7	.7	.5
Acoustical plaster	.1	.2	.5	.6	.7	.7
Ordinary plaster, on lath	.2	.15	.1	.05	.04	.05
Gypsum wallboard, $\frac{1}{2}''$ on studs	.3	.1	.05	.04	.07	.1
Plywood sheet, $\frac{1}{4}''$ on studs	.6	.3	.1	.1	.1	.1
Concrete block, unpainted	.4	.4	.3	.3	.4	.3
Concrete block, painted	.1	.05	.06	.07	.1	.1
Concrete, poured	.01	.01	.02	.02	.02	.03
Brick	.03	.03	.03	.04	.05	.07
Vinyl tile, on concrete	.02	.03	.03	.03	.03	.02
Heavy carpet, on concrete	.02	.06	.15	.4	.6	.6
Heavy carpet, on felt backing	.1	.3	.4	.5	.6	.7
Platform floor, wooden	.4	.3	.2	.2	.15	.1
Ordinary window glass	.3	.2	.2	.1	.07	.04
Heavy plate glass	.2	.06	.04	.03	.02	.02
Draperies, medium velour	.07	.3	.5	.7	.7	.6
Upholstered seating, unoccupied	.2	.4	.6	.7	.6	.6
Upholstered seating, occupied	.4	.6	.8	.9	.9	.9
Wood/metal seating, unoccupied	.02	.03	.03	.06	.06	.05
Wooden pews, occupied	.4	.4	.7	.7	.8	.7

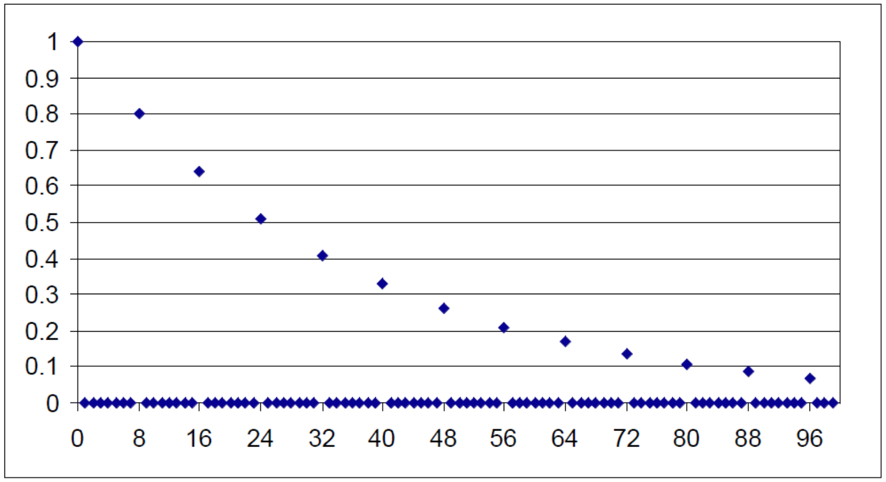
SOURCES: Backus (p. 172) and L. Doelle, *Environmental Acoustics* (McGraw-Hill, 1972), p. 227.

Simulating Reverberation: Schroeder Algorithm

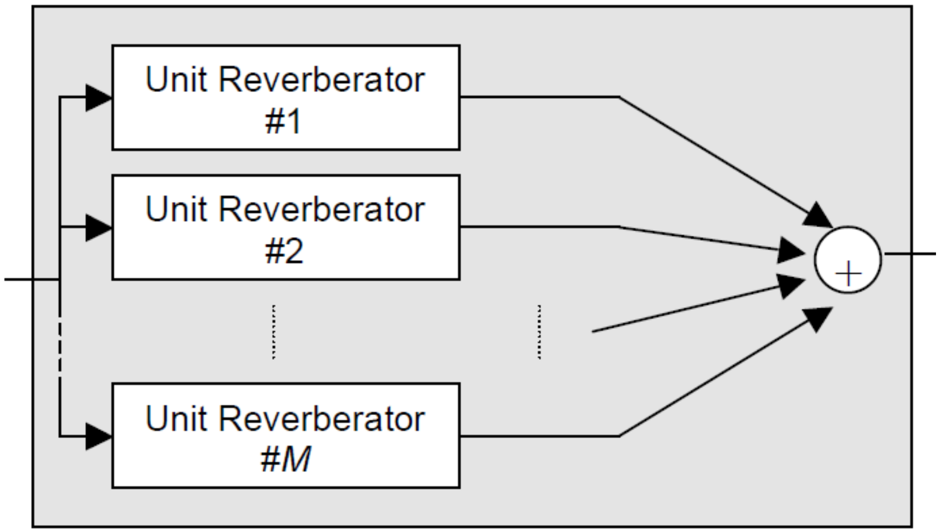


Recirculating Delay Line: IIR Filter

We start with a basic recirculating delay line to provide sound reflections. However, a single delay line will not provide too much of a regular predictable response as generated by a single reflecting point (one ray trace); this is not typical of real spaces and therefore will not sound natural.



Simulating Reverberation: Schroeder Algorithm



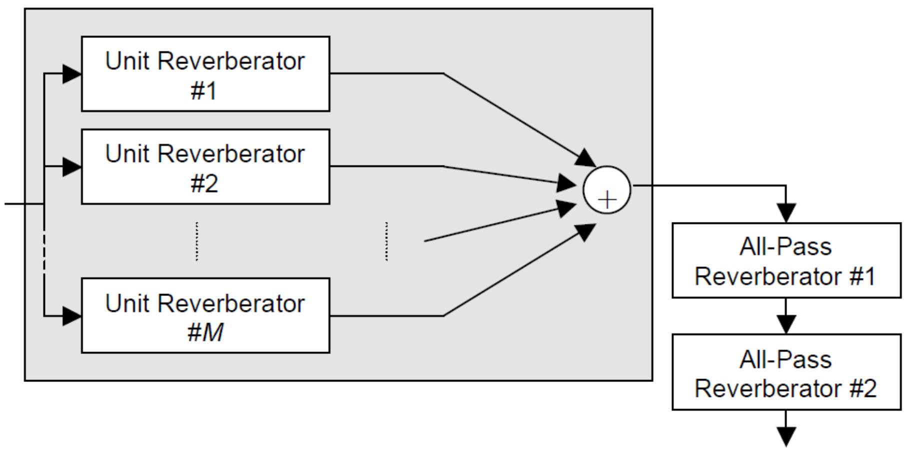
Multiple Unit Reverberators in Parallel:
Shroeders basic algorithm uses 4 units

We can increase the density and irregularity of the reflections by using multiple IIR filters in parallel.

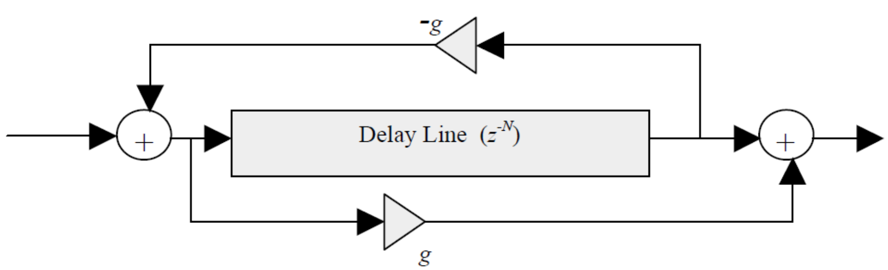
It is important that the delay times N do not represent multiples of the same period so that the individual impulse response of each reverberator does not coincide.

We are basically trying to recreate the relationship of multiple surfaces creating multiple unrelated reflections in a space.

Simulating Reverberation: Schroeder Algorithm



Multiple All-Pass Filters in Series:
Shroeders basic algorithm uses 2 units



Unit All Pass Filter

Finally, two all pass filters are added in series.

All pass filters provide a constant magnitude response in the frequency domain but their phase response varies rapidly therefore increasing the density of the reflections without affecting the frequency characteristics provided by the unit reverberators.

Simulating Reverberation

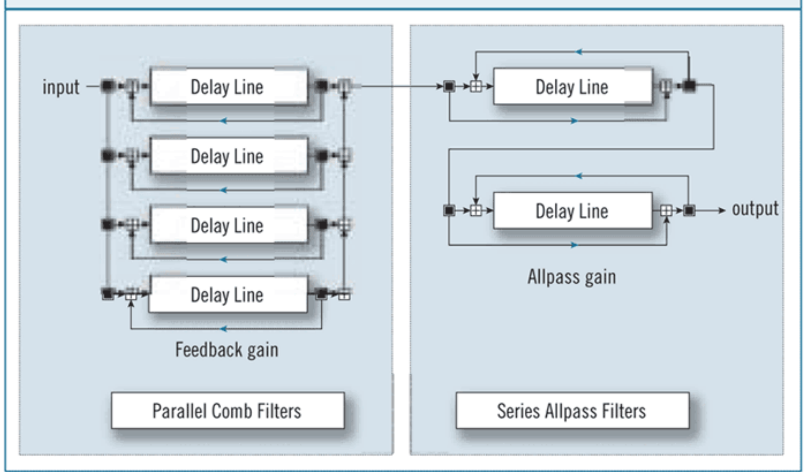
There are multiple methods/algorithms that can be used to simulate reverberation:

- Shroeder Algorithm: As described in the pages that follow, it involves a set of paralell comb filters and two all pass filters in series. Standard implementation for many years but not used that much anymore.

- Moorer Algorithm: Substitutes the basic feedback gains with recursive low-pass filters that simulate the effects of air absortion and provide a much more realistic effect.

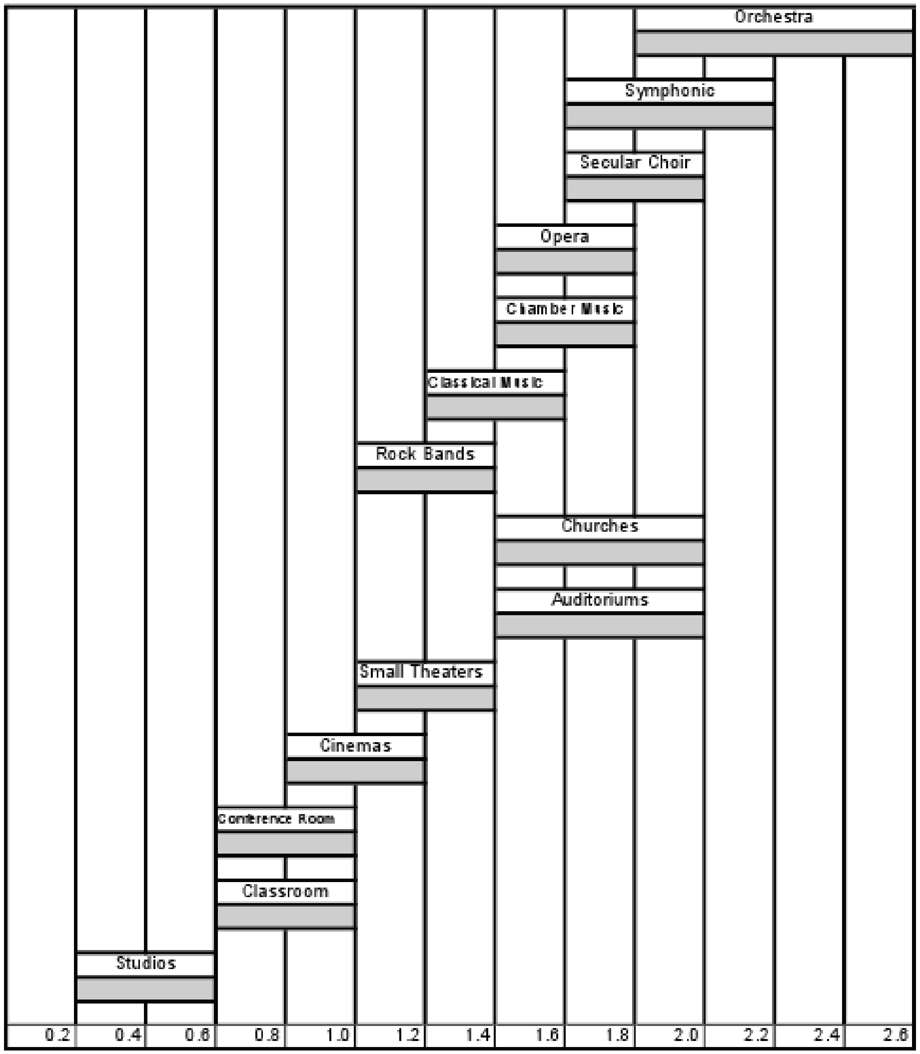
- Gardner, Dattoro and Jot Algorithms: Different implementations using combinations of variable low-pass filters on feedback loops and/or all pass filters on time varying delay lines.

Figure 1 Schroeder Reverb



Room Modeling

Typical reverberation times for different spaces:



Reverberation Time in seconds

Room modeling involves simulating the acoustical characteristics of a space. Since as we've discussed, reflections, reverberation and the spectral response of a space in general are fairly complex, accurately modeling a space requires a detailed description of the materials, architecture and positions of listeners/sources in it.

Some applications for room modeling:

- Effects used for sound recording and production
- Sound reinforcement models used for sound system design
- Acoustics analysis for architectural design

Lab #5 : Review

Macros vs Subroutines

When using the same algorithm multiple times in a process, it is useful to define a subroutine or macro instead of writing the same code several times (for example, for the unit reverberators in parallel only the parameters vary but the algorithm is the same).

- Subroutines are calls made during run-time that require the program pointer to jump to a different place in memory to execute the code. So basically the code is only stored once in memory. However, there is an overhead cost in processing and memory for the DSP to process and handle the jump.

- Macros allow the code to be written only once but the compiler will actually recreate the code in program memory wherever it is called. So there are no jumps or branches involved (and hence no overhead). The downside (there always is a downside!) is that additional program memory is required every time the macro is called.

Granted that processing and memory requirements (data and/or program) are not limited, there is no functional difference to implementing the code either way. However, in real world applications the consequences of either method must be analyzed to determine the best approach.

Macro Definition

Freescale DSP Assembler Reference Manual Chapter 5

Macro definition header: <label> MACRO [<dummy argument list>] [<comment>]

```
N_R_MUL   MACRO     NMUL,AVEC,BVEC,RESULT                header
;
;This macro implements N real multiplies
;RESULT(I) = AVEC(I) * BVEC(I) I=1..NMUL
;where
;         NMUL      = number of multiplications
;         AVEC      = base address of array AVEC(I)
;         BVEC      = base address of array BVEC(I)
;         RESULT    = base address of array RESULT(I)
;
          MOVE      #AVEC,R0                             body
          MOVE      #BVEC,R4
          MOVE      #RESULT,R1
          MOVE      X:(R0)+,D4.S       Y:(R4)+,D7.S
          DO        #NMUL,_ENDLOOP
          FMPY.S    D4,D7,D0           X:(R0)+,D4.S       Y:(R4)+,D7.S
          MOVE      D0.S,X:(R1)+
_ENDLOOP
          ENDM                                           terminator
```

Macro call: [<label>] <macro name> [<arguments>] [<comment>]

For example: N_R_MUL CNT+1,VEC1,VEC2,OUT

Macro Definition: Dummy Arguments and Operators

The DSP allows text substitutions of the arguments during the macro expansion and recognizes certain text operators for added flexibility:

- Concatenation Operator - \
- Return Value Operator - ?
- Return Hex Value operator - %
- String Operator - "
- Macro Local Label Override Operator - ^

For example:

Macro Definition:

```
SWAP_REG   MACRO REG1,REG2 ;swap REG1,REG2 using X0 as temp
           MOVE R\REG1,X0
           MOVE R\REG2,R\REG1
           MOVE X0,R\REG2
           ENDM
```

Macro Call: SWAP_REG 0,1

Macro Expansion:

```
           MOVE R0,X0
           MOVE R1,R0
           MOVE X0,R1
```

Freescale DSP Assembly: Expressions

Chapter 3 of the DSP Assembler Reference Manual describes the use of expressions. Additionally to basic logical, arithmetic and other operators, expressions also include some built in functions that are included in the assembler and allow data conversion, string comparison and transcendental math computations.

@POW(<expr1>,<expr2>**)**

Returns <expr1> raised to the power <expr2> as a floating point value. <expr1> and <expr2> must be separated by a comma. The memory space attribute of the result will be **N**one.

Example:

```
BUF     EQU     @CVI(@POW(2.0,3.0))          ; BUF = 8
```

@CVI(<expression>**)**

Converts the result of <expression> to an integer value. This function should be used with caution since the conversions can be inexact (e.g., floating point values are truncated). The memory space attribute of the result will be **N**one.

Example:

```
INT     SET     @CVI(-1.05)          ; INT = -1
```

IIR Filters

IIR vs FIR

- FIR filter equation:

$$y[n] = \sum_{k=0}^{M} b_k x[n-k] \quad \textbf{(A)}$$

Transfer Function:

$$H(z) = \sum_{k=0}^{M} h(k) z^{-k}$$

- IIR filter equation:

$$\sum_{i=0}^{M} a_i y[n-i] = \sum_{k=0}^{N} b_k x[n-k] \quad \textbf{(B)}$$

Transfer Function:

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}}$$

Theory References:

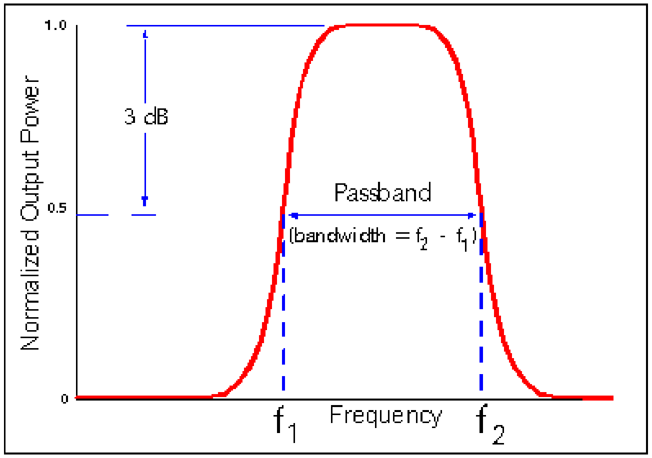
- Oppenheim, Schafer & Buck, Chapter 7
- http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-341-discrete-time-signal-processing-fall-2005/lecture-notes/lec09.pdf
- http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-341-discrete-time-signal-processing-fall-2005/lecture-notes/lec08.pdf

FIR Filters in comparison to IIR filters

	FIR Filters	IIR Filters
Resource demands	High, in both memory and MAC's per input sample	Relatively low, in both respects
Phase response	Excellent. FIR filters have linear phase when h(k) has symmetry, which is easily obtained.	The phase is nonlinear in the pass-band, although it can be approximately linear. The unit pulse response cannot achieve symmetry.
Stability	Not an issue. FIR filters are always stable.	The poles must be inside the unit circle.
Coefficient representation as a fraction in the 56K	Not an issue for LP, HP, BP, etc filters, since the numbers h(k) will have magnitudes less than one, and typically much less than one.	Stable second order direct form filters can have coefficients whose absolute value is greater than one. SOAP structures avoid this problem; all coefficients will have magnitude less than one.
Roundoff noise	Minimal, when a double precision accumulator is used. Then there is but one round per filter output sample.	Can be severe. There are several rounds per filter output sample and the round-off errors can remain because of the feedback loops inherent in IIR filters.
Coefficient sensitivity	Not an issue, at least for reasonably large word sizes.	Can be severe.
Overflow	Not an issue.	Can be severe. Second order sections in cascade must be separately scaled to prevent overflow. SOAP structures need be scaled only at the input.

Band Selective Filters

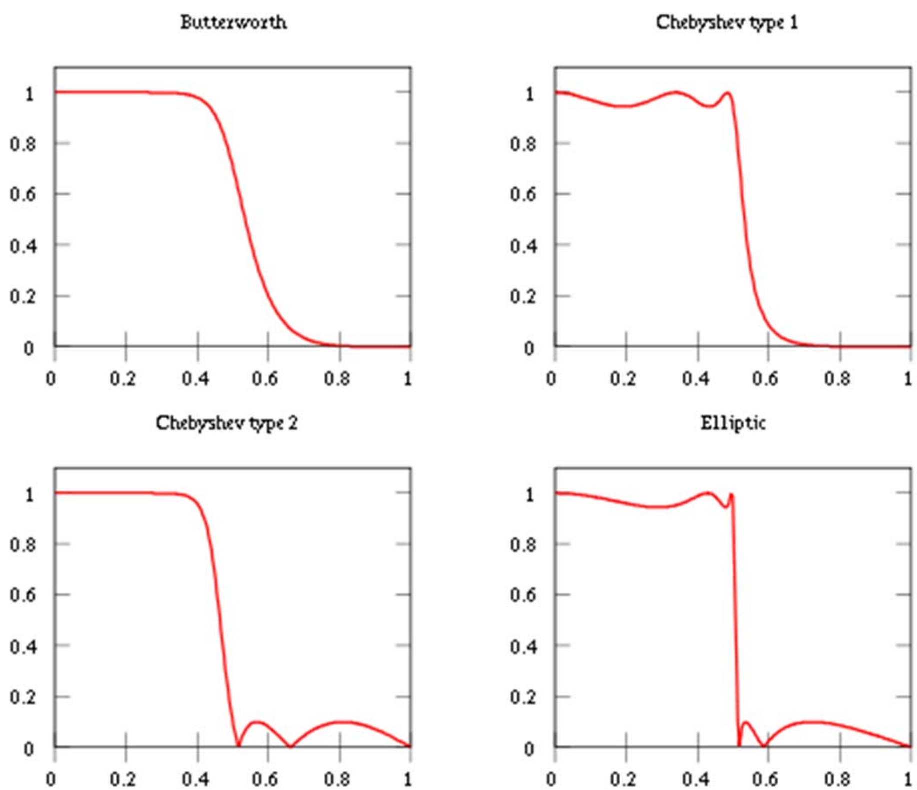
Until now most of the filters we have worked with are low-pass filters but there is often a need to provide a band-pass filter that will only let a certain range of frequencies through.

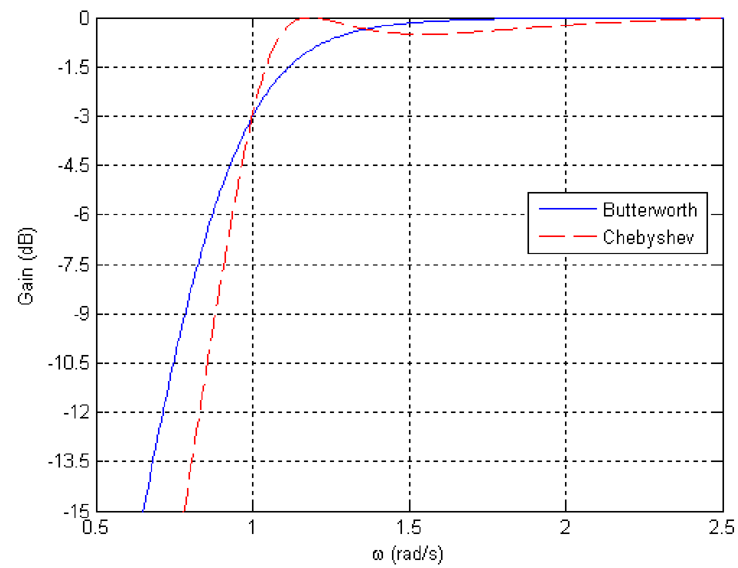


bandpass filter

A method for designing a band selective filter is to first design a prototype half-band low pass filter and then perform a frequency transformation filter. We define the filter's characteristics for the half-band low-pass filter and then perform a frequency shift to the desired frequency range.

Chebyshev and Butterworth Filters





Typically it is a tradeoff between ripple and steepness of the curve

Semester End Plan

- All final projects need to be submitted by Monday May 7
- Lectures during April will be on special topics and open for review and support at lab.
- Final project presentations on April 30.
- There are 6 slots available for presentations to class on Monday April 25th during regular class hours. Groups/students that present to the class will get 2 to 4 bonus points depending on quality of project and presentation.
  - First come first serve
  - Only 1 Image Processing Lab #5 presentation allowed
- Otherwise, I will make available the following time slots for demonstrating the final project:
  - Tuesday May 1st from 4:00-5:00pm
  - Wednesday May 2nd from 3:00-4:30pm
  - Monday May 7 from 4:00-5:00pm