MUS420 Lecture
Time Varying Delay Effects

Julius O. Smith III (jos@ccrma.stanford.edu) , Stefania Serafin,
Jonathan S. Abel, and David P. Berners

Center for Computer Research in Music and Acoustics (CCRMA)
Department of Music, Stanford University
Stanford, California 94305

June 27, 2020

Outline

- Variable Delay Lines
- Flanging
- Phasing
- Vibrato
- Doppler Effect and its simulation
- Chorus Effect
- Modeling the Leslie Effect

Variable Delay Line

```
static double A[N];
static double *rptr = A; // read ptr
static double *wptr = A; // write ptr

double setdelay(int M) {
    rptr = wptr - M;
    while (rptr < A) { rptr += N; }
}

double delayline(double x)
{
  double y;
  *wptr++ = x; // A[wptr++] = x;
  y = *rptr++; // y = A[rptr++];
  if ((wptr-A) >= N) { wptr -= N; }
  if ((rptr-A) >= N) { rptr -= N; }
  return y;
}
```

Note separate read- and write-pointers

Variable Delay Line Applications

Variable delay lines are used for

- Phasing
- Flanging
- Chorus
- Leslie
- Reverb

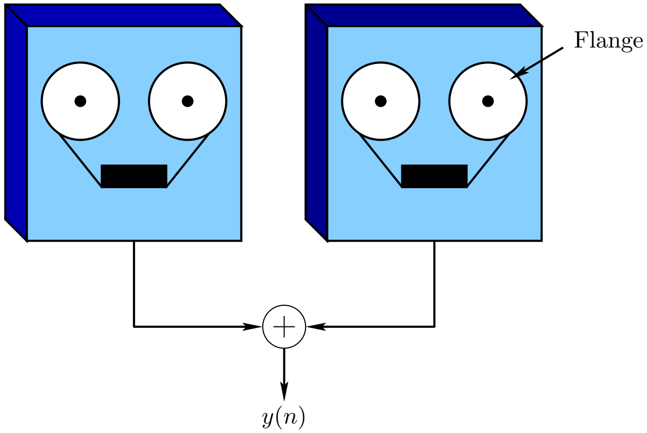
In *digital waveguide synthesis*, variable delay lines are used for

- Vibrating strings (guitars, violins, . . .)
- Woodwind bores
- Horns
- Tonal percussion (rods, membranes)

Flanging

Perhaps the simplest time-varying delay effect:

- Used since the 1960s
- Two identical tape machines mixed equally:

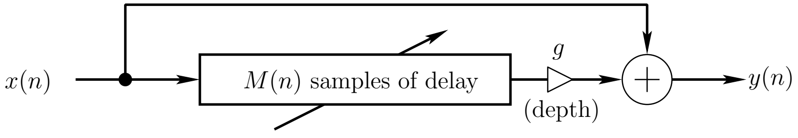


- Touch the *flange* of one of the supply reels, slowing it
- Touch the flange of the other supply reel to slow it
- Repeat, creating an oscillating *variable delay*
- Keep the maximum delay below 10 ms or so
- Sound Examples[1]

[1] http://en.wikipedia.org/wiki/Flanging

Flanging Model

Model = Feedforward comb filter with variable delay M:

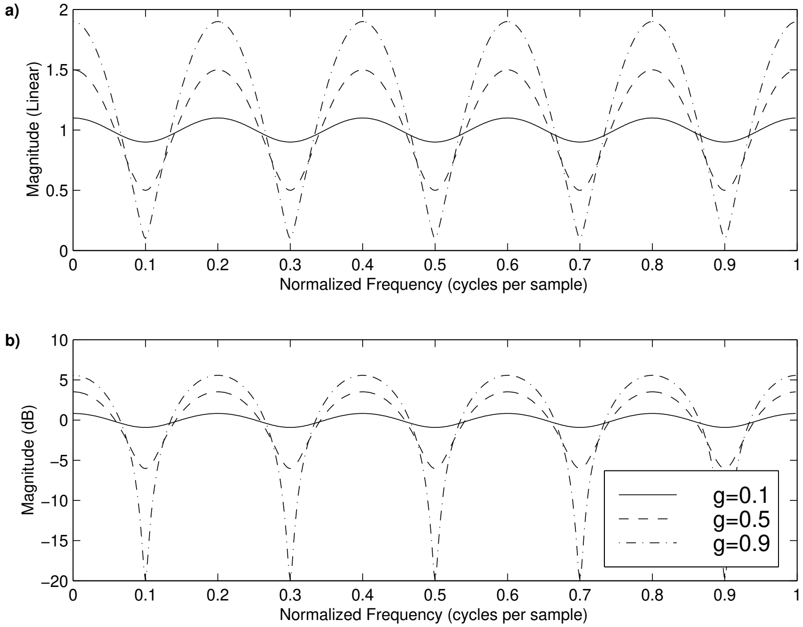


$$y(n) = x(n) + gx[n - M(n)]$$

- $g =$ "depth" of the flanging effect
- $M(n) =$ length of the delay-line at time n
- $M(n)$ typically triangular or sinusoidal
- Delay length is *modulated* by an "LFO"
- Smooth delay variation demands *interpolation*

Feed-Forward Comb-Filter Amplitude Response

Recall:



- Linear (top) and decibel (bottom) amplitude scales
- $H(z) = 1 + gz^{-M}$
  - $M = 5$
  - $g = 0.1, 0.5, 0.9$

- $G(\omega) \stackrel{\Delta}{=} \left|H(e^{j\omega T})\right| = \left|1 + ge^{-jM\omega T}\right| \to$ $2\cos(M\omega T/2)$ when $g = 1$
- In *flangers*, these nulls slowly move with time

For $g > 0$, there are M *peaks* in the frequency response, centered about frequencies

$$\omega_k^{(p)} = k\frac{2\pi}{M}, \quad k = 0, 1, 2, \ldots, M-1.$$

For $g = 1$ (maximum "flanging depth"), the peaks are maximally pronounced, with M *notches* occurring between them at frequencies $\omega_k^{(n)} = \omega_k^{(p)} + \pi/M$.

Notch spacing is inversely proportional to delay-line length

Summary of Flanging

A *flanger* provides *uniformly spaced notches* and/or *peaks* in the frequency response that *move* over time

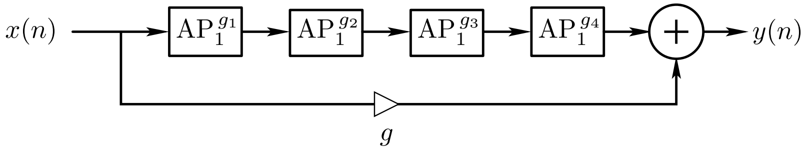
"Basilar membrane comb massage"

Drawbacks to Flanging:

- Uniform notches can *zero* a periodic signal
- Uniform peaks can impose "resonant pitch" (sounds like being inside a resonant tube)
- *Exponentially* spaced notches (uniformly spaced on a log frequency scale) would be more uniform perceptually

Phasing

Phasers typically use *nonuniformly spaced* notches made using a *series chain of allpass filters*:



See for example the Univibe[2]

First-Order Allpass Filter Section

Analog Transfer Function:

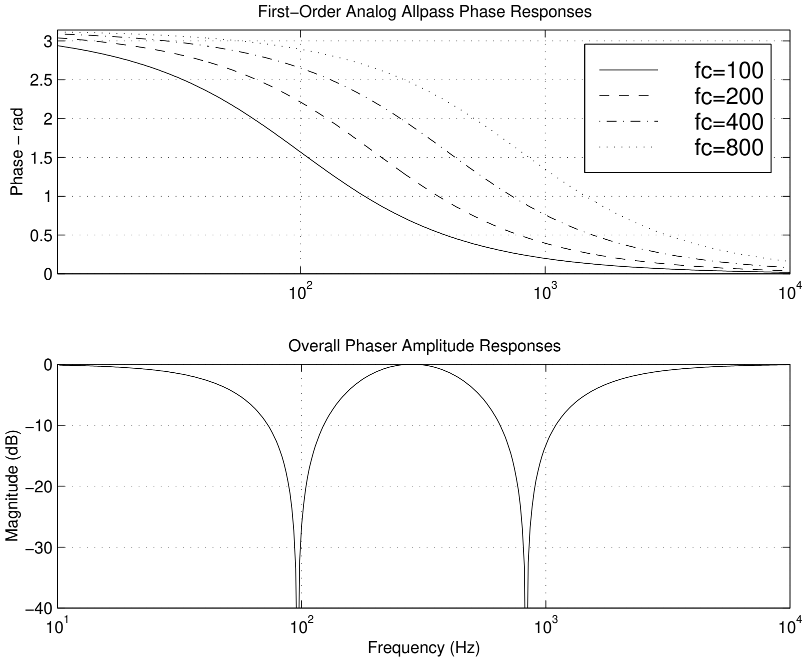
$$\mathsf{AP}_1^{\omega_b} \triangleq \frac{s-\omega_b}{s+\omega_b}.$$

Digital Transfer Function:

$$\mathsf{AP}_1^{g_i} \triangleq \frac{g_i+z^{-1}}{1+g_i z^{-1}}.$$

[2] http://www.geofex.com/Article_Folders/univibe/uvfrindx.htm

Four First-Order Analog Allpass Filters



(a) Phase reponses of first-order analog allpass sections with $\Theta(\omega T)=\pi/2$ frequencies at 100, 200, 400, and 800 Hz. (b) Corresponding phaser amplitude response.

In practical operation, the $\pi/2$ frequencies *change dynamically*, usually periodically at some rate.

Vibrato Simulation

Vibrato can be simulated approximately using a *modulated delay line*

A time-varying delay line can be interpreted as a *Doppler-shift* simulator due to motion of the listener relative to the source.

Doppler Effect

Doppler Visualization by Daniel Russel at Kettering University[3]

- The *Doppler effect* causes the pitch of a sound source to appear to rise or fall due to *motion* of the source and/or listener relative to each other.
- Recall the sound of a passing horn
- Pitch is higher when source moves toward you
- Pitch is lower when source moves away from you
- Doppler shift is significant in the "Leslie effect"

[3] http://paws.kettering.edu/~drussell/Demos/doppler/doppler.html

Doppler Shift Formula

The *Doppler shift* is given by

$$\omega_l = \omega_s \frac{1 + \frac{v_{ls}}{c}}{1 - \frac{v_{s,l}}{c}}$$

where

- $\omega_s =$ radian frequency emitted by the source at rest
- $\omega_l =$ frequency received by the listener
- $v_{ls} =$ *speed* of the listener
  relative to the propagation medium
  in the direction of the source
- $v_{s,l} =$ speed of the source
  relative to the propagation medium
  in the direction of the listener
- $c =$ sound speed

Note that all quantities in this formula are scalars (not vectors)

Vector Formulation

Let

- $\underline{v}_s(t) =$ sound-source *velocity* as a function of time t
- $\underline{v}_l(t) =$ velocity of the listener, if any
- $\underline{x}_s(t) =$ *position* of source
- $\underline{x}_l(t) =$ *position* of listener
- $\underline{x} \overset{\Delta}{=} (x_1, x_2, x_3)^T$ is 3D position

We have velocity related to position by

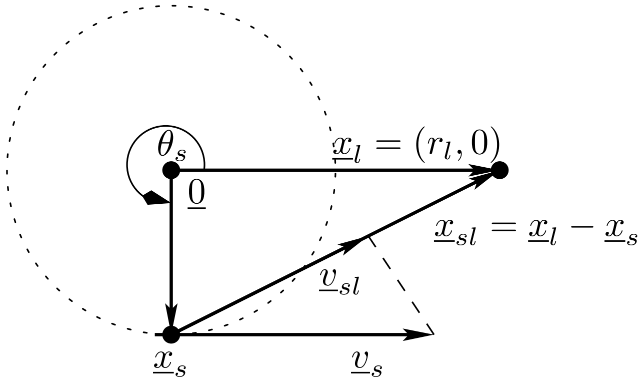
$$\underline{v}_s = \frac{d}{dt}\underline{x}_s(t) \qquad \underline{v}_l = \frac{d}{dt}\underline{x}_l(t).$$

Consider a Fourier component of the source at frequency ω_s. We wish to know how this frequency is shifted to ω_l at the listener due to the Doppler effect.

Velocity Projection

The Doppler effect depends only on velocity components along the line connecting the source and listener.

We may therefore *orthogonally project* the source and listener velocities onto the vector $\underline{x}_{sl} = \underline{x}_l - \underline{x}_s$ pointing from the source to the listener.



θ_s
$\underline{x}_l = (r_l, 0)$
$\underline{0}$
$\underline{x}_{sl} = \underline{x}_l - \underline{x}_s$
$\underline{v}_{sl}$
$\underline{x}_s$
$\underline{v}_s$

Relevant geometry for a rotating horn

Projected source velocity:

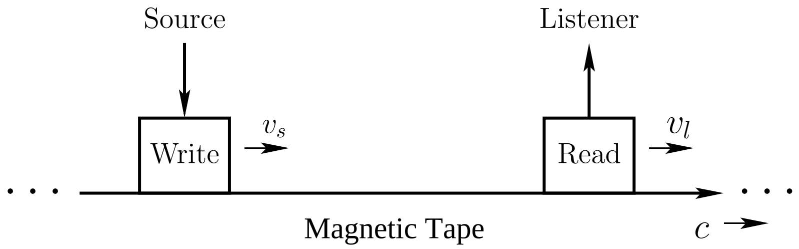
$$\underline{v}_{sl} = \mathcal{P}_{\underline{x}_{sl}}(\underline{v}_s) = \frac{\langle \underline{v}_s, \underline{x}_{sl} \rangle}{\| \underline{x}_{sl} \|^2}\underline{x}_{sl} = \frac{\langle \underline{v}_s, \underline{x}_l - \underline{x}_s \rangle}{\| \underline{x}_l - \underline{x}_s \|^2}(\underline{x}_l - \underline{x}_s)$$

$$\approx \frac{\langle \underline{v}_s, \underline{x}_l \rangle}{\| \underline{x}_l \|^2}\underline{x}_l = \mathcal{P}_{\underline{x}_l}(\underline{v}_s) \quad \text{in far field } (\| \underline{x}_l \| \gg \| \underline{x}_s \|)$$

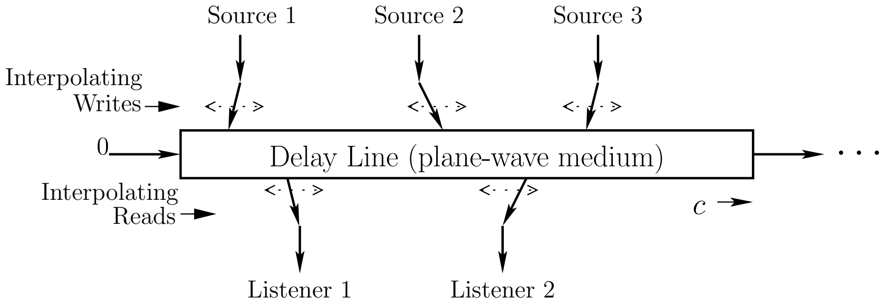
Doppler Simulation

Consider Doppler shift from a physical point of view:

- Air analogous to a *magnetic tape* moving from source to listener at speed c
- Source is analogous to the *write-head* of a tape recorder
- Listener corresponds to the *read-head*
- When either head moves, a Doppler shift is heard



Doppler Simulation via Delay Lines



The magnetic tape is now the delay line:

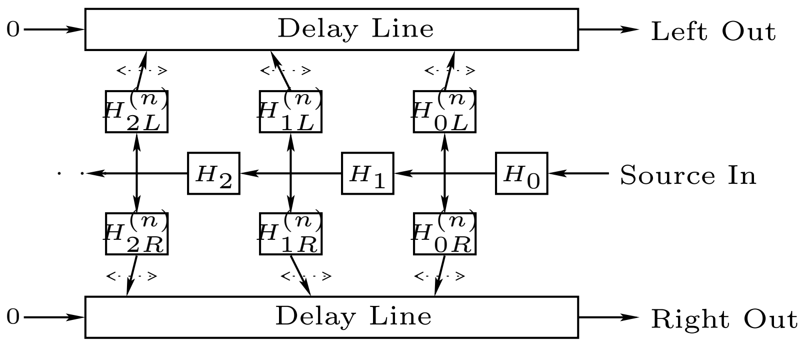
- Tape read-head → delay-line read-pointer
- Tape write-head → delay-line write-pointer
- Listener motion away from the source at speed v_{ls} corresponds to advancing the read-pointer by $1 + v_{ls}/c$ (thereby requiring interpolated reads)
- Changing the *write-pointer* increment from 1 to $1 + v_{s,l}/c$ corresponds *source motion toward the listener* at speed $v_{s,l}$ (thereby requiring *interpolating writes* into the delay memory)
- Recall that interpolating writes are formally the graph-theoretic *transpose* of interpolating reads (ordinary "interpolation")

Multiple Read/Write Pointers

Misc. Refinements:

- Using *multiple read pointers*, multiple moving listeners are simulated
- Each read-pointer signal can be *filtered* to simulate propagation losses and radiation characteristics of the source in the direction of the listener
- Read-pointers can move independently to simulate the different Doppler shifts associated with different listener motions and relative source directions.
- Using *multiple write pointers*, multiple moving sources are simulated
- Watch out for aliasing due to upward Doppler shift

Stereo Multiple-Moving-Source Simulation



- Input and output signals are on the right
- Only one input source is shown
- Additional input sources are handled identically, summing into the same delay lines in the same way.
- Source signal first passes through filter $H_0(z)$, which provides *time-invariant* filtering common to all propagation paths.
- Left- and right-channel filters $H_{0L}^{(n)}(z)$ and $H_{0R}^{(n)}(z)$ are typically low-order, linear, *time-varying* filters implementing the time-varying characteristics of the shortest (time-varying) propagation path from the source to each listener.

- These filter outputs *sum* into the delay lines at arbitrary (time-varying) locations using interpolating writes (de-interpolation)
- The zero signals entering each delay line on the left can be omitted if the left-most filter overwrites delay memory instead of summing into it.
- The outputs of $H_{0L}^{(n)}(z)$ and $H_{0R}^{(n)}(z)$ correspond to the "direct signal" from the moving source, when a direct signal exists
- These filters may incorporate modulation of losses due to the changing propagation distance from the moving source to each listener, and they may include dynamic equalization corresponding to the changing radiation strength in different directions from the moving (and possibly turning) source toward each listener.
- The next trio of filters, $H_1(z)$, $H_{1L}^{(n)}(z)$, and $H_{1R}^{(n)}(z)$, correspond to the next-to-shortest acoustic propagation path, typically the "first reflection," such as from a wall close to the source.
- Since a reflection path is longer than the direct path, and since a reflection itself can attenuate (or scatter) an incident sound ray, there is generally more filtering required relative to the direct signal. This additional

filtering can be decomposed into its fixed component $H_1(z)$ and time-varying components $H_{1L}^{(n)}(z)$ and $H_{1R}^{(n)}(z)$.

- Note that acceptable results may be obtained without implementing all of the filters indicated. Furthermore, it can be convenient to incorporate $H_i(z)$ into $H_{iL}^{(n)}(z)$ and $H_{iR}^{(n)}(z)$ when doing so does not increase their orders significantly.
- The source-filters $H_{iL}^{(n)}(z)$ and $H_{iR}^{(n)}(z)$ may include HRTF filtering in order to impart illusory angles of arrival in 3D space.

Chorus Effect

- The *chorus effect* (or “choralizer”) makes one sound source (such as a voice) sound like *many* singing (or playing) in *unison*
- Since performance in unison is never exact, chorus effects simulate this by making independently modified copies of the input signal
- Modifications include

(1) delay
(2) frequency shift
(3) amplitude modulation

- Typical implementations use several *time-varying delay lines* for (1) and (2)
- Multipath filtering and reverb provide (3) indirectly
- Before digital delay lines, analog LC ladder networks were used (*e.g.*, in the Hammond organ)
- An efficient chorus-effect implementation may be based on *multiple interpolating taps* working on a single delay line
- Each tap should be individually spatialized

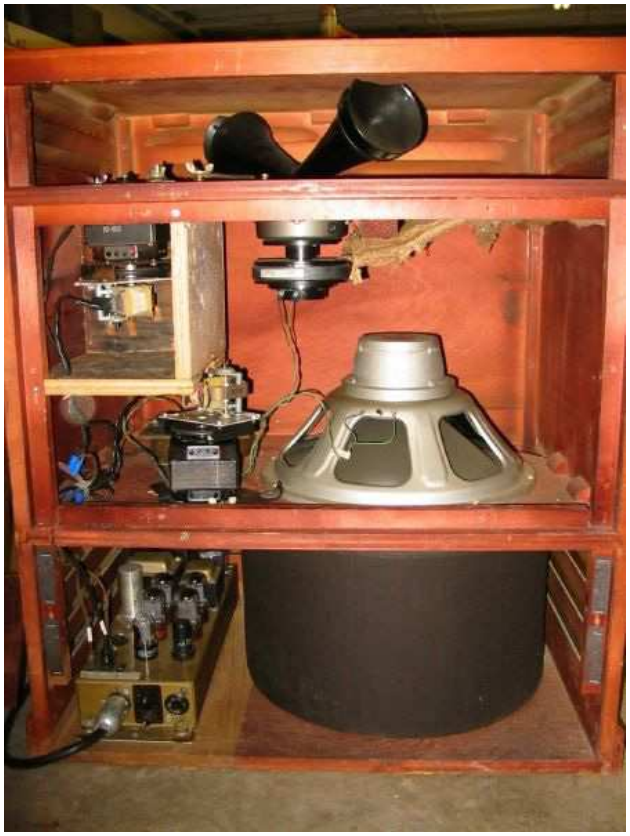
The Leslie

Leslie cabinet

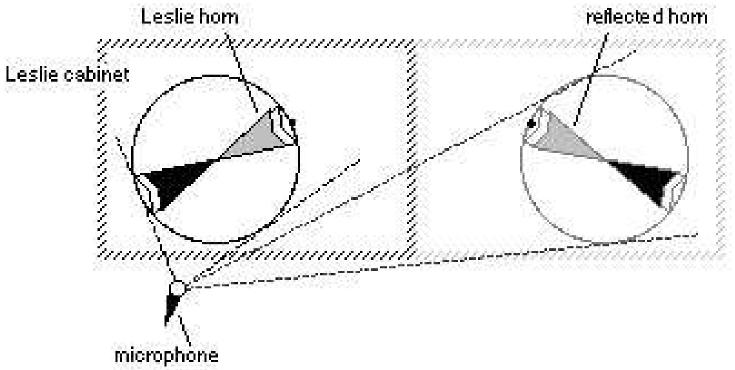
- The *Leslie*, invented by Don Leslie, is typically used to *choralize* electronic organs and other instruments
- See the Leslie Wikipedia page[4] for an introduction. (Stereo sound example just under the first picture best heard in headphones.)
- Primarily a *rotating horn* and *rotating speaker port*
- The listener hears *multiple cabinet reflections* at *different Doppler shifts*, giving a kind of *chorus effect*

[4] http://en.wikipedia.org/wiki/Leslie_speaker

Inside the Leslie



Leslie Operation



Schematic operation of the Leslie horn.

- Rotating horn (and dummy horn)
- Multiple “image sources” due to cabinet reflections
- Reflections heard with *different Doppler shifts*

Rotating Horn Simulation

Horn source-position model:

$$\underline{x}_s(t) = \left[\begin{array}{c} r_s\cos(\omega_m t) \\ r_s\sin(\omega_m t) \end{array}\right]$$

where

$$\begin{aligned} r_s &= \text{circular radius} \\ \omega_m &= \text{angular velocity} \end{aligned}$$

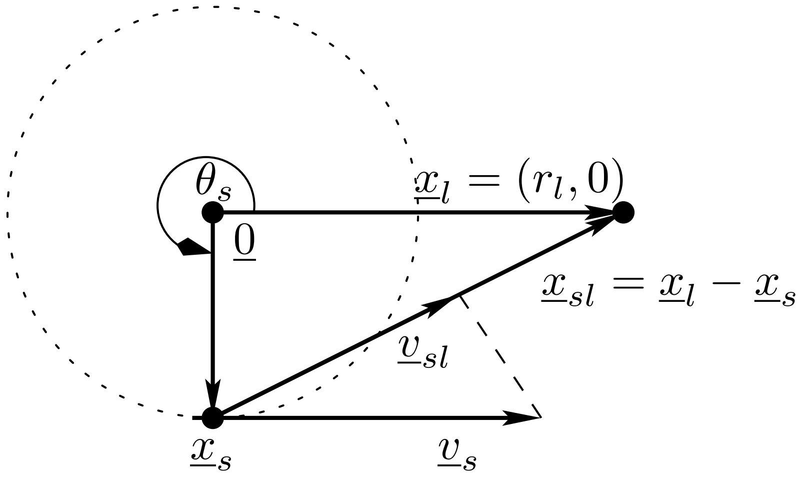
This expression ignores any *directionality* of the horn radiation, and approximates the horn as an omnidirectional radiator located at the same radius for all frequencies.

Horn source-velocity model:

$$\underline{v}_s(t) \;=\; \frac{d}{dt}\underline{x}_s(t) \;=\; \left[\begin{array}{c} -r_s\omega_m\sin(\omega_m t) \\ r_s\omega_m\cos(\omega_m t) \end{array}\right]$$

For circular motion about the origin, tangential velocity is always orthogonal to the position

Rotating Horn Doppler Shift


θ_s
$\underline{x}_l = (r_l, 0)$
$\underline{0}$
$\underline{x}_{sl} = \underline{x}_l - \underline{x}_s$
$\underline{v}_{sl}$
$\underline{x}_s$
$\underline{v}_s$

Relevant geometry for a rotating horn

Source velocity projected onto source-listener path:

$$\underline{v}_{sl} = \mathcal{P}_{\underline{x}_{sl}}(\underline{v}_s) = \frac{\langle \underline{v}_s, \underline{x}_l \rangle}{\| \, \underline{x}_l - \underline{x}_s \, \|^2} \, (\underline{x}_l - \underline{x}_s) \, .$$

Choosing $\underline{x}_l = (r_l, 0)$ yields

$$\underline{v}_{sl} = \frac{-r_l r_s \omega_m \sin(\omega_m t)}{r_l^2 + 2 r_l r_s \cos(\omega_m t) + r_s^2} \left[\begin{array}{c} r_l - r_s \cos(\omega_m t) \\ -r_s \sin(\omega_m) t \end{array} \right] .$$

Far field approximation:

$$\underline{v}_{sl} \approx -r_s \omega_m \sin(\omega_m t) \left[\begin{array}{c} 1 \\ 0 \end{array} \right] .$$

Rotating Horn Doppler Shift, Cont'd

Substituting into the Doppler formula with the listener velocity v_l set to zero yields

$$\omega_l = \frac{\omega_s}{1 + r_s \omega_m \sin(\omega_m t)/c} \approx \omega_s \left[1 - \frac{r_s \omega_m}{c} \sin(\omega_m t) \right] ,$$

where the approximation is valid for small Doppler shifts.

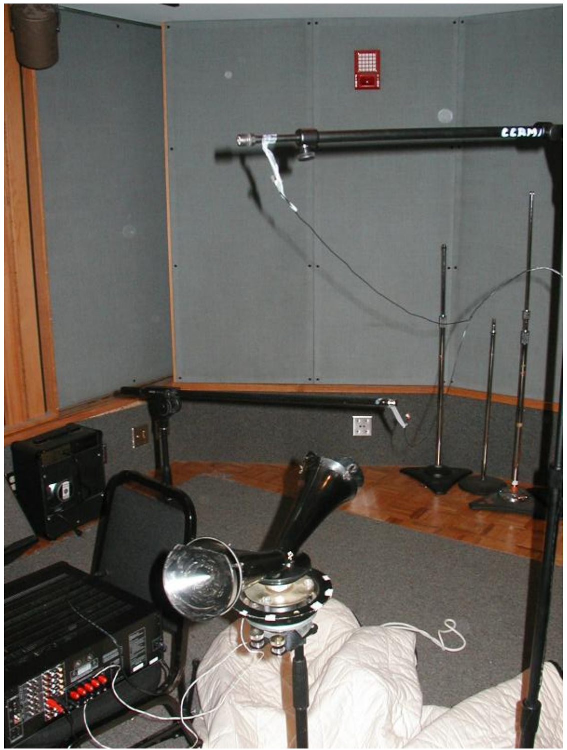
Thus, in the far field, a rotating horn causes an approximately *sinusoidal* multiplicative frequency shift:

> *Doppler shift $=$ horn length r_s times horn angular velocity ω_m divided by sound speed c*

Note that $r_s \omega_m$ is the *tangential speed* of the assumed point of horn radiation.

Computational Modeling of the Leslie

Leslie Free-Field Horn Measurements

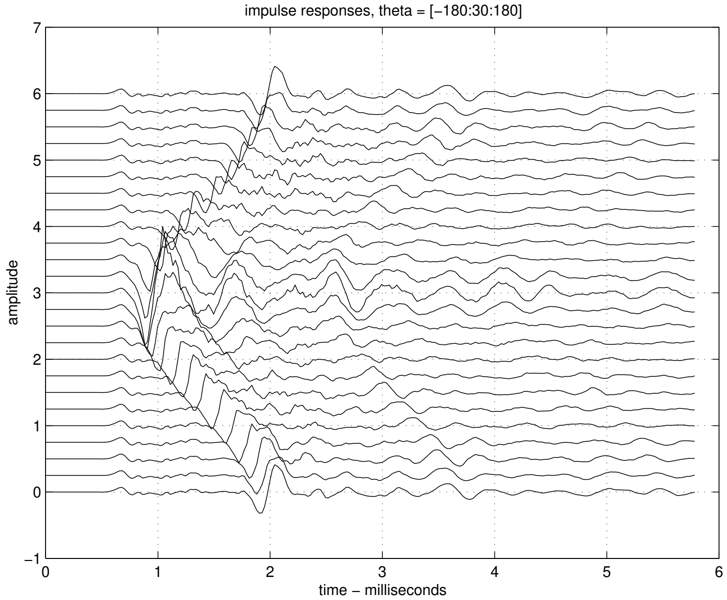


Rotating horn recording set up

Microphone close-up

- Horn set manually to fixed angles from -180 to 180 degrees in increments of 15 degrees
- Impulse response measured using 2048-long Golay-code pairs (see MUS424)

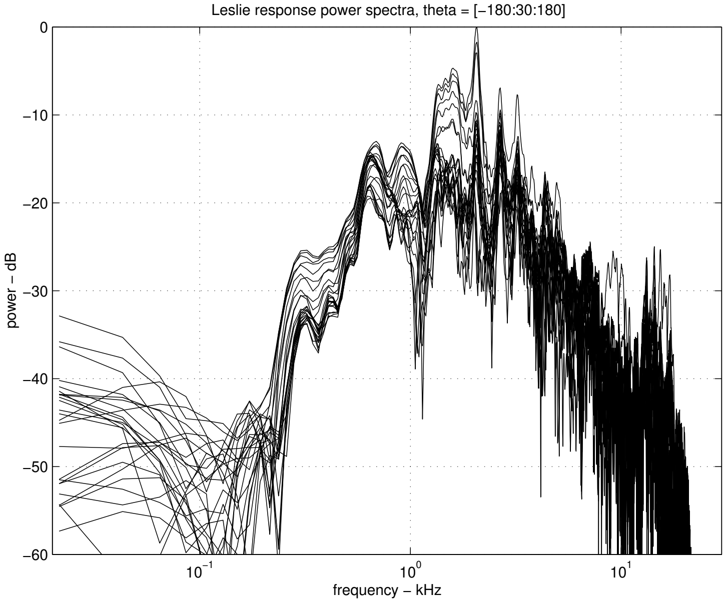
Measured impulse responses



Measured impulse-responses of the Leslie 600 rotating-horn at multiples of 15 degrees. The middle trace is recorded with the microphone along the axis of the horn

Note “leakage” component from base of horn

Measured amplitude responses



Measured amplitude-responses of the Leslie 600 rotating-horn at multiples of 15 degrees

Sound Example

- Input signal (sinusoids)
- Measured horn output signal
- Simulated horn output signal