DOPPLER SIMULATION AND THE LESLIE

Julius Smith, Stefania Serafin

Center for Computer Research in
Music and Acoustics (CCRMA)
Stanford University
http://www-ccrma.stanford.edu
jos|serafin@ccrma.stanford.edu

Jonathan Abel, David Berners

Universal Audio, Inc.
Santa Cruz, California
http://www.uaudio.com
abel@uaudio.com
dpberner@uaudio.com

ABSTRACT

An efficient algorithm for simulating the Doppler effect using interpolating and de-interpolating delay lines is described. The Doppler simulator is used to simulate a rotating horn to achieve the *Leslie* effect. Measurements of a horn from a real Leslie are used to calibrate angle-dependent digital filters which simulate the changing, angle-dependent, frequency response of the rotating horn.

1. INTRODUCTION

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. The Doppler effect has been used to enhance the realism of simulated moving sound sources for compositional purposes [1], and it is an important component of the “Leslie effect.”

The *Leslie* is a popular audio processor used with electronic organs and other instruments [2, 3]. It employs a rotating horn and rotating speaker port to “choralize” the sound. Since the horn rotates within a cabinet, the listener hears multiple reflections at different Doppler shifts, giving a kind of *chorus effect*. Additionally, the Leslie amplifier distorts at high volumes, producing a pleasing “growl” highly prized by keyboard players.

In this paper, an efficient algorithm for digital simulation of the Doppler effect is presented, and the algorithm is applied to the problem of rotating-horn simulation for the Leslie effect. In the next section, the Doppler effect is reviewed, followed by a description of the algorithm for its simulation. Next, the problem of rotating-horn simulation is discussed, followed by a description of a Leslie simulator.

2. THE DOPPLER EFFECT

In the real world, a *Doppler shift* is an apparent change in acoustic frequency content of a sound source due to *motion* of the source relative to the listener. You have probably heard the pitch of a horn drop lower as it passes by (e.g., from a moving train). While a pitched sound-source is moving toward the listener, the heard pitch is higher than it is at rest; while the source is moving away from the listener, its pitch is lowered.

As derived in elementary physics texts, the *Doppler shift* is given by

$$\omega_l = \omega_s \frac{1+\frac{v_{ls}}{c}}{1-\frac{v_{sl}}{c}} \tag{1}$$

where ω_s is the radian frequency emitted by the source at rest, ω_l is the frequency received by the listener, v_{ls} denotes the *speed* of the listener relative to the propagation medium in the direction of the source, v_{sl} denotes the speed of the source relative to the propagation medium in the direction of the listener, and c denotes sound speed. Note that all quantities in this formula are scalars.

2.1. Vector Formulation

Denote the sound-source *velocity* by $\underline{v}_s(\underline{x},t)$ where $\underline{x} \triangleq (x_1,x_2,x_3)$ is 3D position and t is time. Similarly, let $\underline{v}_l(\underline{x},t)$ denote the velocity of the listener, if any. The *position* of source and listener are denoted $\underline{x}_s(t)$ and $\underline{x}_l(t)$, respectively. We have velocity related to position by

$$\underline{v}_s = \frac{d}{dt}\underline{x}_s(t) \qquad \underline{v}_l = \frac{d}{dt}\underline{x}_l(t). \tag{2}$$

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.

The Doppler effect depends only on the relative motion between the source and listener [4, p. 453]. We must 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. (See Fig. 3 for a specific example.)

The *orthogonal projection* of a vector $\underline{x}$ onto a vector $\underline{y}$ is given by [5]

$$\mathcal{P}_{\underline{y}}(\underline{x}) = \frac{\langle \underline{x},\underline{y}\rangle}{\|\,\underline{y}\,\|^2}\underline{y} \triangleq \frac{\underline{x}^T\underline{y}}{\underline{y}^T\underline{y}}\underline{y} \tag{3}$$

Therefore, we can write the projected source velocity as

$$\underline{v}_{sl} = \mathcal{P}_{\underline{x}_{sl}}(\underline{v}_s) = \frac{\langle \underline{v}_s,\underline{x}_{sl}\rangle}{\|\,\underline{x}_{sl}\,\|^2}\underline{x}_{sl} = \frac{\langle \underline{v}_s,\underline{x}_l-\underline{x}_s\rangle}{\|\,\underline{x}_l-\underline{x}_s\,\|^2}\left(\underline{x}_l-\underline{x}_s\right). \tag{4}$$

In the *far field* (listener far away), (4) reduces to

$$\underline{v}_{sl} \approx \frac{\langle \underline{v}_s,\underline{x}_l\rangle}{\|\,\underline{x}_l\,\|^2}\underline{x}_l = \mathcal{P}_{\underline{x}_l}(\underline{v}_s) \qquad (\|\,\underline{x}_l\,\| \gg \|\,\underline{x}_s\,\|). \tag{5}$$

3. DOPPLER SIMULATION

It is well known that a time-varying delay line results in a frequency shift. Time-varying delay is often used, for example, to provide *vibrato* and *chorus* effects [6]. We therefore expect a time-varying delay-line to be capable of precise Doppler simulation.

Consider Doppler shift from a physical point of view. The air can be considered as analogous to a *magnetic tape* which moves from source to listener at speed c. The source is analogous to the write-head of a tape recorder, and the listener corresponds to the read-head. When the source and listener are fixed, the listener receives what the source records. When either moves, a Doppler shift is observed by the listener, according to (1).

3.1. Doppler Simulation via Delay Lines

This analogy also works for a delay-line based computational model. The magnetic tape is now the delay line, the tape read-head is the read-pointer of the delay line, and the write-head is the delay-line write-pointer. In this analogy, it is readily verified that modulating delay by changing the read-pointer increment from 1 to $1 + v_{ls}/c$ (thereby requiring interpolated reads) corresponds to listener motion away from the source at speed v_{ls}. It also follows that changing the *write-pointer* increment from 1 to $1 + v_{sl}/c$ corresponds *source motion toward the listener* at speed v_{sl}. When this is done, we must use *interpolating writes* into the delay memory. Interpolating writes are often called *de-interpolation* [7], and they are formally the graph-theoretic *transpose* of interpolating reads (ordinary "interpolation") [8]. A review of time-varying, interpolating, delay-line reads and writes, together with a method using a single shared pointer, are given in [9].

3.2. Time-Varying Delay-Line Reads

If $x(t)$ denotes the input to a time-varying delay, the output can be written as

$$y(t) = x(t - D_t). \tag{6}$$

where D_t denotes the time-varying delay in seconds. In discrete-time implementations, when D_t is not an integer multiple of the sampling interval, $x(t - D_t)$ may be approximated to arbitrary accuracy (in a finite band) using *bandlimited interpolation* [10] or other techniques for implementation of *fractional delay* [11, 9].

Let's analyze the frequency shift caused by a time-varying delay by setting $x(t)$ to a complex sinusoid at frequency ω_s:

$$x(t) = e^{j\omega_s t} \tag{7}$$

The output is now

$$y(t) = x(t - D_t) = e^{j\omega_s \cdot (t - D_t)}. \tag{8}$$

The instantaneous phase of this signal is

$$\theta(t) = \angle y(t) = \omega_s \cdot (t - D_t) \tag{9}$$

which can be differentiated to give the instantaneous frequency

$$\omega_l = \omega_s(1 - \dot{D}_t) \tag{10}$$

where ω_l denotes the output frequency, and $\dot{D}_t \triangleq \frac{d}{dt}D_t$ denotes the time derivative of the delay D_t. Thus, *the delay growth-rate*, $\dot{D}_t$, equals the *relative frequency downshift*:

$$\dot{D}_t = \frac{\omega_s - \omega_l}{\omega_s}. \tag{11}$$

Comparing (10) to (1), we find that the time-varying delay most naturally simulates Doppler shift caused by a *moving listener*, with

$$\dot{D}_t = -\frac{v_{ls}}{c}. \tag{12}$$

That is, the delay growth-rate, $\dot{D}_t$, should be set to the speed of the listener *away* from the source, normalized by sound speed c.

Simulating *source* motion is also possible, but the relation between delay change and desired frequency shift is more complex, viz., from (1) and (10),

$$\dot{D}_t = -\frac{\frac{v_{ls}}{c} + \frac{v_{sl}}{c}}{1 - \frac{v_{sl}}{c}} \approx -\left(\frac{v_{ls}}{c} + \frac{v_{sl}}{c}\right) \tag{13}$$

where the approximation is valid for $v_{sl} \ll c$. In Section 3.5, a simplified approach is proposed based on moving the delay *input* instead of its output.

3.3. Variable Delay Line in Software

Let `A` denote an array of length N. Then we can implement an M-sample variable delay line in the `C` programming language as shown in Fig. 1. We require, of course, $M \leq N$.

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

Figure 1: *The M-sample variable delay line using separate read- and write-pointers.*

The Synthesis Tool Kit, Version 4 (STK-4) [12] contains the `C++` class "`Delay`" which implements this type of variable (but non-interpolating) delay line. There are additional subclasses which provide *interpolating reads* by various methods. In particular, the class `DelayL` implements continuously variable delay lengths using *linear interpolation.* The code listing in Fig. 1 can be modified to use linear interpolation by replacing the line

```
y = A[rptr++];
```

with

```
long rpi = (long)floor(rptr);
double a = rptr - (double)rpi;
y = a * A[rpi] + (1-a) * A[rpi+1];
rptr += 1;
```

To implement a *continuously* varying delay, we add a "delay growth parameter" g to the delayline function in Fig. 1, and change the line

```
rptr += 1; // pointer update
```

above to

```
rptr += 1 - g; // pointer update
```

When g is 0, we have a fixed delay line, corresponding to $\dot{D}_t = 0$ in (10). When $g > 0$, the delay grows g samples per sample, which we may also interpret as seconds per second, i.e., $\dot{D}_t = g$. By (12), we see that we need

$$g = -\frac{v_{ls}}{c} \tag{14}$$

to simulate a listener traveling toward the source at speed v_{ls}.

Note that when the read- and write-pointers are driven directly from a model of physical propagation-path geometry, they are always separated by predictable minimum and maximum delay intervals. This implies it is unnecessary to worry about the read-pointer *passing* the write-pointers or vice versa. In generic *frequency shifters* [13], or in a Doppler simulator not driven by a changing geometry, a pointer cross-fade scheme may be necessary when the read- and write-pointers get too close to each other.

3.4. Multiple Read Pointers

Using *multiple read pointers*, multiple listeners can be simulated. Furthermore, each read-pointer signal can be filtered to simulate propagation losses and radiation characteristics of the source in the direction of the listener. The read-pointers can move independently to simulate the different Doppler shifts associated with different listener motions and relative source directions.

3.5. Multiple Write Pointers

It is interesting to consider also what effects can be achieved using multiple de-interpolating write pointers. From the considerations in §3.1, we see that multiple write-pointers correspond to multiple write-heads on a magnetic tape recorder. If they are arranged at a fixed spacing, they are equivalent to multiple read pointers, providing a basic multipath simulation. If, however, the write pointers are moving *independently*, they induce *independent Doppler shifts due to source motion.* In particular, each write-pointer can lay down a signal from a separate source to a single listener with its own Doppler shift. Furthermore, each write-signal can be passed through its own filter. Such an individualized source filter can implement all filtering incurred along the propagation path from each source to the listener.

When all write pointers have the same input signal, their filters can be implemented using a *series chain* in which the outputs of successive filters in the chain correspond to progressively longer propagation paths (progressively more filtering). Such an implementation can greatly reduce the filter order required for propagation paths longer than the shortest.

The write-pointers may cross each other with no ill effects, since all but the first[1] simply sum into the shared delay line.

We have seen that a single delay line can be used to simulate any number of moving listeners (§3.4) or any number of moving sources. However, when simulating *both* multiple listeners and multiple sources, it is not possible to share a single delay line. This is because the different listeners do not see the same Doppler shift for each moving source, and while the listener's read-pointer motion can be adjusted to correct for the Doppler shift seen from any particular source, it cannot correct for more than one in general. Thus, in general, we need as many delay lines as there are sources or listeners, whichever is smaller. More precisely, if there are N moving sources and M moving listeners, simulation requires $\min(N, M)$ delay lines.

3.6. Stereo Processing

As a special case, stereo processing of any number of sources can be accomplished using two delay lines, corresponding to left and right stereo channels. The stereo mix may contain a panned mixture of any number sources, each with its own stereo placement, path filtering, and Doppler shift. The two stereo outputs may correspond to "left and right ears," or, more generally, to left- and right-channel microphones in a studio recording set-up.

3.7. System Block Diagram

A schematic diagram of a stereo multiple-source simulation is shown in Fig. 2. To simplify the layout, the input and output signals are all on the right in the diagram. For further simplicity, only one input source is shown. Additional input sources are handled identically, summing into the same delay lines in the same way.

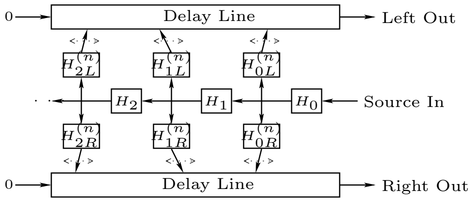

Figure 2: *Block diagram of a stereo simulator for any number of moving sound sources.*

The input source signal first passes through filter $H_0(z)$, which provides *time-invariant* filtering common to all propagation paths. The 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

[1] The "first" write-pointer is defined as the one writing farthest ahead in time; it must *overwrite* memory, instead of summing into it, when a circular buffer is being used, as is typical.

shortest (time-varying) propagation path from the source to each listener. (The (n) superscript here indicates a time-varying filter.) 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 $H_{0R}^{(n)}(z)$ in Fig. 2 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 in Fig. 2, $H_1(z)$, $H_{1L}^{(n)}(z)$, 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 in Fig. 2. 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.

Note also that the source-filters $H_{iL}^{(n)}(z)$ and $H_{iR}^{(n)}(z)$ may include HRTF filtering [14, 15] in order to impart illusory angles of arrival in 3D space.

4. LESLIE SIMULATION

The Leslie consists primarily of a rotating horn and a rotating speaker port inside a wooden cabinet enclosure [3]. This paper will focus on the rotating horn, leaving more complete consideration of the rotating port and cabinet enclosure for a later paper. After deriving a theoretical model of the predicted Doppler shift, laboratory measurements of a real Leslie will be presented and discussed.

4.1. Rotating Horn Simulation

The heart of the Leslie effect is a rotating horn loudspeaker. The rotating horn from a Model 600 Leslie can be seen mounted on a microphone stand in Fig. 4. Two horns are apparent, but one is a dummy, serving mainly to cancel the centrifugal force of the other during rotation. The Model 44W horn is identical to that of the Model 600, and evidently standard across all Leslie models [3]. For a circularly rotating horn, the source position can be approximated as

$$\underline{x}_s(t) = \left[\begin{array}{c} r_s\cos(\omega_m t) \\ r_s\sin(\omega_m t) \end{array}\right] \tag{15}$$

where r_s is the circular radius and ω_m is angular velocity. 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. In the Leslie, a *diffuser* is inserted into the end of the horn in order to make the radiation pattern closer to uniform [3], so the omnidirectional assumption is reasonably accurate.

By (2), the source velocity for the circularly rotating horn is

$$\underline{v}_s(t) = \frac{d}{dt}\underline{x}_s(t) = \left[\begin{array}{c} -r_s\omega_m\sin(\omega_m t) \\ r_s\omega_m\cos(\omega_m t) \end{array}\right] \tag{16}$$

Note that the source velocity vector is always orthogonal to the source position vector, as indicated in Fig. 3.

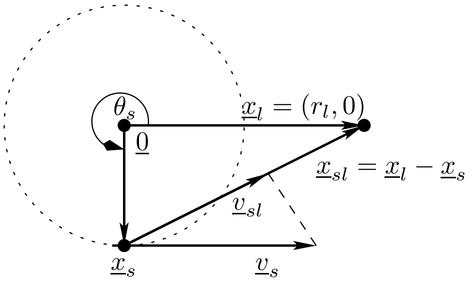
θ_s
$\underline{0}$
$\underline{x}_l = (r_l, 0)$
$\underline{x}_{sl} = \underline{x}_l - \underline{x}_s$
$\underline{v}_{sl}$
$\underline{x}_s$
$\underline{v}_s$

Figure 3: *Relevant geometry for a rotating horn.*

Since $\underline{v}_s$ and $\underline{x}_s$ are orthogonal, the projected source velocity (4) simplifies to

$$\underline{v}_{sl} = \mathcal{P}_{\underline{x}_{sl}}(\underline{v}_s) = \frac{\langle \underline{v}_s, \underline{x}_l\rangle}{\|\, \underline{x}_l - \underline{x}_s \,\|^2}\,(\underline{x}_l - \underline{x}_s)\,. \tag{17}$$

Arbitrarily choosing $\underline{x}_l = (r_l, 0)$ (see Fig. 3), and substituting (15) and (16) into (17) yields

$$\underline{v}_{sl} = \frac{-r_l r_s\omega_m\sin(\omega_m t)}{r_l^2 + 2r_l r_s\cos(\omega_m t) + r_s^2}\left[\begin{array}{c} r_l - r_s\cos(\omega_m t) \\ -r_s\sin(\omega_m)t \end{array}\right]. \tag{18}$$

In the far field, this reduces simply to

$$\underline{v}_{sl} \approx -r_s\omega_m\sin(\omega_m t)\left[\begin{array}{c} 1 \\ 0 \end{array}\right]. \tag{19}$$

Substituting into the Doppler expression (1) with the listener velocity v_l set to zero yields

$$\omega_l = \frac{\omega_s}{1 + r_s\omega_m\sin(\omega_m t)/c} \approx \omega_s\left[1 - \frac{r_s\omega_m}{c}\sin(\omega_m t)\right], \tag{20}$$

where the approximation is valid for small Doppler shifts. Thus, in the far field, a rotating horn causes an approximately *sinusoidal* multiplicative frequency shift, with the amplitude given by 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.

4.2. Leslie Free-Field Horn Measurements

The free-field radiation pattern of a Model 600 Leslie rotating horn was measured using the experimental set-up shown in Fig. 4. A matched pair of Panasonic microphone elements (Crystal River Snapshot system) were used to measure the horn response both in the plane of rotation and along the axis of rotation (where no Doppler shift or radiation pattern variation is expected). The microphones were mounted on separate boom microphone stands, as shown in the figure. A close-up of the plane-of-rotation mic is shown in Fig. 5.

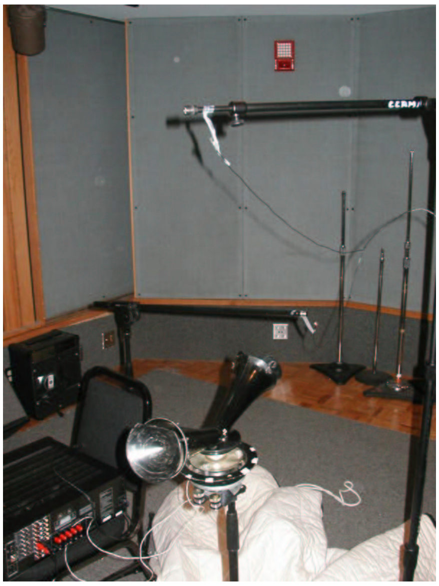

Figure 4: *Rotating horn recording set up.*

Figure 5: *Microphone close-up.*

The horn was set manually to fixed angles from -180 to 180 degrees in increments of 15 degrees, and at each angle the impulse response was measured using 2048-long Golay-code pairs [16].

Figure 6 shows the measured impulses responses and Fig. 7 shows the corresponding amplitude responses at the various angles. Note that the beginning of each impulse response contains a fixed portion which does not depend significantly on the angle. This is thought to be due to “leakage” from the base of the horn. It arrives first since the straight-line path from the enclosed speaker to the microphone is shorter than that traveling through the horn assembly.

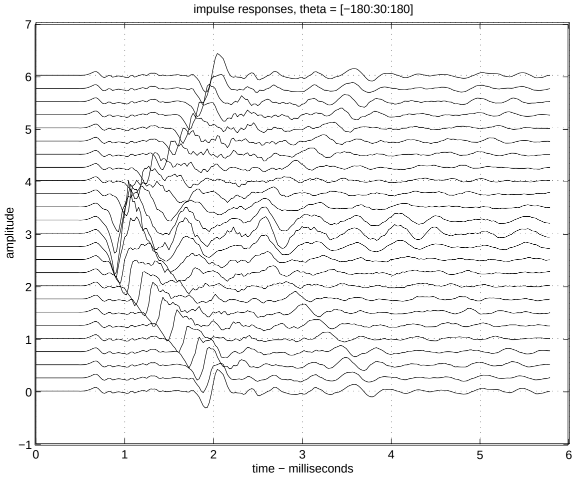

Figure 6: *Measured impulse-responses of the Leslie 600 rotating-horn at multiples of 15 degrees. The middle trace is recorded with the microphone along the axis of the horn.*

4.3. Separating Horn Output from Base Leakage

Since Fig. 6 indicates the existence of fixed and angle-dependent components in the measured impulse responses, and since such angle-independent component is strongly suppressed by baffling in the cabinet enclosure, it is desirable to eliminate this fixed component from the measurements. For this purpose, an iterative algorithm was developed which models the two components separately.

Let $M = 256$ denote the number of impulse-response samples in each measured impulse response,and let $N = 25$ denote the number of angles (`-180:15:180`) at which impulse-response measurements were taken. We denote the $M \times N$ impulse-response matrix by $\mathbf{h}$. Each column of $\mathbf{h}$ is an impulse response at some horn angle. (Figure 6 can be interpreted as a plot of the *transpose* of $\mathbf{h}$.)

We model $\mathbf{h}$ as

$$\mathbf{h} = \boldsymbol{\alpha} + \boldsymbol{\gamma} \cdot \text{diag}(z^{-\tau_i}) + \mathbf{e} \tag{21}$$

where τ_i is the arrival-time delay, in samples, for the horn output in the ith row (the delays clearly visible in Fig. 6 as a function of angle). These arrival times are estimated as the location of the peak in the cross-correlation between the ith impulse response and the same impulse response after converting it to minimum phase [8]. The diagonal matrix $\text{diag}(z^{-\tau_i})$ denotes a *shift operator* which delays the ith column of $\boldsymbol{\gamma}$ by τ_i samples. Thus, $\boldsymbol{\gamma}$ contains the horn-output impulse response (without the base leakage) shifted to time zero (i.e., the angle-dependent delay is removed). Finally, the error matrix $\mathbf{e}$ is to be minimized in the least-squares sense.

Each column of the matrix $\boldsymbol{\alpha}$ contains a copy of the estimated horn-base leakage impulse-response:

$$\boldsymbol{\alpha} = \underline{a} \cdot \underline{1}^T \tag{22}$$

where $\underline{1}^T = [1, 1, \ldots, 1]$.

The estimated angle-dependent impulse-responses in $\boldsymbol{\gamma}$ are modeled as linear combinations of $K = 5$ *fixed* impulse

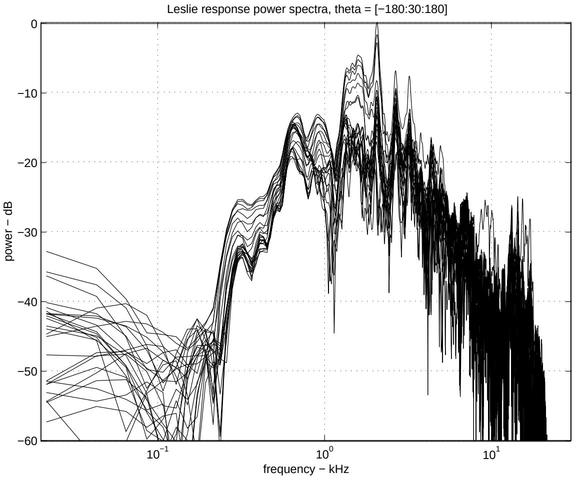

Figure 7: *Measured amplitude-responses of the Leslie 600 rotating-horn at multiples of 15 degrees.*

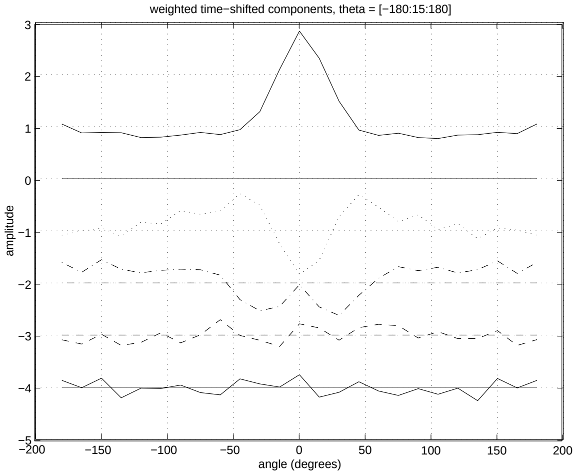

Figure 8: *First 5 principal components weighted by their corresponding singular values. Each angle-dependent impulse response is modeled as a linear combination of these angle-independent impulse-response components.*

responses, viewed (loosely) as *principal components*:

$$\boldsymbol{\gamma} = \mathbf{g} \cdot \mathbf{w} \tag{23}$$

where $\mathbf{g}$ is the $M \times K$ orthonormal matrix of fixed filters (principal components), and $\mathbf{w}$ is a $K \times N$ matrix of *weights*, found in the usual way by a truncated *singular value decomposition* (SVD) [17].

4.3.1. Algorithm

To start the separation algorithm, $\boldsymbol{\gamma}_0$ is initialized to the zero-shifted impulse response data $\mathbf{h} \cdot \text{diag}(z^{\tau_i})$, ignoring the tails of the base-leakage they may contain. Then $\boldsymbol{\alpha}_0$ is estimated as the mean of $\mathbf{h} - \boldsymbol{\gamma}_0 \text{diag}(z^{-\tau_i})$. This mean is then subtracted from $\mathbf{h}$ to produce $\mathbf{b}_1 = (\mathbf{h} - \boldsymbol{\alpha}_0)\text{diag}(z^{-\tau_i})$ which is then then converted to $\boldsymbol{\gamma}_1 = \mathbf{g}_1 \cdot \mathbf{w}_1$ by a truncated SVD. A revised base-leakage estimate $\boldsymbol{\alpha}_1$ is then formed as $\mathbf{h} - \boldsymbol{\gamma}_1 \text{diag}(z^{-\tau_i})$, and so on, until convergence is achieved.

4.3.2. Results

Figure 8 plots the $K = 5$ weighted principal components identified for the angle-dependent component of the horn radiativity. Each component is weighted by its corresponding singular value, thus visually indicating its importance. Also plotted using the same line type are the zero-lines for each principal component. Note in particular that the first (largest) principal component is entirely positive.

Figure 9 shows the complete horn impulse-response model $(\boldsymbol{\alpha} + \boldsymbol{\gamma} \cdot \text{diag}(z^{-\tau_i}))$, overlaid with the original raw data $\mathbf{h}$. We see that both the fixed base-leakage and the angle-dependent horn-output response are closely followed by the fitted model.

Figure 10 shows the estimated impulse response of the base-leakage component $\underline{a}(n)$, and Fig. 11 shows the modeled angle-dependent horn-output components $\boldsymbol{\gamma}$ delayed out to their natural arrival times.

Figure 12 shows the average power response of the horn outputs. Also overlaid in that figure is the average response smoothed according to Bark frequency resolution [18]. This equalizer then becomes $H_0(z)$ in Fig. 2. The filters $H_{0L}(z)$ and $H_{0R}(z)$ in Fig. 2 are obtained by dividing the Bark-smoothed frequency-response at each angle by $H_0(z)$ and designing a low-order recursive filter to provide that equalization dynamically as a function of horn angle. The impulse-response arrival times τ_i determine where in the delay lines the filter-outputs are to be summed in Fig. 2.

Figure 13 shows a spectrogram view of the angle-dependent amplitude responses of the horn with $H_0(z)$ (Bark-smoothed curve in Fig. 12) divided out. This angle-dependent, differential equalization is used to design the filters $H_{0L}(z)$ and $H_{0R}(z)$ in Fig. 2. Note that below 12 Barks or so, the angle-dependence is primarily to decrease amplitude as the horn points away from the listener, with high frequencies decreasing somewhat faster with angle than low frequencies.

5. ROTATING WOOFER-PORT AND CABINET SIMULATION

Due to space limitations, we cannot yet present our measurements and analysis of (1) the rotating woofer port, and (2) the rotating horn *en vivo* (within its wooden cabinet enclosure). It is straightforward to extend our computational model to include these elements as follows:

- In [3], it is mentioned that an AM "throb" is the main effect of the rotating woffer port. A modulated lowpass-filter cut-off frequency has been used for this purpose by others. Our measured data will be used to construct angle-dependent filtering in a manner analogous to that of the rotating horn, and this "woofer filter" runs in parallel with the rotating horn model.
- The Leslie cabinet multiply-reflects the sound emanating from the rotating horn. The first few early reflections are simply handled as additional sources in Fig. 2. We are working to extend the impulse-response-component separation algorithm of §4.3.1 to

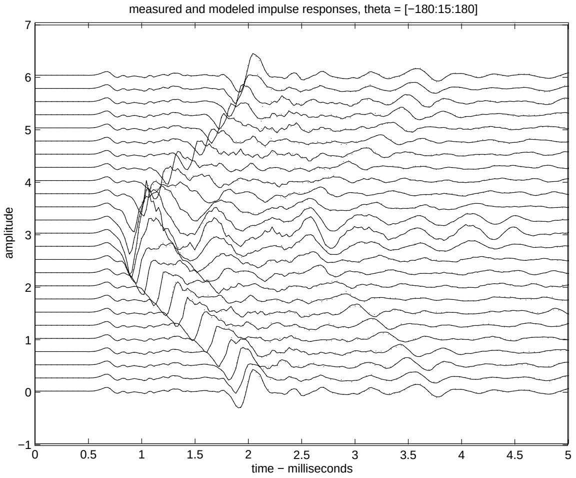

Figure 9: *Overlay of measured (solid) and modeled (dotted) impulse-responses at multiples of 15 degrees.*

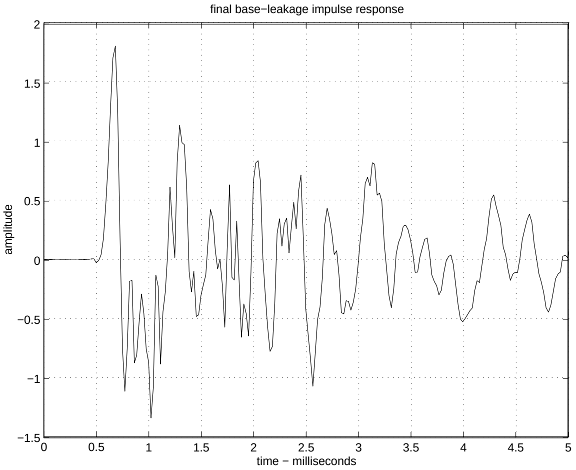

Figure 10: *Modeled base-leakage impulse-response (angle-independent).*

the case of superimposed early reflections in the impulse response, and preliminary results are promising.

- To qualitatively simulate later, more *reverberant* reflections in the Leslie cabinet, we feed a portion of the rotating-horn and speaker-port signals to separate states of an *artificial reverberator* [19]. This reverberator may be configured as a "very small room" corresponding to the dimensions and scattering characteristics of the Leslie cabinet, and details of the response may be calibrated using measurements of the impulse response of the Leslie cabinet. Finally, in order to emulate the natural spatial diversity of a radiating Leslie cabinet in a room, "virtual cabinet vent outputs" can be extracted from the model and fed into separate states of a *room reverberator*.

In summary, we use multiple interpolating write-pointers to individually simulate the early cabinet reflections, and a

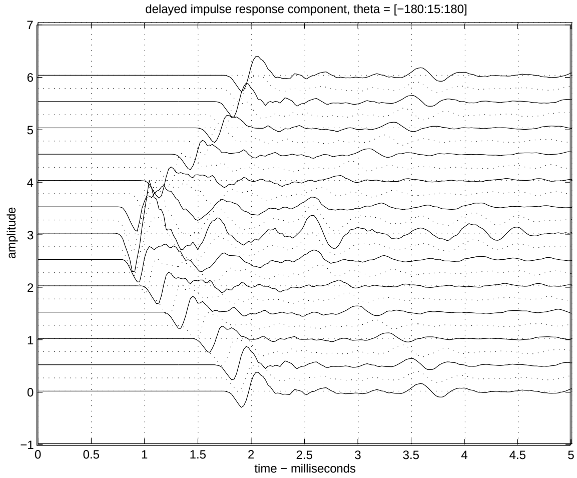

Figure 11: *Modeled horn-output impulse-responses at multiples of 15 degrees.*

"Leslie cabinet" reveberator for handling later reflections more statistically.

6. CONCLUSIONS

A computational model of Doppler shift was presented, accommodating any number of moving sound sources and moving listeners. For Leslie simulation, multiple sources correspond to the direct and reflected signals from the rotating horn, and two fixed listeners correspond to two ears or two studio microphones. Measurements were made on a real Leslie in order to calibrate the angle-dependent filters corresponding to each propagation path geometry from horn to listener. It was found that the horn impulse response had prominent angle-dependent and angle-independent components, and a relaxation algorithm was devised for separating them.

7. ACKNOWLEDGEMENTS

Many thanks to Will Shanks of *Ear to the Ground Studio* for loaning us the Leslies and for setting up (and choosing) the mics, and to David Romblom for help in the lab in numerous capacities.

8. REFERENCES

[1] John M. Chowning, "The simulation of moving sound sources," *J. Audio Engineering Soc.*, vol. 19, pp. 2–6, 1971.

[2] Harald Bode, "History of electronic sound modification," *J. Audio Engineering Soc.*, vol. 32, no. 10, pp. 730–739, October 1984.

[3] Clifford A. Henricksen, "Unearthing the mysteries of the Leslie cabinet," *Recording Engineer/Producer magazine*, April 1981, available online at `http://www.theatreorgans.com/hammond/faq/mystery/mystery.html`.

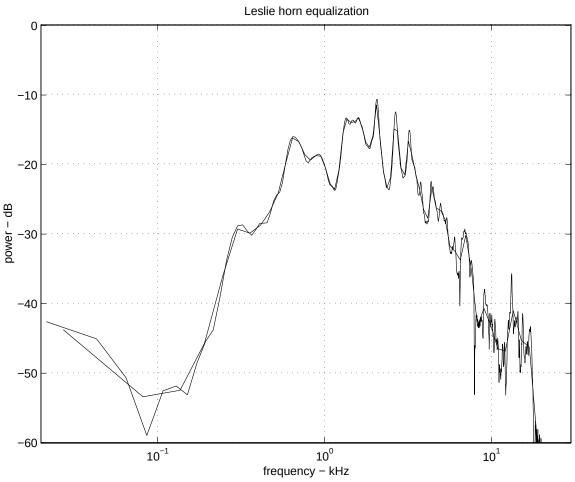

Figure 12: *Average angle-dependent amplitude response overlaid with Bark-smoothed response to be used as a fixed equalization applied to the source.*

[4] Allan D. Pierce, *Acoustics*, American Institute of Physics, for the Acoustical Society of America, (516)349-7800 x 481, 1989.

[5] Julius O. Smith III, *Mathematics of the Discrete Fourier Transform (DFT)*, December 20, 2001: `http://www-ccrma.stanford.edu/~jos/mdft/`

[6] Craig Anderton, *Digital Delay Handbook*, Amsco Publications, New York, 1985.

[7] Vesa Välimäki, *Discrete-Time Modeling of Acoustic Tubes Using Fractional Delay Filters*, Ph.D. thesis, Electrical Engineering, Helsinki University of Technology, Dec. 1995, `http://www.acoustics.hut.fi/~vpv/publications/vesa_phd.html`.

[8] Alan V. Oppenheim and R. Schafer, *Digital Signal Processing*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1975.

[9] Davide Rocchesso, "Fractionally addressed delay lines," *IEEE Trans. Speech and Audio Processing*, vol. 8, no. 6, pp. 717–727, November 2000.

[10] Julius O. Smith and P. Gossett, "A flexible sampling-rate conversion method," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing, San Diego*, New York, March 1984, vol. 2, pp. 19.4.1–19.4.2, IEEE Press. Expanded tutorial and associated free software available at `http://www-ccrma.stanford.edu/~jos/resample/`.

[11] Timo I. Laakso, Vesa Välimäki, Matti Karjalainen, and Unto K. Laine, "Splitting the Unit Delay—Tools for Fractional Delay Filter Design," *IEEE Signal Processing Magazine*, vol. 13, no. 1, pp. 30–60, January 1996.

[12] Perry Cook and Gary Scavone, *Synthesis Tool Kit in C++, Version 4.0*, March 2002: `http://www-ccrma.stanford.edu/CCRMA/Software/STK/`

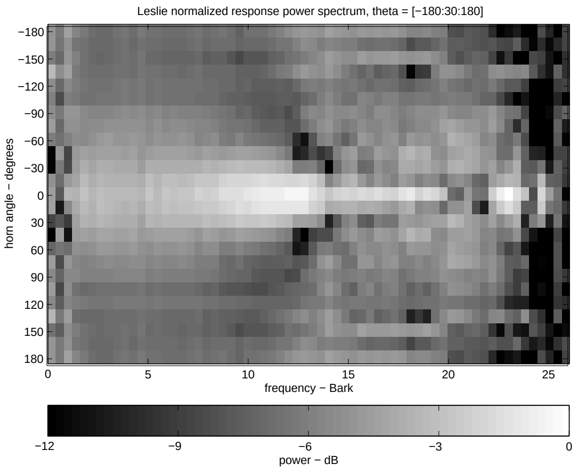

Figure 13: *Angle-dependent amplitude response divided by Bark-smoothed average response to be used as the basis for design of time-varying, angle-dependent equalization to be applied after $H_0(z)$.*

[13] Jean Laroche, "Time and pitch scale modification of audio signals," in *Applications of DSP to Audio & Acoustics*, Mark Kahrs and Karlheinz Brandenburg, Eds., pp. 279–309. Kluwer Academic Publishers, 1998.

[14] Jens Blauert, *Spatial Hearing. The Psychophysics of Human Sound Localization*, MIT Press, Cambridge, MA, USA, 1997.

[15] Elizabeth M. Wenzel, "Localization in virtual acoustic displays," *Presence*, vol. 1, no. 1, pp. 80–107, Summer 1992.

[16] Scott Foster, "Impulse response measurements using golay codes," *Proc. IEEE ICASSP-86*, pp. 929–932, 1986.

[17] Gene. H. Golub and Charles F. Van Loan, *Matrix Computations, 2nd Edition*, The Johns Hopkins University Press, Baltimore, 1989.

[18] Julius O. Smith and Jonathan S. Abel, "Bark and ERB bilinear transforms," *IEEE Trans. Speech and Audio Processing*, pp. 697–708, November 1999. Matlab code for the main figures are available online at `http://www-ccrma.stanford.edu/~jos/bbt/`.

[19] Julius O. Smith III, "Artificial Reverberation," in *Digital Waveguide Modeling of Musical Instruments*, CCRMA, Stanford University, May 9, 2002: `http://www-ccrma.stanford.edu/~jos/waveguide/Artificial_Reverberation.html`