

1 Electromechanical Effects

Stefan Bilbao

A rather special family of audio effects consists of those which employ mechanical elements in conjunction with analog electronics—though many classic effects rely on such components (such as, for example, the Leslie speaker, or tape delays), for some problems in the world of virtual analog, an involved treatment of the mechanical part of the effect is necessary, in order to be able to capture the perceptually salient quality of the effect. This is particularly the case for electromechanical artificial reverberation, which exhibits an extremely complex response. Two interesting cases, namely plate and spring reverberation, will be briefly described here.

The attribute of these devices which distinguishes them from other electromechanical effects is that the mechanical components cannot be modelled as "lumped"; structurally, a spring or a plate occupies space, and its vibration pattern varies from one point to the next. It is precisely the distributed character of these components which gives these effects their complex sound—and which, at the same time, requires a somewhat different approach to emulation. One must now think of the behaviour of the component in terms of various different modes of vibration, or in terms of waves which require finite propagation times, just like real acoustic spaces. Indeed, spring and plate reverbs were originally intended as convenient substitutes for real room reverbs—but developed a loyal audience of their own. Originally, such popularity was probably in spite of this distinction, but as such units have become scarce, convenient digital substitutes for these sometimes bulky and damage-prone units have become sought-after—and make for a fascinating application of virtual analog modeling!

Distributed modelling is seen only rarely in the world of digital audio effects, but plays, of course, a central role in sound synthesis based on physical models of musical instruments—a topic which is too broad to fit into this volume! In this short section, some of the basics of distributed modelling for electromechanical elements are covered, from a high-level perspective, focussing on the nature of the phenomena involved, and some simulation techniques (and in particular, finite difference schemes). It's probably worth mentioning that distributed modelling can be a lot more expensive, in terms of number-crunching, than other analog effect emulation—which is true of any digital reverberation algorithm. But, these days, standard computer hardware is becoming powerful enough to tackle these modeling problems in real time, or something close to it.

1.1 Room Reverberation and the 3D Wave Equation

When looking at artificial reverberation techniques, it's useful to first take a look at the behaviour of real rooms—to understand the ways in which electromechanical effects attempt to emulate this real-world effect, and, more importantly, how they differ!

In air, to a very good approximation, waves travel at a constant speed, here referred to as c . c has a mild dependence on temperature, and is approximately 343 m/s at atmospheric pressure, and near room temperature [12]. From a given acoustic point source, waves travel uniformly in all directions (isotropically) at this speed, and there is a reduction in amplitude as the waves spread out. Formally, to a very good approximation, the time evolution of a pressure field in three dimensions may be described by the wave equation:

$$\frac{\partial^2 p}{\partial t^2} = c^2 \nabla^2 p \quad (1)$$

where here, $p(x, y, z, t)$ is the pressure at spatial coordinates x , y and z , and at time t . ∇^2 is the 3D Laplacian operator. For more on this equation, see [19].

A key concept in all distributed modeling for linear systems is a dispersion relation, relating angular frequency ω (in radians/sec) to a vector wavenumber β (measured in units of m^{-1}) for a wave component of the solution. Such a component will be of the form

$$p(x, t) = e^{j\omega t + \beta \cdot x} \quad (2)$$

in 3D, and the dispersion relation for the wave equation above takes the following form:

$$\omega = c|\beta| \quad (3)$$

It turns out to be convenient to use ω and β here, but one could equally well write this relationship in terms of frequency $f = \omega/2\pi$ and wavelength $\lambda = 2\pi/|\beta|$ —the above dispersion relation then becomes the familiar expression $c = \lambda f$. In the case of the wave equation, this expression is of a very simple form; for other systems it will not be, and often, even for a single system, there may be many such relations, which may not be expressible in simple closed form. The dispersion relation, in general, yields a huge amount of information about the behaviour of a system which is very important acoustically. When a system's geometry is specified, one can say a great deal not only about acoustic attributes such as echo density and mode density, but also about how much computational power will be required to perform a digital emulation!

Expressions for wave speed as a function of frequency may be derived from a dispersion relation as:

$$v_{phase} = \frac{\omega}{|\beta|} \quad v_{group} = \frac{d\omega}{d|\beta|} \quad (4)$$

v_{phase} is the phase velocity, or propagation speed of a wave at wavenumber β , and v_{group} is the group, or packet velocity. In the case of rooms, from the dispersion relation given above, these speeds are both c . Such wave propagation is often referred to as non-dispersive; all disturbances propagate at the same speed, regardless of wavelength or frequency. In the present case of wave propagation in air, it also implies that waveforms which propagate from one point to another preserve shape, or remain coherent during propagation.

When a given delimited acoustic space is defined, the obvious acoustic effect is that of echoes. For a room with reflective walls, a typical response consists of series of distinct echoes (early reflections) which are perceptually distinct, and which serve as localization cues, followed by increasingly dense late reflections, and, finally when reflections become so dense so as to be not perceptually distinct, a reverberant tail. It is room geometry which gives acoustic responses their characteristic complexity—and this is definitely not the case for plate and spring reverberation, which emulate this response using relatively simple structures.

Digital emulation of 3D acoustic spaces, through the direct solution of the wave equation is a daunting task, from the point of view of computational complexity (though it is an active research topic [8, 20], which will surely eventually bear fruit). In fact, by further examination of the dispersion relation one may arrive at bounds on computational cost: For any simulation of the 3D wave equation over a space of volume V , and which is intended to produce an approximation to the response over a band of frequencies from 0 Hz, to $f_s/2$ Hz, for some frequency f_s (i.e., an audio sample rate), the amount of memory required will be approximately:

$$N(f_s) \propto V f_s^3 / c^3 \quad (5)$$

where the constant of proportionality depends on the particular method employed. For $c = 340$, and at at typical audio rate $f_s = 48$ kHz, and for even a moderately sized room ($V = 1000 \text{ m}^3$), this can be very large indeed—here, for (say) a finite difference scheme $N = 2.8 \times 10^9$! Because arithmetic operations will presumably be performed on all the data, locally, at the audio rate,

the operation count will be on the order of $N \times f_s$ flops, which is approaching the petaflop range. For more on the topic of computational complexity estimates in physical audio applications, see [5].

Most reverberation techniques are thus based around perceptual simplifications, as described in detail in Chapter ???. In industrial acoustical applications, methods based on the image source method [2] are popular, and rely on simplified models of reflection from surfaces; in audio, feedback delay networks [22, 23, 15], which are very efficient but not based on direct solution of the 3D wave equation are commonly employed.

1.2 Plates and Plate Reverberation

Plate reverberation was originally intended as a convenient means of applying a reverberant effect to a recorded sound [16], long before the advent of digital reverberation. Such effects were extensively researched and eventually commercialized, with the EMT-140 the most successful resulting product. The basic operation of such a unit is relatively simple: a metal plate, normally rectangular, made of steel and supported by a frame is driven by an input signal through an actuator, and outputs are read at a pair of pickups. See Figure 1(a). Such a device is often complemented by adjustable absorbing plates which allow for some control over the decay time. The response of a plate is generally quite different from that of room acoustic reverberation—distinct reflections are notably absent, and the response as a whole has a smooth noise-like character.

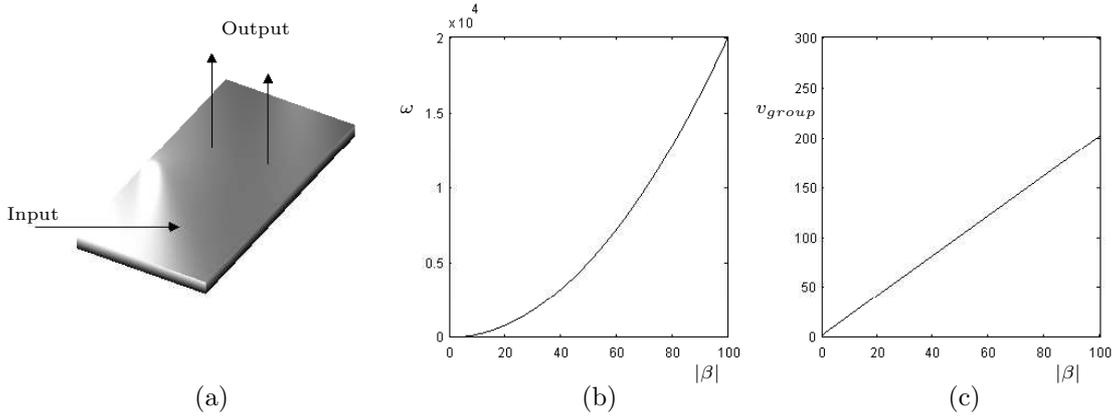


Figure 1: (a) A metal plate, driven by an input signal, and from which output(s) are read at distinct locations. (b) Dispersion relation, and (c) group velocity for a thin plate.

A given plate is characterized by its density ρ , in kg/m^3 , Young’s modulus E , in $\text{kg}/\text{s}^2\text{m}$, Poisson’s ratio ν (dimensionless), and geometrical properties such as its thickness H in m , and its surface area A in m^2 , as well as the particular boundary conditions. A typical reverberation unit is constructed from steel, with a thickness of approximately 0.5 mm , and with a surface area of approximately 2 m^2 (other materials and dimensions are also in use—the much smaller EMT 240 unit employs thin gold foil instead of steel). Usually the edges are very nearly free to vibrate, except for at the plate supports. For a sufficiently thin lossless plate, the equation of motion may be written succinctly as

$$\frac{\partial^2 u}{\partial t^2} = -\kappa^2 \nabla^2 \nabla^2 u \quad (6)$$

where ∇^2 is the 2D Laplacian operator defined, in Cartesian coordinates (x, y) , as

$$\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \quad (7)$$

For more on the physics of plates, see, e.g., [12, 13, 19].

The dispersion relation for a plate is very different from that corresponding to wave propagation in air:

$$\omega = \kappa\beta^2 \quad (8)$$

where $\kappa = \sqrt{EH^2/(12(1-\nu^2)\rho)}$. Now, the phase and group velocities are no longer constant:

$$v_{phase} = \kappa\beta = \sqrt{\kappa\omega} \quad v_{group} = 2\kappa\beta = 2\sqrt{\kappa\omega} \quad (9)$$

See Figure 1 (b) and (c). As a result, wave propagation is dispersive, and highly so. See Figure 2, showing the time evolution of plate displacement in response to a pulse. Notice in particular that the high frequency components of the pulse travel more quickly than the low frequency components—and thus the pulse rapidly loses coherence as it travels. Another related distinction between plate and room responses is the mode density, which is roughly constant in the case of the plate, but which increases as the square of frequency for a room. See Figure 3.

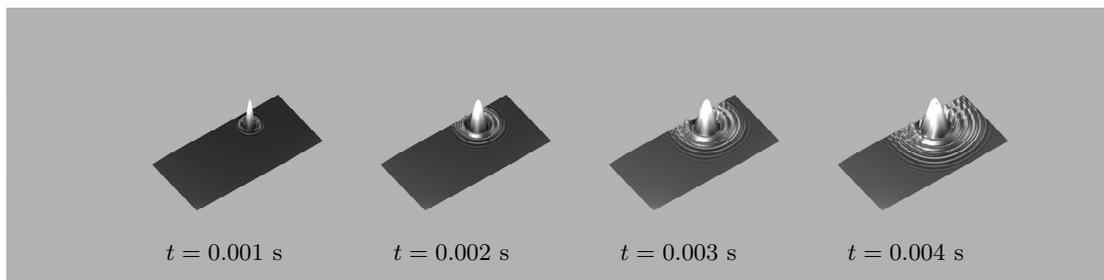


Figure 2: *Time evolution of plate displacement, with $\kappa = 0.5$, and of aspect ratio 2, in response to a pulse, at times as indicated.*

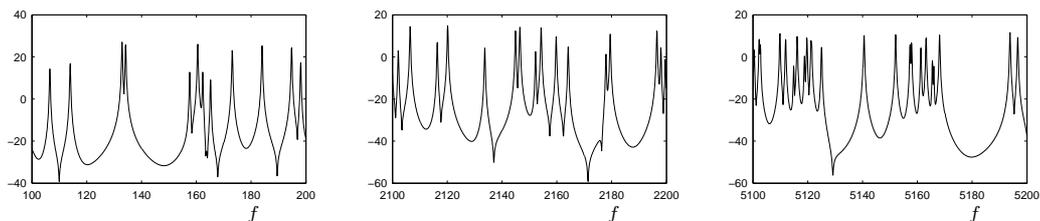


Figure 3: *Spectrum of impulse response of a plate, with $\kappa = 2$, and of aspect ratio 2, over different frequency ranges, as indicated.*

One may show, as in the case of the wave equation, that for a plate of stiffness parameter κ , and of surface area A , that the amount of memory required will be approximately

$$N(f_s) \propto Af_s/\kappa \quad (10)$$

which, for typical plate materials and geometries ($\kappa = 0.737$, $A = 2$), and at an audio sample rate such as 48 kHz, is not particularly large (here, for a finite difference scheme, $N = 8.29 \times 10^4$). The

dependence of memory requirements may be directly related to both the dispersive character of the plate, as well as the constancy of mode density [5]. The operation count will again scale with $N \cdot f_s$ —which, in comparison with full 3D modeling, is in the Gflop range, which is expensive still by today’s standards but not unreasonably so. As such, it is worth looking at how one might design such a simulation through the direct time/space solution of the plate equation.

Case Study: Finite Difference Plate Reverberation

The emulation of distributed models, such as plates or springs requires a different treatment from lumped analog components—probably the best-known distributed modeling techniques in sound synthesis are digital waveguides [24], which are extremely efficient for problems in 1D, but for which such an efficiency advantage does not extend to problems in higher dimensions. Mainstream time-domain numerical simulation techniques (of which finite difference schemes [26] are the easiest to understand, and simplest to program) are an attractive alternative. Here, a very simple time domain finite difference scheme [26] will be presented, based on the direct solution to the equation of motion (6) for a thin plate. Finite difference schemes for plates are discussed by various authors (see, e.g., [27]), and have been used in musical acoustics applications [17], as well as in the present case of plate reverberation [3, 6, 4].

Equation (6) describes free vibration of a lossless thin plate. A simple extension to the more realistic case of frequency-independent loss, and where the plate is driven externally (as in a reverberation unit) is

$$\frac{\partial^2 u}{\partial t^2} = -\kappa^2 \nabla^2 \nabla^2 u - 2\sigma \frac{\partial u}{\partial t} + \delta(x - x_i, y - y_i) f(t) / \rho H \quad (11)$$

Here $\sigma \geq 0$ is a loss parameter—it is related to a global T_{60} reverberation time by $T_{60} = 6 \ln(10) / \sigma$. More realistic (frequency dependent) models of loss are available [9], but the above is sufficient for a simple demonstration of finite difference techniques. $f(t)$ is a force signal in N, assumed known, and applied at the location $x = x_i, y = y_i$, where here, δ is a Dirac delta function.

The first step in the design of any audio effect is the choice of a sample rate f_s . In simulation applications, it is more natural to make use of a time step $T = 1/f_s$. Another choice which must be made of that of a spatial grid, at the points of which an approximate solution to a partial differential equation will be computed. In the case of a plate defined over a rectangular domain $x \in [0, L_x]$, and $y \in [0, L_y]$, the obvious choice is a rectangular grid, of spacing X between adjacent points. The grid function $u_{l,m}^n$ then represents an approximation to the solution $u(x, y, t)$ of (6), at times $t = nT$, and at locations $x = lX$, and $y = mX$, for integer n, l and m . See Figure 4(a). It makes sense to choose X such that the grid spacing divides the side lengths as evenly as possible, i.e., $L_x/X \cong N_x, L_y/X \cong N_y$ for some integers N_x and N_y . It is not possible to do this exactly for arbitrary side lengths—to remedy this, one could choose distinct grid spacings X and Y in the x and y directions, but this subtle point will not be explored further here. There are, of course, many other ways of choosing grids, which may or may not be regular—finite element methods [10], for example, are usually designed to operate over unstructured grids, and are suitable for problems in irregular geometries. Consider first the first and second partial time derivatives which appear in (6). The simplest finite difference approximations, δ_t and δ_{tt} may be written as

$$\delta_t u_{l,m}^n = \frac{1}{2T} \left(u_{l,m}^{n+1} - u_{l,m}^{n-1} \right) \cong \frac{\partial u}{\partial t} (x = lX, y = mX, t = nT) \quad (12)$$

$$\delta_{tt} u_{l,m}^n = \frac{1}{T^2} \left(u_{l,m}^{n+1} - 2u_{l,m}^n + u_{l,m}^{n-1} \right) \cong \frac{\partial^2 u}{\partial t^2} (x = lX, y = mX, t = nT) \quad (13)$$

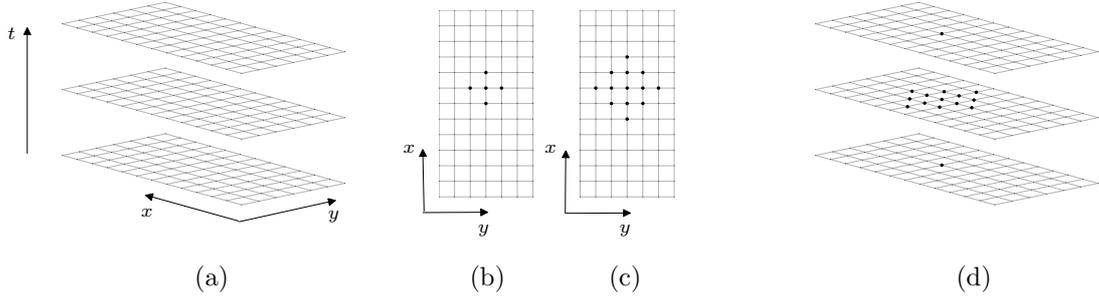


Figure 4: (a) Computational grid, defined over Cartesian coordinates x and y , and for time t . (b) Computational footprint of the five-point Laplacian operator, and (c) when applied twice. (d) Computational footprint for the scheme (15).

The other operation which must be approximated is the Laplacian ∇^2 , as defined in (7). Here is the simplest (five-point) choice:

$$\delta_{\nabla^2} u_{l,m}^n = \frac{1}{X^2} (u_{l+1,m}^n + u_{l-1,m}^n + u_{l,m+1}^n + u_{l,m-1}^n - 4u_{l,m}^n) \cong \nabla^2 u(x = lX, y = mX, t = nT) \quad (14)$$

See Figure 4(b). System (11) requires a double application of this operator—see Figure 4(c). Near the edges of the domain, this operation appears to require values which lie beyond the edges of the grid—these may be set by applying the appropriate boundary conditions. For a plate reverberation unit, these will be of free type, but in the Matlab example which follows, for simplicity, they have been set to be of simply supported type (i.e., the plate is constrained at its edges, but able to pivot). A complete treatment of numerical boundary conditions is beyond the scope of the present chapter—see [5] for more information.

A complete scheme for (6) then follows as

$$\delta_{tt} u_{l,m}^n = -\kappa^2 \delta_{\nabla^2} \delta_{\nabla^2} u_{l,m}^n - 2\sigma \delta_t u_{l,m}^n + \frac{1}{\rho H X^2} \delta_{l_i, m_i} f^n \quad (15)$$

This scheme relies on a discrete delta function δ_{l_i, m_i} , picking out a grid location corresponding to $x = x_i$, and $y = y_i$ (perhaps through truncation) and a sampled input signal f^n . The scheme operates over three time levels, or steps, as shown in (d), and is also explicit—values at the unknown time step $n + 1$ may be written in terms of previously computed values (after expanding out the operator notation above) as

$$\begin{aligned} (1 + \sigma T) u_{l,m}^{n+1} &= 2u_{l,m}^n - (1 - \sigma T) u_{l,m}^{n-1} - \mu^2 (u_{l+2,m}^n + u_{l-2,m}^n + u_{l,m+2}^n + u_{l,m-2}^n) \\ &\quad - 2\mu^2 (u_{l+1,m+1}^n + u_{l+1,m-1}^n + u_{l-1,m+1}^n + u_{l-1,m-1}^n) \\ &\quad + 8\mu^2 (u_{l+1,m}^n + u_{l-1,m}^n + u_{l,m+1}^n + u_{l,m-1}^n) - 20\mu^2 u_{l,m}^n + \frac{T^2}{\rho H X^2} \delta_{l_i, m_i} f^n \end{aligned} \quad (16)$$

where $\mu = \kappa T / X^2$ is a numerical parameter for the scheme. It may be shown that for stability, one must choose

$$\mu \leq 1/4 \quad (17)$$

and in fact, in order to reduce artifacts due to numerical dispersion (i.e., additional unwanted dispersive behaviour induced by the scheme itself, on top of that inherent to the plate itself), this

bound should be satisfied as near to equality as possible. In audio applications, this allows the grid spacing X to be chosen in terms of the sample rate. See [5] for much more on the properties of this scheme.

In implementation, it can be useful to rewrite this scheme in a vector-matrix form. A grid function $u_{l,m}^n$, representing the values of a grid function over a rectangular region, may be rewritten as a column vector \mathbf{u}^n , consisting of consecutive “stacked” columns of values over the grid, at time step n . Because the scheme as a whole is linear, and explicit, it must then be possible to rewrite it as the two-step recursion:

$$\mathbf{u}^{n+1} = \mathbf{B}\mathbf{u}^n - \mathbf{C}\mathbf{u}^{n-1} + \mathbf{r}f^n \quad (18)$$

Here, the effects of loss and the spatial difference operators have been consolidated in the (extremely sparse) matrices \mathbf{B} and \mathbf{C} , and \mathbf{r} is a column vector which chooses an input location. Output may be drawn from the scheme at a location $x = x_o, y = y_o$ by simply reading u_{l_o, m_o}^n at each time step. In vector form, the output y^n may be written as

$$y^n = \mathbf{q}^T \mathbf{u}^n \quad (19)$$

for some vector \mathbf{q} , which, in the simplest case, will consist of a single value at the readout location.

Below is a very simple Matlab implementation of the plate reverberation algorithm mentioned above—it generates a reverberant output, for given plate, sample rate and excitation and readout points. It’s not particularly fast (at least in Matlab)—but not extremely slow either!

```
clear all;

% global parameters

[f, SR] = wavread('bassoon.wav'); % read input soundfile and sample rate
rho = 7850; % mass density (kg/m^3)
E = 2e11; % Young's modulus (Pa)
nu = 0.3; % Poisson's ratio (nondimensional)
H = 0.0005; % thickness (m)
Lx = 1; % plate side length---x (m)
Ly = 2; % plate side length---y (m)
T60 = 8; % 60 dB decay time (s)
TF = 1; % extra duration of simulation (s)
ep = [0.5 0.4]; % center location ([0-1,0-1]) nondimensional
rp = [0.3 0.7]; % position of readout([0-1,0-1]) nondimensional

% derived parameters

K_over_A = sqrt(E*H^2/(12*rho*(1-nu^2)))/(Lx*Ly); % stiffness parameter/area
epsilon = Lx/Ly; % domain aspect ratio
T = 1/SR; % time step
NF = floor(SR*TF)+size(f,1); % total duration of simulation (samples)
sigma = 6*log(10)/T60; % loss parameter

% stability condition/scheme parameters

X = 2*sqrt(K_over_A*T); % find grid spacing, from stability condition
Nx = floor(sqrt(epsilon)/X); % number of x-subdivisions of spatial domain
Ny = floor(1/(sqrt(epsilon)*X)); % number of y-subdivisions of spatial domain
```

```

X = sqrt(epsilon)/Nx;           % reset grid spacing
ss = (Nx-1)*(Ny-1);           % total grid size

% generate scheme matrices

Dxx = sparse(toeplitz([-2;1;zeros(Nx-3,1)]));
Dyy = sparse(toeplitz([-2;1;zeros(Ny-3,1)]));
D = kron(speye(Nx-1), Dyy)+kron(Dxx, speye(Ny-1)); DD = D*D/X^4;
B = sparse((2*speye(ss)-K_over_A^2*T^2*DD)/(1+sigma*T));
C = ((1-sigma*T)/(1+sigma*T))*speye(ss);

% generate I/O vectors

rp_index = (Ny-1)*floor(rp(1)*Nx)+floor(rp(2)*Ny);
ep_index = (Ny-1)*floor(ep(1)*Nx)+floor(ep(2)*Ny);
r = sparse(ss,1); r(ep_index) = T^2/(X^2*rho*H);
q = sparse(ss,1); q(rp_index) = 1;

% initialize state variables and input/output

u = zeros(ss,1); u2 = u; u1 = u;
f = [f;zeros(NF-size(f,1),1)];
out = zeros(NF,1);

% main loop

for n=1:NF
    u = B*u1-C*u2+r*f(n); % difference scheme calculation
    out(n) = q'*u; % output
    u2 = u1; u1 = u; % shift data
end

% play input and output

soundsc(f,SR); pause(2); soundsc(out,SR);

```

This example is written to be compact and readable—and many more realistic features which have been omitted could be added in. Among these are:

- Frequency-dependent damping—at the moment the T_{60} is set to be uniform at all frequencies. This could be done directly in the PDE model above, or possibly through introducing a model of a damping plate.
- Boundary conditions are set to be simply supported—allowing the easy generation of difference matrices. These should ideally be set to be free, or going further, to incorporate the effect of point supports.
- Only a single output is generated—multiple outputs (such as stereo, but perhaps more in a virtual environment) should be allowed. Notice that multiple outputs may be generated essentially “for free,” as the entire state (\mathbf{u}) is available.
- Output displacement is taken; in real plate reverberation units, it is rather an acceleration—acceleration can be derived from displacement easily through a double time difference operation.

- Post-processing (analog electronic) and the behaviour of the pickup are not modelled.

Information on all these extensions appears in various publications, including those mentioned at the beginning of this section.

1.3 Springs and Spring Reverberation

The helical spring under low tension has long been used as an artificial reverberation device [14, 29, 25]. In most units, the spring is driven through an electromagnetic coupling at one end, and readout is taken via a pickup at the opposite end. Spring reverbs come in a variety of sizes and configurations, sometimes involving multiple springs. A typical set-up is shown in Figure 5(a).

The physics of helical structures is far more involved than that of the flat plate—see [28, 11] for typical models. The important point is that the various types of motion (transverse to the spring, in both polarizations, and longitudinal) are strongly coupled in a way that they are not for a straight wire. A complex combination of relatively coherent wave propagation (leading to echoes) and highly dispersive propagation is present, leading to a characteristic sound which resembles both real room responses, and plate reverbs.

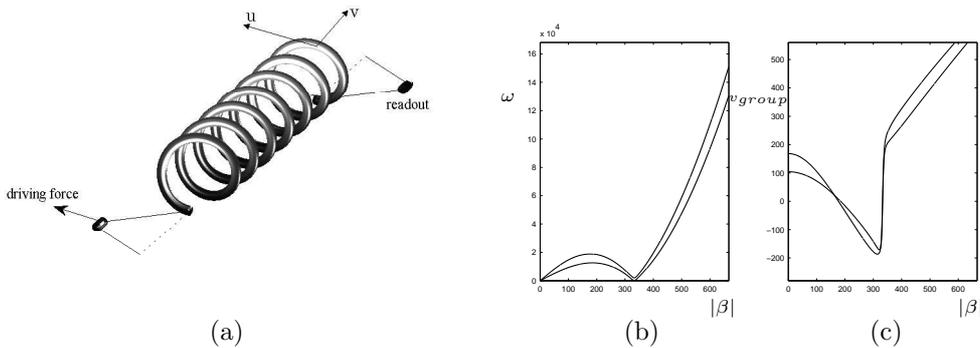


Figure 5: (a) A typical spring reverberation configuration, for a spring driven at one end, and with readout taken from the other; longitudinal and transverse motion of the spring are indicated by u and v . (b) Dispersion relations $\omega(\beta)$, and (c) group velocities $v_{group}(\beta)$.

One can garner much useful information from the dispersion relations for a spring, as shown in Figure 5(b). For springs of dimensions and composition as found in reverberation devices, there are two relations which lie within the audio range—both correspond to mixtures (wavenumber-dependent!) of longitudinal and transverse motion. Both curves exhibit a “hump” in the low wavenumber range corresponding to a cutoff—for most spring reverb units, this occurs in the range between 2 kHz and 5 kHz. An examination of the group velocities, as shown in Figure 5(c) allows for some insight with regard to the behaviour of spring responses, an example of which is shown at top in Figure 6. In the range of low wavenumbers, the group velocities approach a constant value, and thus dispersion is low at low frequencies, and strong echoes are present. But as the wavenumber increases towards the cutoff, the group velocity decreases, and thus the frequency components of the echoes are distorted increasingly (slowed) near the cutoff. Above the cutoff, the group velocity is much faster, and leads to distorted echoes which recur at a higher rate. Above the cutoff, the response is very similar to transverse motion for a straight bar.

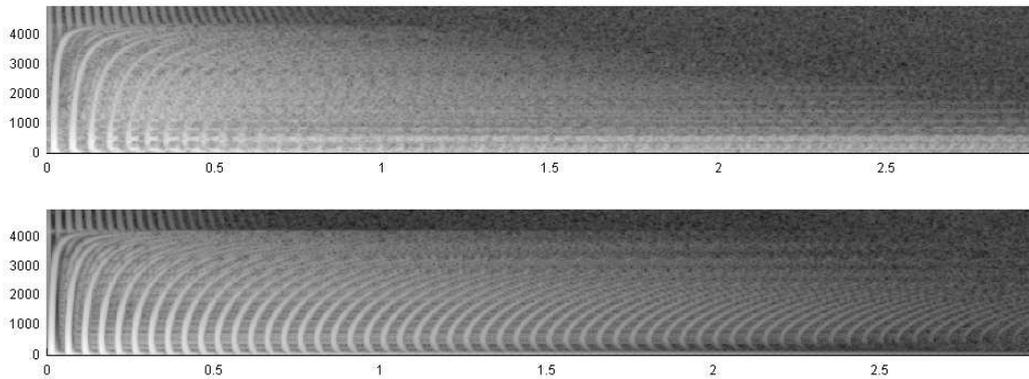


Figure 6: *Top, spectrogram of a measured spring reverberation response, and bottom, that of a synthetic response produced using a finite difference model.*

1.3.1 Emulation Techniques

As in the case of plate reverberation, there is a variety of techniques available—and as of 2010, physical modeling of springs is still an open research problem. The most rigorous approach, given a partial differential equation model, is again a time-stepping method such as finite differences, which has been used by this author [5, 7], as well as Parker [21] in order to generate synthetic responses. See Figure 6(b) for a comparison between a measured response and synthetic output. This approach turns out to be rather expensive, and given that the system is essentially 1D (i.e., for a thin spring, the equations of motion may be written with respect to a single coordinate running along the spring helix), structures based on delay lines and allpass networks are a possibility, and an excellent compromise between physical and perceptual modeling. See, e.g., [1].

The difficulties here lie in associating such a model with an underlying model in a definitive way. Another approach, as yet unexplored, could be based on the use of a modal decomposition. In other words, given appropriate boundary conditions, one may solve for the modes of vibration of the helix. This is the approach taken to the analysis of helical structures in mainstream applications—see, e.g., [18].

References

- [1] J. Abel, D. Berners, S. Costello, and J. O. Smith III. Spring reverb emulation using dispersive allpass filters in a waveguide structure. Presented at the 121st Audio Engineering Society Convention, San Francisco, California, October, 2006. Preprint 6954.
- [2] J. Allen and D. Berkley. Image method for efficiently simulating small-room acoustics. *Journal of the Acoustical Society of America*, 65(4):943–950, 1979.
- [3] K. Arcas. *Simulation numérique d’un réverbérateur à plaque*. PhD thesis, Ecole Nationale Supérieure de Techniques Avancées, Palaiseau, France, 2008.
- [4] S. Bilbao. A digital plate reverberation algorithm. *Journal of the Audio Engineering Society*, 55(3):135–144, 2007.

- [5] S. Bilbao. *Numerical Sound Synthesis: Finite Difference Schemes and Simulation in Musical Acoustics*. John Wiley and Sons, Chichester, UK, 2009.
- [6] S. Bilbao, K. Arcas, and A. Chaigne. A physical model of plate reverberation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 165–168, Toulouse, France, 2006.
- [7] S. Bilbao and J. Parker. A virtual model of spring reverberation. In press, *IEEE Transactions on Audio Speech and Language Processing*, 2010.
- [8] D. Botteldooren. Finite-difference time-domain simulation of low-frequency room acoustic problems. *Journal of the Acoustical Society of America*, 98(6):3302–3308, 1995.
- [9] A. Chaigne and C. Lambourg. Time-domain simulation of damped impacted plates. I Theory and experiments. *Journal of the Acoustical Society of America*, 109(4):1422–1432, 2001.
- [10] R. Cook, editor. *Concepts and applications of finite element analysis*. John Wiley and Sons, New York, New York, fourth edition, 2002.
- [11] L. Della Pietra and S. della Valle. On the dynamic behaviour of axially excited helical springs. *Meccanica*, 17:31–43, 1982.
- [12] N. Fletcher and T. Rossing. *The Physics of Musical Instruments*. Springer-Verlag, New York, New York, 1991.
- [13] K. Graff. *Wave Motion in Elastic Solids*. Dover, New York, New York, 1975.
- [14] L. Hammond. Electrical musical instrument, February 2 1941. US Patent 2,230,836.
- [15] J.-M. Jot and A. Chaigne. Digital delay networks for designing artificial reverberators. Presented at the 90th Audio Engineering Society Convention, Paris, France, February, 1991. Preprint 3030.
- [16] W. Kuhl. The acoustical and technological properties of the reverberation plate. *E. B. U. Review*, A(49), 1958.
- [17] C. Lambourg, A. Chaigne, and D. Matignon. Time-domain simulation of damped impacted plates. II Numerical model and results. *Journal of the Acoustical Society of America*, 109(4):1433–1447, 2001.
- [18] J. Lee and D. Thompson. Dynamic stiffness formulation, free vibration, and wave motion of helical springs. *Journal of Sound and Vibration*, 239(2):297–320, 2001.
- [19] P. Morse and U. Ingard. *Theoretical Acoustics*. Princeton University Press, Princeton, New Jersey, 1968.
- [20] D. Murphy, A. Kelloniemi, J. Mullen, and S. Shelley. Acoustic modelling using the digital waveguide mesh. *IEEE Signal Processing Magazine*, 24(2):55–66, 2007.
- [21] J. Parker. Spring reverberation: A finite difference approach. Master’s thesis, University of Edinburgh, 2008.
- [22] M. Puckette. Designing multichannel reverberators. *Computer Music Journal*, 6(1):52–65, 1982.

- [23] D. Rocchesso. Maximally diffusive yet efficient feedback delay networks for artificial reverberation. *IEEE Signal Processing Letters*, 4(9):252–255, 1997.
- [24] J. O. Smith III. *Physical Audio Signal Processing*. Stanford, CA, 2004. Draft version. Available online at <http://ccrma.stanford.edu/~jos/pasp04/>.
- [25] J.D. Stack. Sound reverberating device, March 9 1948. US Patent 2,437,445.
- [26] J. Strikwerda. *Finite Difference Schemes and Partial Differential Equations*. Wadsworth and Brooks/Cole Advanced Books and Software, Pacific Grove, California, 1989.
- [27] R. Szilard. *Theory and Analysis of Plates*. Prentice Hall, Englewood Cliffs, New Jersey, 1974.
- [28] W. Wittrick. On elastic wave propagation in helical springs. *International Journal of Mechanical Sciences*, 8:25–47, 1966.
- [29] A.C. Young and P. It. Artificial reverberation unit, October 8 1963. US Patent 3,106,610.