Proc. of the 10th Int. Conference on Digital Audio Effects (DAFx-07), Bordeaux, France, September 10-15, 2007

REAL-TIME REVERB SIMULATION FOR ARBITRARY OBJECT SHAPES

Cynthia Bruyns Maxwell
University of California at Berkeley
Computer Science Department
Center for New Music and Audio Technologies
Berkeley, California USA
cbruyns@cs.berkeley.edu

ABSTRACT

We present a method for simulating reverberation in real-time using arbitrary object shapes. This method is an extension of digital plate reverberation where a dry signal is filtered through a physical model of an object vibrating in response to audio input. Using the modal synthesis method, we can simulate the vibration of many different shapes and materials in real time. Sound samples are available at the follwing website:
http://cynthia.code404.com/dafx-audio/.

1. INTRODUCTION

Historically, plate reverberation was used as a synthetic means to simulate large room acoustics. It was one of the first types of artificial reverberation used in recording [1]. Despite the unnatural sound produced as compared to large room reverberation, plates were used extensively due to their relative low cost and small size. Recently, researchers have looked for a means of digitally simulating plate reverberation to recreate this unique analog recording style [2].

Analog plate reverberation works by mounting a steel plate with tension supplied by springs at the corners where the plate is attached to a stable frame. A signal from a transducer is applied to the plate, causing it to vibrate. This vibration is then sensed elsewhere on the plate with contact microphones. A nearby absorbing pad can also be used to control the near-field radiation.

Before the prevalence of physical modeling, plate reverberation was simulated by recording impulse responses of plates made from different materials and different geometries. By convolution of the input with the impulse response of choice, the resulting audio could sound as though it was recorded through an actual plate. Although this method is very efficient, it is limited by the number of, and variations in, the recordings available.

Bilbao et al. [2] demonstrated a model for plate reverberation using a linear Kirchoff plate formulation. By using a physical model instead of convolution with impulse responses, they could modify the geometry of the plate and input/output parameters. To simulate plate vibration, the model was discretized in space and time using finite differences. One drawback of this method, however, was the large performance requirements preventing their model from running in real-time on an average digital workstation.

Using the modal synthesis method, we can compute a plate reverberation model in real-time and still allow for modifications of the plate and input/output parameters. To achieve this performance, we use the same finite element model as described in [3] and apply forces using the discrete convolution integral method as described in [4] and [5]. We implement this reverberation as an effect plug-in that takes an audio stream as the input and produces the sound of the object vibration as the output.

The main contributions of this paper are to extend the use of the modal synthesis and the discrete convolution integral for the rapid simulation of the motion of objects in response to arbitrary loading profiles. We give examples of using this technique for the deformation of linear shell models of simple and complex shapes in a real-time synthesis environment.

2. METHODS

To briefly review the steps in obtaining a resonator bank from a arbitrary geometry we begin by discretizing the Mindlin/Reissner thin plate equations as found in [6] and [7]. This plate model differs from Kirchoff plate theory by adding the effects of shear deformation across the plate thickness. This additional motion allows for modeling thin and thick plates. We also extend the plate model to include membrane forces essentially creating a shell model from planar elements as described in [8]. The planar shell elements can be made of quadrilateral or triangular patches.

This element allows for five degrees-of-freedom at each vertex as demonstrated in Figure 1. In-plane displacement is captured by the degrees-of-freedom, u and v. For out-of-plane motion, bending is represented by adding the rotational degrees-of-freedom, $\theta_x = \frac{dw}{dx}$ and $\theta_y = \frac{dw}{dy}$, about the x and y axes, as well as an out-of-plane displacement, w. This formulation leads to the following element stiffness representation that accounts for membrane K_m, shear K_s and bending stiffness K_b (Equation 1).

$$k^e = \int_{\Omega_e} B_b^T D_b B_b d\Omega + \int_{\Omega_e} B_s^T D_s B_s d\Omega + \int_{\Omega_e} B_m^T D_m B_m d\Omega \quad (1)$$

where D represents the constitutive matrices for each stress condition, N represents the interpolating functions and B represents the operator applied on these functions. The exact entries in N and B depend on the number of nodes per element and the order of the interpolating polynomials used.

Similarly, the element mass matrix is represented as:

$$m^e = \int_{\Omega_e} N^T N d\Omega \quad (2)$$

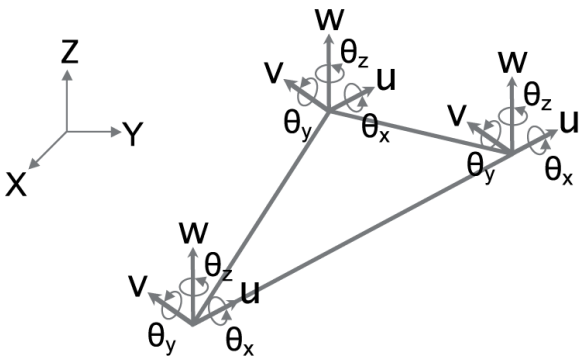

Figure 1: Normal, shear and bending forces and moments.

A more detailed discussion of the exact entries in these matrices is given in [6] and [7].

We sum the element matrices, k^e and m^e together to form an overall algebraic representation of the system. The result is the canonical representation for the second order partial differential equation of motion:

$$M\ddot{u} + D\dot{u} + Ku \quad = \quad F(t) \tag{3}$$

where M is the matrix representing the distribution of mass in the system, D is a measure of damping, K is the stiffness matrix and $F(t)$ is the force applied to the object over time. This equation expresses the balance of forces generated by the acceleration, velocity and displacement of the object. In this form, the system of equations for each degree-of-freedom (DOF) are coupled and thus the solution involves manipulation of these large system matrices. Alternatively, modal analysis seeks to decouple this system into single DOF oscillators. For a detailed description of the modal synthesis method as well as a comparison with other physical modeling techniques see [3] and references therein.

Without damping, the procedure for uncoupling these equations is straightforward using the general eigenvalue decomposition $Kx = \lambda Mx$. However, with damping, decoupling these equations requires some assumptions to be made [9].

In many finite element representations, there is no straightforward method of generating the damping matrix D. "A major reason for this is that, in contrast with inertia and stiffness forces, the physics behind the damping forces is in general not clear. As a consequence, modelling of damping from the first principles is difficult, if not impossible, for real-life engineering structures. The common approach is to use the proportional damping model, where it is assumed that the damping matrix is proportional to mass and stiffness matrices [10]." This proportionality is represented as:

$$D = \alpha_1 M + \alpha_2 K \tag{4}$$

where $\alpha_1$and α_2 are real scalars. This damping model is also known as Rayleigh damping or classical damping. Modes of classically damped systems preserve the simplicity of the real normal modes as in the undamped case.

The main limitation of the proportional damping approximation comes from the fact that the variation of damping factors with respect to vibration frequency cannot be modelled accurately by using this approach [10]. Research into the error introduced by assuming proportional damping is ongoing and current results seem to suggest that there may never be one static assumption that accurately diagonalizes a coupled damped system [11]. For now we use the proportional damping model for its simplicity.

Substituting back into Equation 3, we have:

$$M(\ddot{u} + \alpha_1 \dot{u}) + K(\alpha_2 \dot{u} + u) \quad = \quad F(t) \tag{5}$$

We assume a particular solution of the form:

$$u \quad = \quad Zv \tag{6}$$

where Z is the matrix of eigenvectors that diagonalizes the system. Substituting back into Equation 3 and pre-multiplying by Z^T we have:

$$Z^T MZ(\ddot{v} + \alpha_1 \dot{v}) + Z^T KZ(\alpha_2 \dot{v} + v) \quad = \quad Z^T F(t) \tag{7}$$

This equation simplifies to:

$$\ddot{v} + (\alpha_1 + \alpha_2 \omega^2)\dot{v} + \omega^2 v \quad = \quad Z^T F(t) \tag{8}$$

Equation 8 represents the uncoupled bank of resonators oscillating at the natural frequencies determined by the eigenvalues of the system. By solving each equation for u we can represent the response at any location on the object at any time. Therefore, to solve for the motion at the pickup locations we weight the contributions of the various modes on the spatial positions of interest.

To apply the input to the system we know that in general, any force-response history can be represented as a succession of infinitesimal impulses. We also know that the response of the system to such a force profile can be built up from the response to each infinitesimal impulse individually [12]. Therefore we can represent a discrete time excitation as a combination of unit step functions:

$$f(n) \quad = \quad \sum_{k=0}^{\infty} f(k)\delta(n-k) \tag{9}$$

where $f(n)$ is the applied force and δ is the Dirac Delta function. The response to this discrete-time excitation is then:

$$x(n) \quad = \quad \sum_{k=0}^{\infty} f(k)g(n-k) = \sum_{k=0}^{n} f(k)g(n-k) \tag{10}$$

where $g(n)$ represents the discrete-time impulse response of a linear time invariant system to the unit impulse $\delta(n)$. Equation 10 essentially approximates the response $x(n)$ in the form of a convolution sum, the discrete counterpart of the convolution integral. The convolution or Duhamel's integral is a means of finding solutions to linear, nonhomogeneous, second order, ordinary differential equations with constant coefficients. The nonhomogeneous part of the equations comes from the forcing function and depending on the complexity of this term, the integral may or may not have a closed-form solution [13]. The convolution integral then, is a means of finding the solution to the original system by summing the individual impulse responses.

Instead of evaluating the non-recursive convolution sum as in Equation 10, DiFilippo and Pai [4] use a different technique to solve for the response to non-harmonic excitation. They use a recursive method for approximating the response at the current time

step by scaling the value at the previous time step and adding that to the response to the new impulse. Thus, for each resonator:

$$x_n(0) = a_n f(0) \quad (11)$$

$$x_n(k) = e^{i\frac{\Omega_n}{F_s}} x_n(k-1) + a_n f(k) \quad (12)$$

where Ω_n is the natural frequency and F_s is the sampling frequency and:

$$\Omega_n = \omega_n + i d_n \quad (13)$$

$$d_n = \frac{\omega_n}{2\pi}\pi \tan(\phi) \quad (14)$$

where ϕ is an internal damping factor. Thus the overall sound generated at time t is:

$$s(t) = Re\left(\sum x_n(k)\right) \quad (15)$$

In this way, the incoming audio signal represents an arbitrary discretized force profile applied to the resonators interpolating the input position.

2.1. Software Implementation

The rendering algorithm works by first performing the modal decomposition and then filtering the incoming audio through the resonator bank produced. The time to compute the modal decomposition depends on the number of modes required and the number of elements in the finite element model. We achieve real-time performance by first computing the decomposition, which can take several seconds. We then evaluate Equation 15 for each audio sample. $s(t)$ is computed in a parallel (or vectorized) fashion as each mode is linearly independent.

The user interface for the plug-in loads an object geometry and displays the surface for specifying the input and pickup locations. The left portion of the user-interface allows for modification of the material parameters, object scale and plate thickness. These parameters are adjusted before modal decomposition. The right portion of the user interface has controls for the audio rendering parameters such as the frequency scaling and resonator decay. These parameters do not require reanalysis, instead they are applied to the bank of resonators as audio is rendered. There is also control for the number of resonators used for simulation. Using more resonators creates a fuller tone but requires more computation.

3. RESULTS

The following examples were computed using one processor of a dual 2.5GHz Power PC G5. The plug-ins were hosted using Apple Inc.'s AULab application and audio input was streamed using the built-in AUFilePlayer component. In each example, the points in green represent the input position and the points in red represent the pickup locations.

For the first example, we load a simple plate model as shown in Figure 2 (top). The model has 100 elements, and the time to compute the decomposition into 485 modes was 0.65 seconds. Figure 3 (top) shows the waveform and Figure 4 (top) shows the spectrogram of the incoming signal applied to the plate. Figure 3 (middle) shows the resulting waveform and Figure 4 (middle) shows the frequency profile generated for the left channel. Immediately, one can see the effect of reverberation on the resulting audio. Where there were once discrete peaks, the audio now blends together. Moreover, the frequency spectrum is low-pass filtered through the number of modes used in the synthesis algorithm.

We can use also this method on novel shapes and explore the effect on the resulting audio. Figure 2 (bottom) shows a more complex shell surface with arbitrary input and output locations. This model had 500 elements and took 24.5 seconds to compute all 1548 modes. Using the same input profile as Figure 3 (top), we can compare the resulting waveform and frequency spectra when rendering through this new geometry (Figure 3 (bottom), Figure 4 (bottom)).

Notice that in Figure 4, the output through the resonator bank has less of the high frequency components than the original signal. This is to be expected as the resonant frequencies of the set of resonators and user-selected damping values will not exactly match the original signal. In some sense, the original signal acts to imprint its frequency spectrum on the resonator bank roughly but need not exactly match.

For both of these examples, simulating object vibration using 20 modes consumed around 1.4% CPU; 100 modes consumed roughly 3%; 1000 modes consumed 22%; and 3000 modes used 84% for two channels of stereo processing. These results suggest that for up to 1000 modes, the method performs well. For the 3000 or so resonators needed for non-metalic, perceptually realistic sounding reverberation [14], the real-time CPU demand is considerable when using only one processor.

4. DISCUSSION

In this investigation we have demonstrated a method for simulating reverberation using the modal synthesis method. By using a physical model of a vibrating object, we are free to use any arbitrary geometry and material.

For plates with very thin cross-sections, it is likely that large applied forces will cause large plate deformation. When this happens, linear models can no longer be used. As a result, techniques such as linear modal superposition must be abandoned for nonlinear modal analysis or nonlinear models and numerical integration. Other researchers are actively investigating the importance of these nonlinearities in plate reverberation models [15].

5. ACKNOWLEDGEMENTS

Thanks to John Lazzaro, Antoine Chaigne and Julius O. Smith III for useful discussions on plate reverberators and to Justin Maxwell for the user interface layout. Also thanks to C.J. Slyfield for editing the complex shape model.

6. REFERENCES

[1] D. C. Myers, "Audio reverberator," Tech. Rep., U.S. Patent 4,653,101, March 24, 1984.

[2] A. Bilbao, K. Arcas, and A. Chaigne, "A physical model for plate reverberation," *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 5, pp. V–165–V–168, 2006.

[3] C. Bruyns, "Modal synthesis for arbitrarily shaped objects," *Computer Music Journal*, vol. 30, no. 3, pp. 22–37, 2006.

[4] D. DiFilippo and D.K. Pai, "The AHI: An audio haptic interface for contact interactions," in *Proceedings of ACM UIST 2000*, 2000.

[5] K. van den Doel, P. G. Kry, and D.K. Pai, "Physically-based sound effects for interactive simulation and animation," in *Proceedings of SIGGRAPH 2001*, 2001.

[6] S. S. Rao, *The Finite Element Method in Engineering Second Edition*, Pergamon Press, Oxford, 1989.

[7] Y. W. Kwon and H. Bang, *The finite element method using MATLAB*, CRC Press, 2000.

[8] O. C. Zienkiewicz and R. L. Taylor, *Finite Element Method: Volume 2, Solid Mechanics*, Butterworth-Heinemann, Burlington, MA, 5th edition, 2000.

[9] G. B. Warburton and S. R. Soni, "Errors in response calculations for non-classically damped structures," *Earthquake Engineering & Structural Dynamics*, vol. 5, no. 4, pp. 365 – 376, 1977.

[10] S. Adhikari, "Damping modelling and identification using generalized proportional damping," in *Proceedings of the 23rd International Modal Analysis Conference (IMAC-XXIII), Orlando, Florida, USA*, Feb 2005.

[11] F. Ma and T.K. Caughey, "Analysis of linear nonconservative vibrations," *Journal of Applied Mechanics*, vol. 62, pp. 685–691, 1995.

[12] L. Meirovitch, *Principles and Techniques of Vibrations*, Prentice Hall, Upper Saddle River, New Jersey, first edition, 1999.

[13] E. Chicurel-Uziel, "Closed-form solution for response of linear systems subjected to periodic non-harmonic excitation," in *Proceedings of the Institute of Mechanical Engineers, Part K*, 2000.

[14] M. Karjalainen and H. Järveläinen, "More about this reverberation science: Perceptually good late reverberation," in *In Proceedings of the AES 111th International Convention*, September 2001.

[15] A. Chaigne, "Plate reverberator: modeling, analysis and sysnthesis," CCRMA Music 420 Seminar, March 2007.

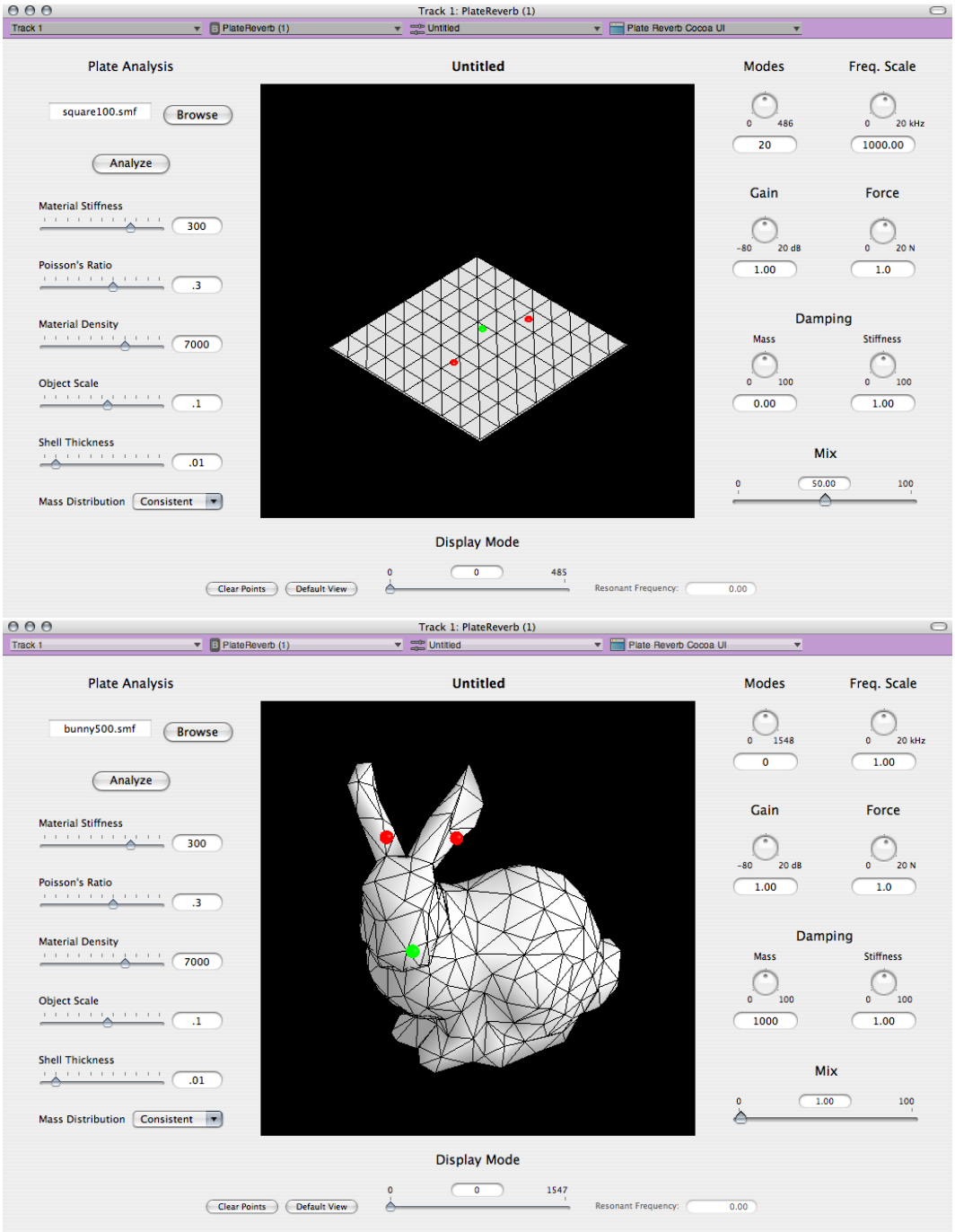

Figure 2: Top: A traditional plate reverb geometry. Bottom: A complex reverb surface.

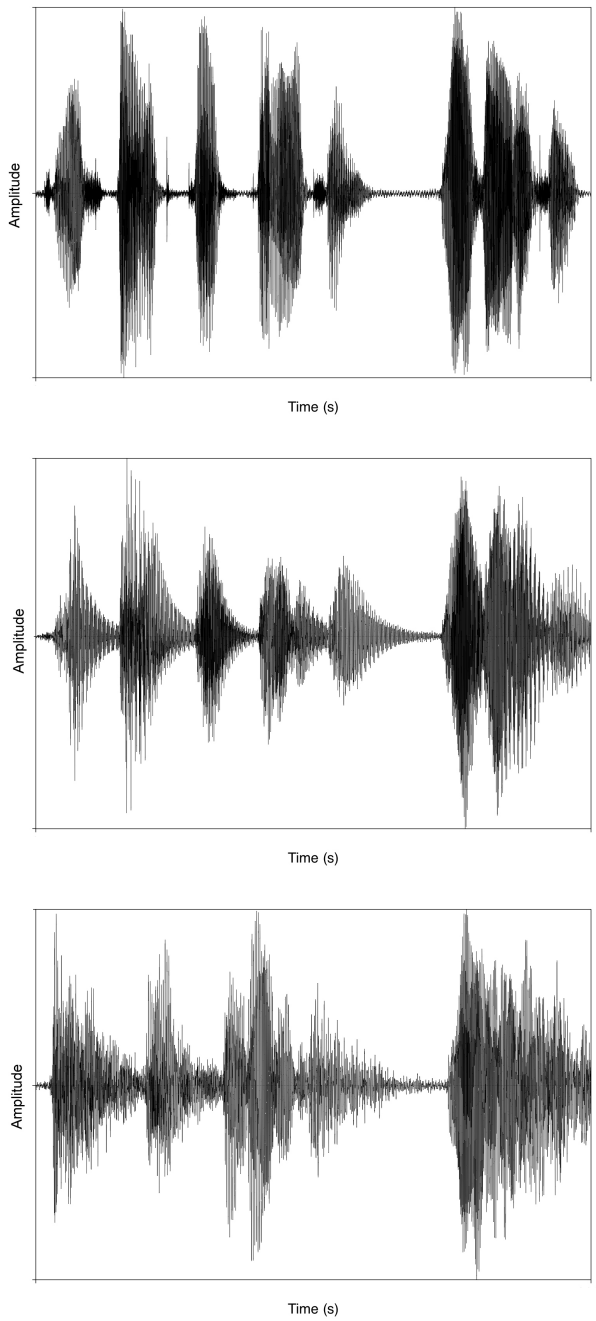

Figure 3: Top: Incoming audio signal applied to the plate. Middle: Simple plate vibration. Bottom: Complex surface vibration.

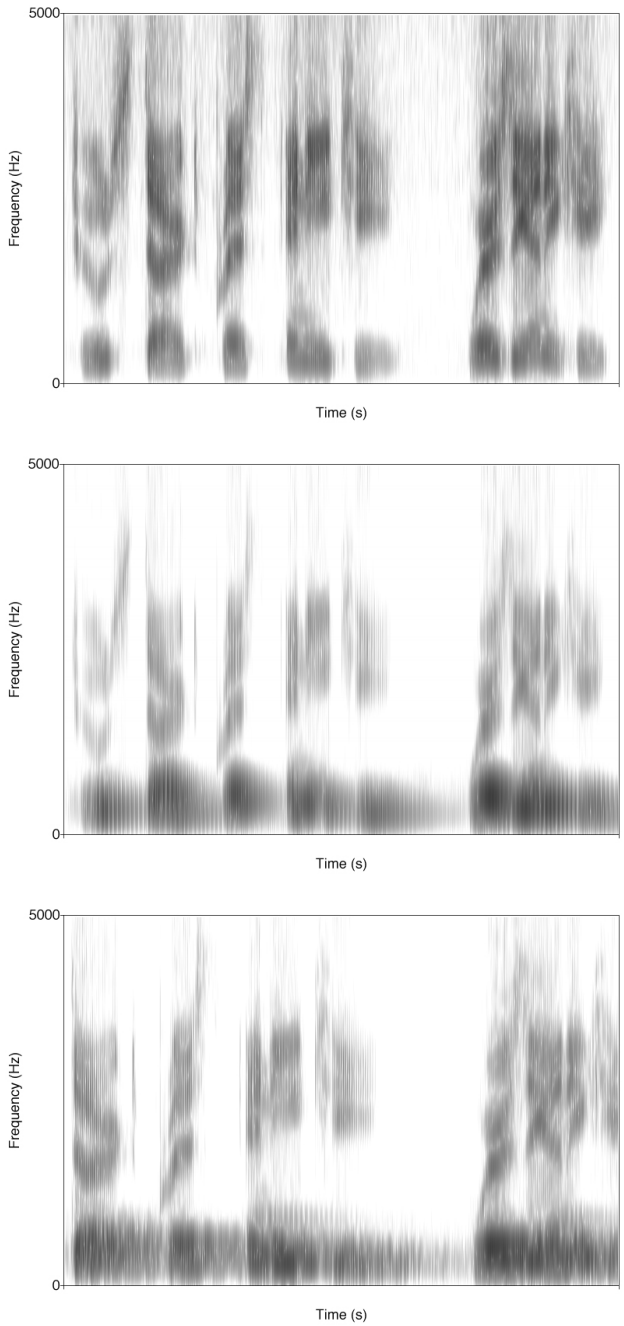

Figure 4: Top: Frequency profile of the force profile. Middle: Simple plate vibration. Bottom: Complex surface vibration.