

MODAL SYNTHESIS FOR PLATE REVERBERATION

Erin Driscoll, Michael Finley, Rob Salino

University of Rochester

ABSTRACT

Popularized in the 1960's, these steel plate apparatuses were often used to simulate reverberation in studio recordings instead of recording in physical reverberant spaces. The goal of this project is to better understand the virtualization of an effect that relies on the physical properties of a plate reverb. Rather than simply create an acceptable digital reverb based on decaying, delayed audio samples, we will need to consider these physical traits to build an algorithm that simulates the movement of a plate. Using MATLAB to create the plugin, we will allow the user to adjust these parameters and observe the correlation between those changes and the change in the reverberation effect.

Index Terms— Plate reverb simulation, audio plug-in, modal synthesis

1. INTRODUCTION

The plate reverbs that were once a staple of mid-20th century music production are now very costly, with arguably the most famous, the EMT 140, often selling for over \$5,000 USD; creating a digital version of this will help make its unique sound accessible. The concept of a plate reverb is rather simple: a transducer inputs the dry signal onto the surface of a suspended metal plate (most often steel) and two pickups are placed at specific points on the plate. This reverberation process is analogous to a speaker cone outputting acoustic waves into a reverberant space and having two separate microphones, one for the left output channel and one for the right output channel, to pick up the signal that was affected by this reverberant space. The vibrations caused by the transducer can be modeled using the Kirchoff-Love plate equation(s) that describe the flexing and movement of a vibrating plate [4]. Simulation would appear to be further complicated by the need to calculate eigenfrequencies from an infinite number of modes, but is simplified because this can be finite in our application [3].

1.1. Previous Work

Previous simulations of plate reverbs rely on finite discrete difference equations [1, 2] or simple delay line models [] that overlook the complexity of the plate reverb. A modal synthesis approach enjoys the benefits of being a high quality

model for what the plate reverb is doing while also providing flexibility for computational resource limits [5]. Furthermore, the plate reverb modal equations appreciate the parallelism that makes computations efficient in real-time. Only fairly recently has modal synthesis become possible in real-time [3]. This approach provided the necessary numerical equations for developing our own modal synthesis method.

2. METHODS AND IMPLEMENTATION

Details for the methodology and implementation of our plate reverberation simulation are detailed in the following sections. Numerical equations used in the simulation were referenced from [3].

2.1. Modal Equations

As discussed, an approximation of the displacement of the plate can be decomposed into a finite sum of modes:

$$w = \sum_{m=1}^M q_m \Phi_m(x, y) \quad (1)$$

where w is the total displacement of the plate, M is the total number of modes considered, q_m is a weight, and $\Phi_m(x, y)$ are the spatial displacement for each mode. The functions (x, y) are calculated using the following equation.

$$\Phi_m(x, y) = \sqrt{\frac{4}{L_x L_y}} \sin\left(\frac{n\pi x}{L_x}\right) \sin\left(\frac{m\pi y}{L_y}\right) \quad (2)$$

where n, m are modal indices, and L_x and L_y are the dimensions of the plate. The Kirchoff-Love equations rely on first and second time derivatives of q_m , in addition to individual eigenfrequencies and physical properties. It is the inclusion as parameters of the physical properties of the plate which is unique to our implementation of the plate reverb simulation. The governing equation for the solution of q is

$$\ddot{q}_m + \omega_m^2 q_m + 2c_m \dot{q}_m - \frac{\Phi_m(x_p, y_p)}{\rho h} P(t) = 0 \quad (3)$$

where q derivatives are time derivatives, ω_m are eigenfrequencies, c_m are loss coefficients, ρ is the volumetric density,

h is the plate thickness, x_p, y_p are the input transducer location, and $P(t)$ is the input audio or pressure applied to the plate [3].

In our implementation, the weights q_m are calculated using a standard central finite difference approximation for first and second order derivatives, where in general for arbitrary function f and sufficiently small increment h ,

$$\ddot{f} = \frac{f(x-h) - 2\dot{f}(x) + f(x+h)}{h^2} \quad (4)$$

$$\dot{f} = \frac{f(x+h) - f(x-h)}{2h} \quad (5)$$

The eigenfrequencies ω_m can be calculated using the following equation.

$$\omega_m = \sqrt{\frac{T_0}{\rho p} \left(\frac{m^2 \pi^2}{L_x^2} + \frac{n^2 \pi^2}{L_y^2} \right) + \frac{D}{\rho p} \left(\frac{m^2 \pi^2}{L_x^2} + \frac{n^2 \pi^2}{L_y^2} \right)^2} \quad (6)$$

where T_0 is the tension, and D is the flexural rigidity. Flexural rigidity is computed using E , Young's modulus, and ν , Poisson's ratio:

$$D \equiv \frac{Eh^2}{12(1-\nu^2)} \quad (7)$$

2.2. MATLAB Implementation

The plate reverb was implemented in MATLAB as an audio-Plugin object with an audioPluginInterface. MATLAB allows users to output a variety of standalone plug-in formats for use with various digital audio workstations (DAWs) which allows for the plugin to work in many contexts. The plugin features a constructor, an initialization method, a process method, and a reset method.

The constructor calls the initialization method. The initialization method initializes necessary vectors to be passed to the process function, as well as calling a separate MATLAB function. Decoupling the initialization function allows for generalization of the plate to be used in a variety of other ways. For example, a parallelized or serialized plate reverb that sends a signal through multiple plates. This generalized function accepts the length, width, and height of the plate, the number of cents (hundredth of a hertz) to be used for quality purposes, the speed of propagation in the medium, the decay time, the input and output locations, the maximum number of modes to consider, and the sampling frequency. This function returns the necessary coefficients discussed in Sec. 2.1 which are the coefficients for Eqn. 1., time step values, the eigenfrequencies of the decomposition, and the left and right plate displacement values detailed in Eqn. 2.

The process function checks that a plate has been initialized, then initializes the output vectors for a single window of audio. The process function checks if any plate attributes

have changed in real-time, if so, the plate values are recalculated. To avoid discontinuities in audio the process function checks that any changes in the plate don't exceed a specific delta. Eigenfrequencies of the decomposition are compared to a stability condition which we found to be:

$$f_{stable} = \frac{fs}{2} \quad (8)$$

Where fs is the frequency of sampling. The stability condition ensures any individual ω_m does not have negative energy. Then the plate displacement is computed based on the input actuator location. Previous, current, and future window plate displacement and coefficient values are stored for future inputs to allow for a finer resolution plate emulation.

An example of the time-domain signal inputs and outputs can be seen in Fig. 1 showcasing the effects of a plate reverb on a simple drum sample.

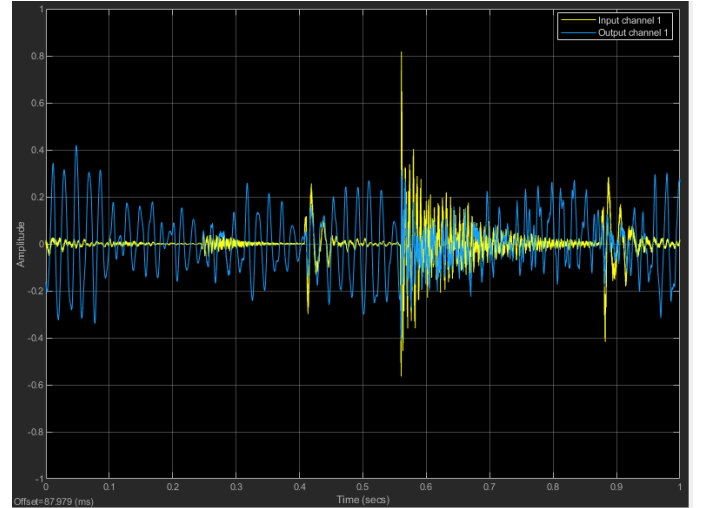


Fig. 1. Input and output of the plate reverb plug-in.

The resulting plug-in was tested in a variety of DAWs and was able to work in real-time without any audible problematic delay. An octave sub-band graph can be seen in Fig. 2.

There is an apparent difference in the high frequency response between a small and large plate. To observe this, impulse responses were recorded for two different sized plates with actuators and pickups at the locations shown in Table 1.

Table 1. Large and small-plate specifications.

	Small Plate (1x0.5m)	Large Plate (2.5x1.75m)
Actuator	[0.4, 0.2075] m	[1, 0.072625] m
Left Pickup	[0.1, 0.225] m	[0.25, 0.7875] m
Right Pickup	[0.84, 0.225] m	[2.1, 0.7875] m

The actuator and pickup positions are placed in relative positions on these differently-sized plates, rather than placed

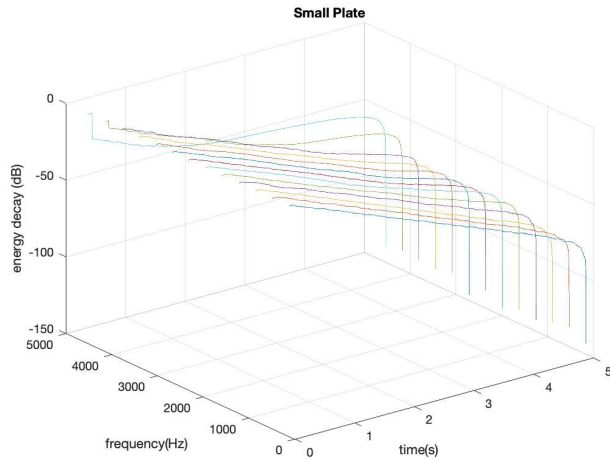


Fig. 2. Octave sub-band impulse response of small-plate.

in a fixed, absolute location. Two VST plugins were created using these separate plate size settings, and an impulse response passing through both of these plates with dry signal at 0% and wet signal at 100% was recorded and analyzed. As expected, we observed high frequency decay is faster on the larger plate than on the smaller plate.

3. CONCLUSION

One goal of the plug-in was to be able to manually select input and output locations of the plate. MATLAB does not allow for mouse-click events as an input at the time of this writing. As a result, the plate diagram showcasing input and output locations as an interactive control for the reverb unit was omitted. However, if the plugin is used in the MATLAB environment (for example, Audio Test Bench as part of MATLAB’s Audio Toolbox), a display of input and output locations can be visualized in a separate figure, Fig. 4.

3.1. Future Work

More work would need to be done in order to make this visualization more efficient, so as to update in real time with no lag, and to find a way to have this feature included in the plugin outside of the MATLAB Coder environment. A visual of the audio plug-in graphical user interface (GUI) can be seen in Fig. 3.

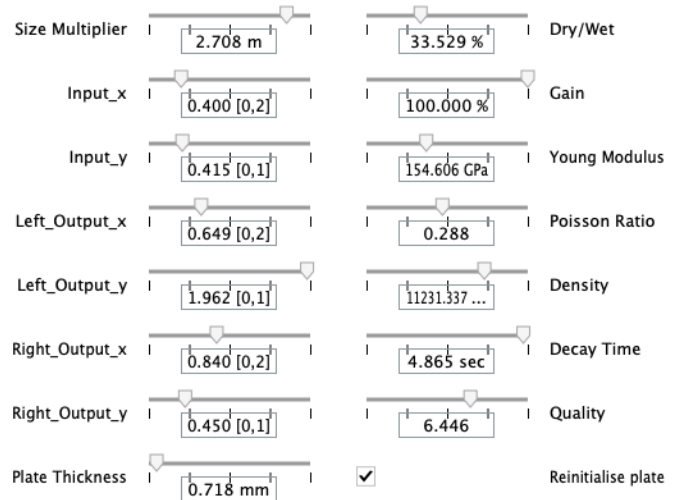
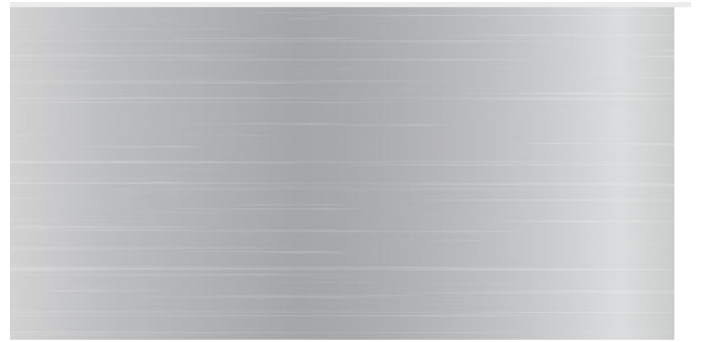


Fig. 3. GUI of the plate reverb plug-in.

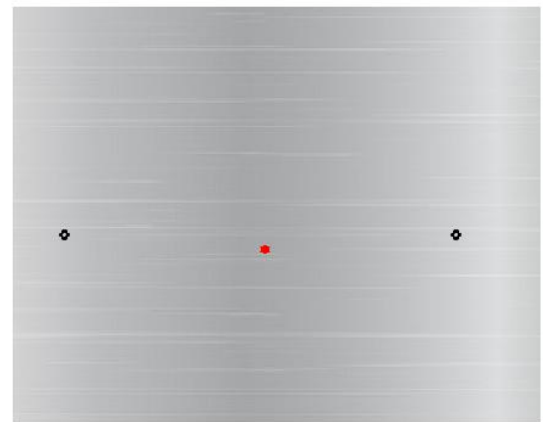


Fig. 4. Default input and output pick-up locations.

4. REFERENCES

- [1] Stefan Bilbao. A digital plate reverberation algorithm. *J. Audio Eng. Soc.*, 55(3):135–144, 2007.
- [2] Stefan Bilbao, Kevin Arcas, and Antoine Chaigne. A physical model for plate reverberation. volume 5, pages V – V, 06 2006.
- [3] Michele Ducceschi and Craig Webb. Plate reverberation: Towards the development of a real-time physical model for the working musician. 09 2016.
- [4] Maria Radwanska, Stankiewicz Anna, Adam Wosatko, and Jerzy Pamin. Plate and shell structures: Selected analytical and finite element solutions. 2005.
- [5] Silvin Willemsen, Stefania Serafin, and Jesper Jensen. Virtual analog simulation and extensions of plate reverberation. 2017.