Virtual Acoustic Guitar Lab

Julius O. Smith III and Nelson Lee

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

June 5, 2008

Abstract

In this lab, you will write C++ classes to simulate an acoustic guitar. This lab extends the REALSIMPLE electric guitar lab[1] to the case of an acoustic guitar.

1 Factoring Modes from an Acoustic Guitar's Body Response

An acoustic guitar simulation needs to take into account the resonances of the body, since they impart a characteristic "color" to the plucked tone. In a commuted waveguide synthesis model,[2] the impulse response of the body can simply be fed in as the excitation to the guitar string. Sometimes it becomes impractical to use the full impulse response due to memory constraints. One technique to shorten the impulse response is to factor out the least damped resonances in the frequency response and implement them separately as second-order filters in series.

1.1 Finding the Least-damped Modes

(10 pts) Find the frequencies of the three least-damped modes in the guitar body impulse response (gtrbody.wav[3]). A quick way to find these is to take an FFT of a portion of the response where most of the modes have decayed considerably (e.g., after 125 ms) and only a few strong modes are left. To get a more accurate estimate of the peak frequency, you can zero-pad and/or use parabolic fitting on a dB scale. This is documented at `http://ccrma.stanford.edu/~jos/parshl/Peak_Detection_Steps_3.html`.

1.2 Removing the Modes

(20 pts) Using the code at `http://ccrma.stanford.edu/~jos/waveguide/Matlab_Code_Inverse_Filtering.html` as a starting point, successively remove the lowest mode (the "Helmholtz mode") and the other two modes.

---

*Work supported by the Wallenberg Global Learning Network

[1] http://ccrma.stanford.edu/~jos/realsimple/electric_guitar/electric_guitar.pdf

[2] `http://ccrma.stanford.edu/~jos/pasp/Commuted_Synthesis_Strings.html`

[3] http://ccrma.stanford.edu/~jos/realsimple/acoustic_guitar/gtrbody.wav

Make sure that the mode removals do not affect the spectral content of the rest of the frequency response. You can estimate the peak bandwidths with any method, or just use 2, 9, and 9 Hz and manually tweak if needed.

Plot the time and frequency responses of the original body response and the three mode-extracted responses. Include a close-up plot of the frequency region where the modes were removed for the 4 cases.

1.3 Relating Body Response Length to Modes

(5 pts) Which of the mode-extracted responses has the shortest effective length? Approximately how long is the response (in ms)?

1.4 Restoring the Extracted Modes Inexpensively

(10 pts) Restore the mode-extracted guitar body impulse responses by using biquads as needed to model the missing modes. Compare these against the original response. Is there an audible difference between the four versions? (You can play sound directly from Matlab using `sound` or `soundsc`.) Plot the amplitude error between the original and mode-restored responses.

1.5 Usefulness of the Decomposition

(5 pts) How might this decomposition of the guitar body impulse response allow for additional controls or effects for the commuted acoustic guitar model? Give a few examples.

2 Building an Acoustic Guitar Synthesizer in STK

(20 pts) The manner of termination for the guitar string at the bridge results in slightly different effective lengths for horizontal and vertical transverse string vibration. In the horizontal plane, the string terminates at the bridge, but it terminates slightly beyond that in the vertical plane of vibration. The non-rigid, yielding bridge also allows for coupling between the two polarizations.

Turn your electric guitar model into an acoustic guitar by using a guitar body impulse response soundfile as the excitation source and implementing coupled strings as diagrammed at `http://ccrma.stanford.edu/~jos/CoupledStrings/Linearly_Coupled_Planes_Vibration.html`. Include a plucking angle control parameter. Some design restrictions are:

1. $N_v > N_h$ *and/or* $P_{vv} > P_{hh}$, where P_x denotes phase delay of filter x.
2. $H_{vh}(z) = H_{hv}(z)$, with lowpass characteristics (can be modeled after $H_{vv}(z)$)
3. $H_{hh}(z)$ should be nearly lossless (e.g., -1 for velocity waves if the nut reflection has not been commuted through)
4. $H_{vv}(z)$ should be a symmetric second-order FIR loop filter with parametric control (as used in the previous assignment)
5. $H_l(z)$ should be nearly lossless (weakly lowpass)

Produce an illustrative example of your acoustic guitar consisting of at least a few notes, and submit it with the rest of your code. Turn in any relevant program listings and document the parameter values you used to generate the example.