Music 420
Winter 2005-2006
Homework #5
Strings, Electric Guitar Synthesis
145 points
Due in one week (Feb/28/2006)

Theory Problems

1. (10 pts) In class, we derived that the *driving-point impedance* of the infinitely long ideal string, driven at an endpoint, was $R = \sqrt{K\epsilon}$. Derive now the driving-point impedance $R(e^{j\omega T}) = F(e^{j\omega T})/V(e^{j\omega T})$ of the length L *rigidly terminated* ideal string[1] at each of the termination points, i.e, at $x = L$ and $x = 0$. Express your answers in terms of R, ω, T, L, and $c = \sqrt{K/\epsilon}$. [Hint: Perform a traveling-wave decomposition on $F(z)$ and $V(z)$.]

2. Consider a basic digital waveguide model for a vibrating electric guitar string with pitch 125 Hz and length 26 inches at a sampling rate of 10 kHz. Assume displacement waves are to be simulated.

   (a) (10 pts) Describe the parameter settings and structure of the model. Provide a diagram of the digital waveguide model.

   (b) (10 pts) Draw an "efficient" form of the previous model but now with a plucked position specified to be 1 inch from the bridge and a magnetic pick-up position 3 inches from the bridge. Assume the loop/damping filter is located in the reverse-traveling wave component between the pluck position and the pick-up. See links below for more details.

   - `http://ccrma.stanford.edu/~jos/SimpleStrings/External_String_Excitation_Point.html`
   - `http://ccrma.stanford.edu/~jos/pasp/Equivalent_Forms.html`

3. (10 pts) Find the decay time, (t_{60}), at each harmonic in first part of the previous problem, assuming the loop filter has a gain given by $1/k^2$, where $k = 1, 2, 3, \ldots$ is the harmonic number.

4. In the system of the previous problem (assume the system is lossless (i.e. no loop filter)),

   (a) (5 pts) Sketch the initial delay line contents when the string is to be plucked at its midpoint at time $t = 0$.

[1] http://ccrma.stanford.edu/~jos/pasp/Rigid_Terminations.html

(b) (5 pts) Sketch the initial delay line contents when the string is to be plucked at one-fourth from an endpoint at time $t = 0$.

(c) (10 pts) For the midpoint-pluck case, sketch the delay line contents and corresponding physical string shape at time $t = 1$, 2, 3, and 4 ms.

Lab Assignments

Lab assignments are due one day after the theory portion of the homework (typically Friday afternoon at 5pm). Lab assignments are to be submitted electronically at http://coursework.stanford.edu. For each assignment, submit a single archive (`zip` or `tar.gz`) file containing all code/figures/output for the assignment. For each problem in the assignment, create a sub-directory in the archive named `hwX_pY`, where `X` is the homework number, and `Y` is the problem number, and place all code/figures/output for that problem in that sub-directory.

Starter code and soundfiles for this assignment are archived in `hw5stuff.tar.gz`.[2] To unpack, first save it into your STK `myproj` directory (typically `~/stk/stk/myproj`). Then use the following command in a terminal at the same directory level: `tar -zxf hw4stuff.tar.gz`.

1. (15 pts) Write an STK class `String.cpp` which implements a digital waveguide string model. Implement the "efficient" block diagram you derived in Prob. 2.(b). The `tick()` function should accept an input sample (for exciting the string) and return the current string output. Control parameters governing construction should include the desired pitch, the plucked position and the observing point. Other characteristics, such as string damping, can be hard-wired or brought out as controls.

   The string loop filter should be a symmetric second-order FIR filter $H_l(z) = a + bz^{-1} + az^{-2}$. The string delay line should be interpolated using allpass interpolation (e.g., using the STK class `DelayA`). If hard-wired, the string loop filter coefficients should be $b = 1/2$, $a = 1/4$.

2. (15 pts) Write an STK class `ElecGuitar.cpp` which uses a single instance `String` to simulate a string vibration. Each `ElecGuitar` string should have the following features

   (a) It excites its String with a white-noise burst which is filtered by a one-pole lowpass filter (the default pole location should be 0.5). The noise is available from an STK class called `Noise` and its burst should have a controllable duration up to at least one period in length (the default pluck time should be 50 samples). The lowpass filter should have dc gain equal to 1 (or slightly less) for all "brightness" settings. It should have a `noteOn` and `noteOff` methods to deal with the start and end of a note. See examples from other instruments available in STK.

[2] http://ccrma.stanford.edu/~jos/hw420/hw5/hw5stuff.tar.gz

(b) It should allow for pluck and magnetic pick-up positions to be specified in its corresponding `String` instance as a ratio between [0,1] of the string length from the bridge. Let the default pluck and pick-up positions be 1 and 3 inches from the bridge (respectively).

(c) It should allow for a feedback input to be added to the string excitation from external programs e.g. via `setFeedback` method.

You may use as an example, `SimpString.cpp`, given in `gtrc` directory from the `jos-overlay`. Note, however, that your `String` should do nothing except setting up the string (including pluck and pick-up positions). It has to be excited through `ElecGuitar`.

3. (5 pts) Write an STK class `Overdrive.cpp` which simulates a nonlinear soft-clip amplifier distortion, as described in *"Amplifier Distortion + Amplifier Feedback".*[3] The controls should be (at least) pre-distortion gain, post-distortion gain, and direct-signal gain. Its `tick` method takes an input and emits an output. In the main program, the input to the distortion unit should be the sum of all strings' outputs.

4. (5 pts) Implement the amplifier feedback, as described in *"Amplifier Distortion + Amplifier Feedback".*[4] except that you can feedback the final output of the distortion unit for modularity. The controls are feedback gain and feedback delay. This can be simply implemented directly in the main test program `mygtr.cpp`. In the main program, the input to the amplifier feedback unit should be the output from the distortion unit. Its output is fed back into each string in `ElecGuitar`.

5. (10 pts) Complete the test program `mygtr.cpp` that plays a simple SKINI score file containing a few notes using `ElecGuitar` just created. Compile and link using the `Makefile` given along with the sample SKINI file `test.ski`. Turn in your commented program listings as usual.

6. (20 pts) Using `test.ski`, produce examples of your electric guitar for the three different playing modes : guitar, guitar with distortion, and guitar with distortion and amplifier feedback. Concatenate these synthesized examples into a SINGLE soundfile and place a copy in your submission folder, using your last name in the soundfile name. Make sure they all sound acceptable for grading.

[3] http://ccrma.stanford.edu/~jos/SimpleStrings/Amplifier_Distortion_Amplifier.html
[4] http://ccrma.stanford.edu/~jos/SimpleStrings/Amplifier_Distortion_Amplifier.html