

Project 2 (Due Sep-26 11:59pm)

1. Envelopes:

1.1 Based on the lecture and reading material, briefly state why using envelopes is very important in sound synthesis.

1.2 We've learned to use both "pwl" and "env" for envelopes in Nyquist. Without stretching, what parameters to a "pwl" function produce the same result as `env(0.01, 0.04, 0.3, 1, 0.7, 0.6, 1) ~ 2` ? If you're synthesizing trumpet-like notes of different length by using a unit generator, which envelope function would be better?

Hint: Look at "env" in the Nyquist Reference Manual and pay attention to its behaviors under different stretch environments.

2. Sampling Theory:

2.1 A theoretical telephone system uses digital audio for voice; the highest frequency carried over the phone system is 4200Hz. What is the theoretical minimum sampling rate needed for this bandwidth?

2.2 If the theoretical phone system uses 10bit linear PCM, what would be the signal-to-noise ratio in decibels?

2.3 Express the decibel figure in 2.2 as a linear ratio of amplitudes. To learn more about decibels, see decibels below.

2.4 What is the purpose of low-pass filtering before converting a digital signal to analog?

3. Composition Programming

3.1 The following code creates a short score and plays it:

```
variable *ascore* =  
{0 1 {note pitch: c2 vel: 50}}  
{1 0.3 {note pitch: d2 vel: 70}}  
{1.2 0.3 {note pitch: f2 vel: 90}}  
{1.4 0.3 {note pitch: g2 vel: 110}}  
{1.6 0.3 {note pitch: a2 vel: 127}}  
{1.8 2 {note pitch: g2 vel: 100}}  
{3.5 2 {note pitch: d2 vel: 30}}}  
play timed-seq(*ascore*)
```

Your first task is to define a new instrument that multiplies the output of a table-lookup oscillator by an envelope. The instrument should consist of (a) a table created as shown in class using the build-harmonic function and containing at least 8 harmonics, (b) an oscillator (use osc), and (c) an envelope. The instrument should be encapsulated in a function that at least has the two keyword parameters seen in *ascore* (pitch: and vel:). The vel: parameter represents velocity, a loudness control based on MIDI key velocity which ranges from 1 to 127. Convert to amplitude using the vel-to-linear function.

To play your instrument, transform *ascore* using one of the score-* functions to use your instrument rather than note. (It is your task to become familiar with the score-* functions and find the one that replaces function names.) Play the altered score. The resulting sound should demonstrate your instrument playing different pitches, durations, and velocities.

3.2 Record a short period of rhythmic tapping sound (<10 sec) and save as tapping.wav. This can be your own tapping or some other source.

3.3 Use the given code tap-detect.sal to detect a list of tap times and tap amplitudes of tapping.wav.

3.4 Write your own functions to map the list of tap times and amplitudes into a Nyquist score.

NOTE: Do NOT use score-gen. Instead, you must use loop and list primitives to construct a score. You may use the linear-to-vel function to convert tap strength to a velocity parameter.

3.5 Create at least 20 seconds of music using your code. You can mix multiple sounds or "tracks," you can create sounds by other means (as long as at least one "track" is derived according to the recipe above), and you can manipulate your results in an audio editor (again, as long as the tap-driven instrument is clearly heard).

Submission

For problem 1-2: a write-up contain the answers for question 1-2. Name it as

"andrewid_p2_shortanswers.txt"

For problem 3:

1) the code of 3.1. Name it as

"andrewid_p2_instrument.sal"

2) input signal of 3.3. Name it as

"andrewid_p2_tapping.wav"

3) the code of 3.4. Name it as

"andrewid_p2_tap-comp.sal"

4) the final mix of 3.5 and all related code. Name the sound file as

"andrewid_p2_tap-comp.wav" // .aiff is also OK

and put all composition related code under

"andrewid_p2_comp/*" // * means all the files related to composition

Add all your files to a new folder named YourAndrewID-P2, zip the directory and deliver it via Autolab.

Decibels

The Bel is a unit-less measure of the ratio of power to some reference. Each Bel represents a factor of 10. If we say the power of a signal increased by one Bel, it means that the power is now 10 times what it was. Here the previous power is the reference, and the current power relative to the reference is +1 Bel. We can calculate Bels with the equation:

$$B = \log_{10}(\text{power} / \text{reference_power})$$

Typically, we never actually use the Bel. Instead, the decibel, or dB, is used. As the name implies, 10 decibels make one Bel, so

$$\text{dB} = 10 \log_{10}(\text{power} / \text{reference_power})$$

But what about power? Power is work per unit of time, but it turns out in audio, samples represent voltage or pressure, which is not power. Electrical engineers should be familiar with: $P = V^2 / R$, that is, power is proportional to voltage squared. The same holds for pressure: power is proportional to pressure squared.

Thus, we can derive:

$$\begin{aligned} \text{dB} &= 10 \log_{10}(\text{power} / \text{reference_power}) \\ &= 10 \log_{10}(\text{amplitude}^2 / \text{reference_amplitude}^2) \\ &= 10 \log_{10}((\text{amplitude} / \text{reference_amplitude})^2) \end{aligned}$$

$$= 20 \log_{10}(\text{amplitude} / \text{reference_amplitude})$$

So, when amplitude changes by a factor of 10, that's a 20 dB difference.

When amplitude changes by a factor of 2, that's a $20 \log_{10}(2.0)$ which is approximately 6 dB.

Remember that 6 dB represents a factor of 2 in amplitude!