MUS 4711 –
Interactive Computer Music

Course Lecture Notes

MODULE 1: MIDI and Instrument Design

PART 4: Synthesis Types: AM, Basic FM, Ring, Karplus Strong

Objects – [preset], [loadmess], [loadbang], [umenu], [rand~], [stripnote], [tapin~], [tapout~], [selector~], [pong~]

Day 1 – Synthesis types

Subtractive – create complex sounds from an oscillator and noise source, multiple oscillators, multiple noise generators, etc. then filter them to remove unwanted frequencies. You can use as many filters as you want, but be careful about type and order so you're not masking or creating bumps through multiple res points stacking on top of each other!

Additive – classically, multiple oscillators, each of different frequencies are summed together to generate complex waveforms. With a little bit of patience, any periodic waveform can be generated in this fashion, including classical forms like saw, square (PWM), triangle, etc. as the sum of sines. The more complex version scales the amplitude of any partial above the fundamental frequency to simulate the relationship of partial frequency|amplitude that occurs in acoustically generated sounds. Also referred to as Fourier Synthesis as it's an application of the Fourier Theorem (any periodic signal can be represented as the sum of sines).

Keep in mind that it doesn't *have* to be sine waves, and can get ***extremely*** interesting with other waveforms subbed in…

AM – modulate the amplitude (volume) of an oscillator with the output of another oscillator scaled by a Depth of Modulation control (very important multiplier of 0-1 to keep things under control and not exploding). Here in MaxMSP, I've also added a DC offset (the inverse added to the Depth multiplier) to prevent drop outs. On a modular synth, this would happen without the need for an offset generator.

Classically, AM tends to focus on low frequencies, 0.000001 – 23Hz or so. Roughly the inaudible (or at least, non-speaker reproducible) frequency range. This creates the effect of tremolo as you can see/hear. Above that, and especially when the modulator signal is well in the audible range, you start to hear difference tones.

Ring Modulation – a close relative of AM, just without the DC offset. Note how it tends to be identical sounding in lower modulation rates, but is far more dependent on having ANY modulation present due to the lack of the DC offset. Indeed, for most cases the AM model with the DC offset is a better choice unless you're really worried about volume scaling. Works best if the frequencies are closely related (e.g. 500 and 510 Hz).

FM – modulate the frequency of an oscillator with the output of another (classically, these are both sine waves). Creates vibrato at low modulator rates with a high depth and DRAMATICALLY different sounds as you get into audio range. Note that the Depth of Modulation is up to the Nyquist frequency (SR/2). Observe how the partials are generated, especially when the modulator and depth are NOT in a nice harmonic relationship to each other and the fundamental.

FM with Wavefolding – I'm going to copy the previous FM patch, but now I'm going to make a few changes. First I'll use an [attrui] to set [spectroscope] into a Display Mode = sonogram, and that the Sonogram Color is turned on. Next I'm copying my scope, and I'm going to set Automatic Mode to on in the inspector. Finally, I'm adding the wavefolding object [pong~] after [cycle~] and before the volume multiplier. [pong~] has three arguments- the mode, lo folding value, and hi folding value. I'll create a umenu for the (mode $1) so we can set the values 0 is fold, 1 is warp, and 2 is clip – all of which have different effects on the timbre. Then I'll create two live.dials with a range of -1 to 1 floats for lo and hi folding value. With that, it's ready to go – watch the spectroscope, and the new scope to see how this shapes the harmonic content of the FM signal.

Karplus-Strong – Physical modeling using a noise source as excitation, multiplied by a VERY short "pluck" (really an envelope), then running it through delays and onepole filters, and feedback networks to shape it to sound like a plucked string. Named after the developers of the algorithm Kevin Karplus and Alex Strong.

Day 2 – Modulation
Continue on through synthesis types as needed. Review control oscillators ([cycle~], [phasor~], and [triangle~]), add depth and use to them to modulate parameters. Don't forget that you can use static values as well as [sig~]. Take the AM or FM and add modulation. Do the same with KP and Additive.

Work on Comp 1

<u>Comp 1 Project</u> – Take your three favorite synthesis methods that we've discussed. Create a [poly~] based patch that allows you to add in different amounts (with sliders or dials) of all three and a noise source to create a polytimbral voice. Add a filter of your choice to the final output of [poly~] (before [gain~] and [ezdac~]). Remember that all oscillators (and potentially a filter for your noise source) will all be using the same MIDI notes/velocity pairs.

Add a [preset] and save a preset for at least four different sounds you come up with on your instrument.

Create an engaging user interface for the instrument that allows the performers to interact with the most important portions of the instrument, and hides the rest through Presentation Mode. Your patch should be set to open in Presentation Mode.

The patch must make use of the MIDI keyboards, knobs, and buttons. You may map out parameters in any way you desire. It will need to be performed live using your computers.

You will need to write a short piece (3-5 minutes) for yourself to play using your instrument. Fully scored (graphical or notated), and indicating which presets to use, and where to change parameters with the knobs and buttons.