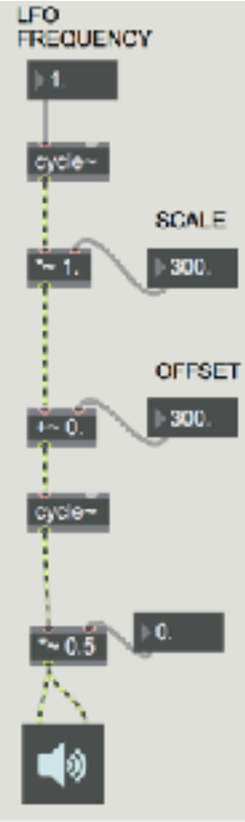
FM SYNTHESIS

One of the beautiful things about Max/MSP is the ability to explore connections - what happens when this object links to another object? How would it sound for this to connect to that? These possibilities can make Max a creative exploration of strange, unnatural, signal flows.

Peculiar things start to happen when signals get brought together. For this lesson, we'll look at what happens when a signal controls another signal's frequency in a common synthesis method called **FM Synthesis**.

LFO Modulation



Recalling the behavior of our Auto Filter effect in Ableton, we had an LFO (low frequency oscillator) that could control the filter cutoff at a subaudible rate. Let's create our own LFO to control the frequency of a sine tone. How might we do this?

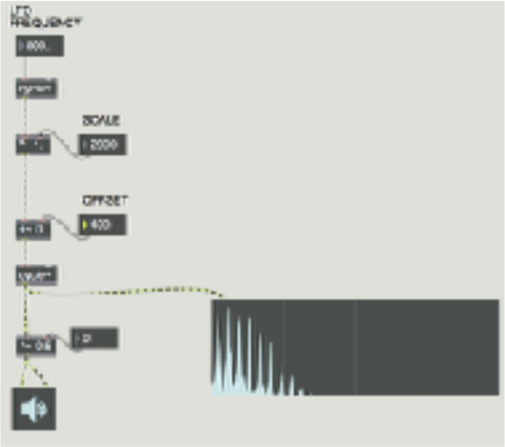
We have one sine wave acting as our LFO. We then scale this LFO and offset its value, and connect that to another sine wave. Play around with these values, try out different values. Don't be afraid of extremes!

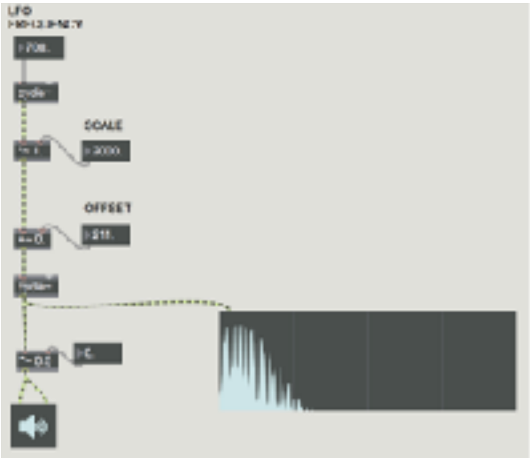
PATCH: fmsynth_ex0

Note: that this schema can be applied to any signal value - amplitude, filter parameters, effects, sequencers, etc. We will explore this later on.

When our LFO rate exceeds 20Hz (thus entering the audible range of 20Hz to 20kHz), the quality of the sound begins to change. The sound is no longer 'wobbly' but rather *textured*. This comes from harmonic patterning that is unique to FM synthesis. Let's take a look at a spectroscope~ to get a better sense for the oscillator's timbral behavior.

Exploring a bit, we find that when the LFO is at a frequency which is an integer multiple of the 'Offset' frequency, that the spectrum resembles that of a square or sawtooth wave. At non-integer ratios, the spectrum becomes more 'noisy'. **PATCH: fmsynth_ex1**

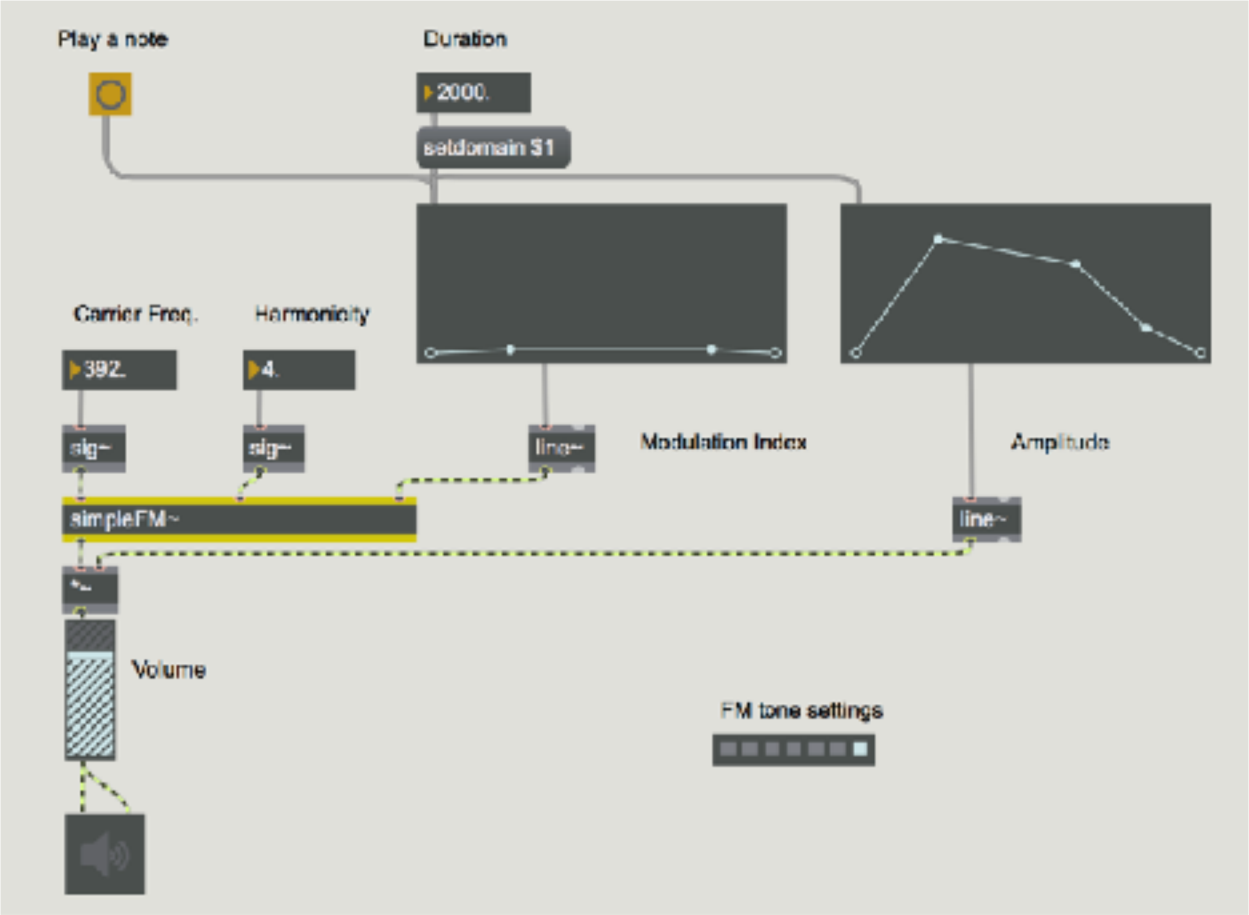




FM Synthesis

This system of relationships between oscillators has been formalized, and there's a specific lexicon that has emerged around it, primarily based on the work of John Chowning at Stanford in the late 1970's. The oscillator which is producing sound is known as the **carrier**, and the oscillator modulating the frequency is known as the **modulator**. The ratio between their frequencies is termed the **harmonicity ratio**, while the depth of the modulation is known as the **modulation index**. In the Documentation for Max, there is a wonderful example of their relationship, as well as an object called **simpleFM~** that encapsulates this functionality.

To make Max extra challenging, they've obscured how to get to the tutorials in recent releases of Max. If you right click the cycle~ object and open its reference, then scroll to the bottom of the page and open one of the Tutorial options, the documentation will open. You can hit the home button in the top left to get to the opening page. The tutorial we're looking for is in MSP, titled 'FM Synthesis". Let's take a look at this patch. **PATCH: fm_tutorial**



We can save this as our own patch and start making modifications. Bring over our keyboard from the other patches to make the patch more playable. How would we connect everything together? How could we map velocity to modulation index?

We can also make our presets using the **preset** object. The preset object (the grey squares where it says ‘FM tone settings’) will store all number box and function values, allowing you to quickly jump back to settings you enjoyed. Scale the preset box to make it bigger and allow for more presets.

Combining It All Together

The real majesty and mayhem of Max lies in the ability to tear apart and recombine *everything*. What if we used our oscillator selector from the subtractive synthesis example as our carrier or modulator oscillator within FM? How would that sound? What if our pitch determined our oscillator type? What if we layered an FM, subtractive and additive synth together? *What if a carrier modulated itself??* The possibilities are endless. Let’s play with these ideas individually for a bit before we conclude.