052483 - COMPUTER MUSIC: LANGUAGES AND SYSTEMS

POLITECNICO DI MILANO

MUSIC AND ACOUSTIC ENGINEERING

Homework 1: Supercollider

Group:
Radical Geeks (ID: 7)
Authors:
Gerardo Cicalese (ID: 10776504)
Alberto Bollino (ID: 10865248)
Umberto Derme (ID: 10662564)
Giorgio Granello (ID: 10869436)

Date: April 3, 2022

1 Overview of FM synths

1.1 History

John Chowning accidentally discovered this new form of sound synthesis while experimenting with vibrato at Stanford University. He found out that when the frequency of the vibrato effect exceeds a given threshold, it's not perceived anymore as vibrato but as a whole new complex tone.
He soon realized that FM is a very powerful method of synthesis and he became the first person to compose and record a piece of music using exclusively FM as means of sound generation. That's why we have decided to realize a chiptune version of a popular disco song as a demonstration for our implementation of an FM synth in Supercollider (2.1.3).

1.2 FM equation

Let's consider two oscillators, A_1 e A_2, described by the following equations:

$$A_1(t) = a_1 cos(\omega_1 t) \tag{1}$$

$$A_2(t) = a_2 cos((\omega_2 + A_1(t))t) \tag{2}$$

We note that A_1 modulates the frequency of A_2, thus we'll say that A_1 is the **modulator**, while A_2 is the **carrier**. Let's explicit A_2:

$$A_2(t) = a_2 cos((\omega_2 + a_1 cos(\omega_1 t))t) \tag{3}$$

1.3 FM in the time domain

Fig. 1 shows what happens to the carrier while it's frequency is being swept up and down by a modulator. In this example, the frequency of that modulator is significantly lower than that of the carrier, and the carrier waveform just looks like a cyclical squeezing or stretching of the original waveform, so we still can't appreciate the potential of FM synthesis.
Let's analyze what happens when the modulator's frequency approaches, equals, or even exceeds that of the carrier. In this cases, the modulation will become a form of distortion within the individual cycles of the carrier waveform. To demonstrate this, let's see what happens when a carrier is modulated by an high frequency modulator (Fig. 2). In this example, the modulator has a low amplitude, but is many times (approximately 60) the frequency of the carrier. In the next section, we'll see what happens in the frequency domain.

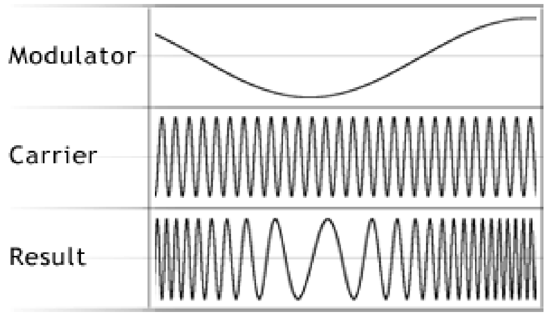

Figure 1: Modulating the frequency of the carrier with a low frequency modulator.

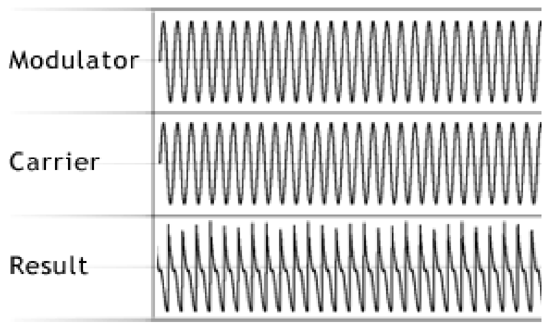

Figure 2: Modulating a carrier with a high-frequency, low-amplitude modulator.

1.4 FM in the frequency domain

First of all, let's define the parameters of our synth. The amplitude of the carrier does not affect the FM effect, while its frequency and modulator frequency are key to the FM effect (actually their ratio, as we'll understand later), along with the amplitude of the modulator. First of all, we'll analyze the effect of the modulator frequency, to which we'll refer with ω_m.
Chowning showed that FM generates side bands, not necessarily harmonically relatex to the frequency of the carrier or modulator. To see how FM produces side bands, let's consider, as an example, a carrier with frequency ω_c, and a modulator of frequency ω_m. It can be shown that FM produces a whole series of side bands that we can express as follows:

$$\omega_{sb} = \omega_c \pm n\omega_m \tag{4}$$

where ω_{sb} represents the series of side band frequencies and n is any integer from 0 to ∞. In theory, applying FM to a signal produces an infinite series of side bands. In the real world, of course, no analogue system has infinite bandwidth, thus we'll just consider a subset of the side bands. Anyway, digital FM systems just produce a subset of the side bands too, because only the ones which reside in the human hearing range are needed.
Thanks to the last equation, it's easy to see where the side bands are located (Fig. 3),

but we still don't know the shape of the spectrum (the amplitude of each of the side bands).
To know this information, we must consider the gain of the modulator, and introduce a new concept called the **modulation index**, or simply β.

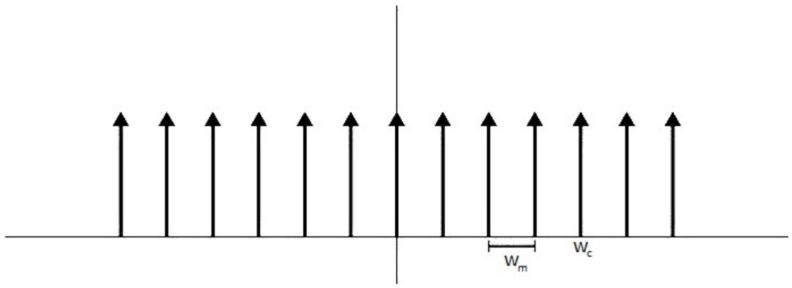

Figure 3: The positions of the side bands.

1.4.1 Modulation index

The FM effect we hear is not only determined by the frequency of the modulator, but, as we said, also by its gain or maximum amplitude. We thus define the modulation index as:

$$\beta = \frac{\Delta\omega_c}{\omega_m} \tag{5}$$

where $\Delta\omega_c$ is the carrier's frequency sweep, namely the amount by which the carrier deviates from its unmodulated frequency. This means that it's the modulation index (directly related to the amplitude of the modulator) which determines the amplitude of each of the components in the spectrum of the output signal.
When the modulation index is low ($\beta < 0.1$), the only significant side bands will be those closest to the carrier frequency. In contrast, if β is significantly higher ($\beta > 5$), we obtain a much broader series of side bands.
Let's introduce a rule-of-thumb to compute the effective bandwidth of the output signal:

$$B = 2\omega_m(1 + \beta) \tag{6}$$

Let's consider the Eq. 5 once again: the denominator is the frequency of the modulator. This means that, as the modulator frequency increases, β decreases. To make up for this effect, the modulator amplitude must be proportional to the modulator frequency.

1.4.2 Carrier:Modulator ratio

If carrier and modulator frequencies are arbitrarily chosen, you'll most likely obtain an enharmonic sound. That's not good, because we would like to obtain musically-pleasing

timbres.
Let's consider the Eq. 4, and let's express the relative frequencies of carrier and modulator using a ratio that we'll refer to as the **C:M ratio**. We can now say that, for any given carrier frequency, the frequencies of the upper side bands lie at:

$$C \pm M, C \pm 2M, C \pm 3M, \dots \tag{7}$$

This is an important result, because it immediately associates itself with the idea of harmonic series.
Let's consider the case where $C : M = 1 : 1$. If we considered just the positive frequencies, we would obtain a perfect harmonic series. Anyway, negative frequencies are out of phase w.r.t the positive frequencies. Luckily, there's no total cancellation because the amplitudes are not equal. The only disadvantage is that it's not trivial to compute the amplitudes, we would need **Bessel functions**, which we regard as being out of the goal of our assignment.

2 FM synth architectures

It's possible to produce a FM synth using just a handful of modules. If we consider the simplest architecture, a basic 2-oscillator FM, we just need two oscillators, a modulator and a carrier, some amplifiers to control their output, and eventually some envelope generators. The input to the oscillators is the pitch which, before reaching the modulator, is multiplied by the $C : M$ ratio. The amplitude of the modulator is amplified by the $C : M$ ratio times the modulation index (β).
In order to discuss more complex architectures, we need to introduce the concept of an **operator**: a combination of an oscillator, an amplifier and an envelope generator. Having done this, we can create all manner of routings (algorithms) in which operators affect each other in different ways. There are interesting combinations:

1. An operator affecting one or more operators (Fig. 4)
2. An operator affected by no operator or one or more operators (Fig. 5)
3. An operator affected by the output of an operator in the same forward chain (feedback, Fig. 6)

2.1 Supercollider implementation

Notice that we treat the $C : M$ ratio as a multiplication factor of the modulator frequency alone ($\omega_m = (C : M)^{-1}\omega$ and $\omega_c = \omega$), another option would be to set the carrier frequency to $\omega_c = C\omega$ and the modulator frequency to $\omega_m = M\omega$, but we believe it's just a complication for our architecture.

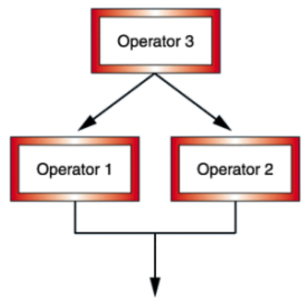

Figure 4: An operator affecting two operators.

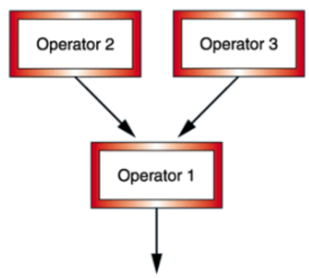

Figure 5: An operator affected by two operators.

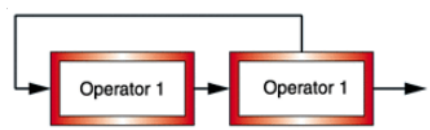

Figure 6: An operator affected by the output of an operator in the same forward chain.

The reason why we are defining two synthdefs is that we want our output to be stereo while, at the same time, we want our modulators to be monophonic, in order to keep the architecture simple. If we wanted to be consistent with the philosophy of reusability explained in 2, we should have created only one synthdef, a more generic **operator**.

2.1.1 Architecture

We have decided to let the user create its own architecture, despite some limitations to keep the code simple. The user can use up to five operators, included the carrier (stereo), and there are three parameters that can be configured:

1. Activation of the left feed forward chain (enabling/disabling the modulator 3, directly connected to the carrier)

2. Mode of operation of the left feed forward chain (choosing whether the input of the modulator 3 is no input or the output of modulator 1 or the carrier (feedback))

3. Activation of the right feed forward chain (enabling/disabling the modulator 4, directly connected to the carrier)
4. Mode of operation of the right feed forward chain (choosing whether the input of the modulator 4 is no input or the output of modulator 2 or the carrier (feedback))

As we said before, we are discarding the idea of operator in order to simplify our problem, and to let our carrier use not only a sinusoidal oscillator but also other waveforms. The modulators and the carrier communicate through some busses.

Modulator Our modulator consists of just a sinusoidal oscillator operating at frequency $\omega_m + \Delta\omega_m$, where $\omega_m = (C : M)^{-1} \cdot f_0$ is the unmodulated frequency of oscillation and f_0 is the frequency corresponding to the note currently played by the instrument, while $\Delta\omega_m = f_{in}$ is the frequency sweep and f_{in} is the input to the modulator.
It's output is mono and it's multiplied by a simple percussive envelope, so the user can specify the attack and release durations. The gain is the modulation index β times the unmodulated frequency (center frequency) of oscillation ω_m, namely $A_{max} = \beta \cdot \omega_m$, in order to make up for the effect explained in (1.4.1).

Carrier Our carrier consists of an oscillator, which can be:

1. A sinusoidal oscillator
2. A triangle wave oscillator
3. A sawtooth wave oscillator

The frequency at which this oscillator operates is $\omega_c + \Delta\omega_c$, where $\omega_c = f_0$ and f_0 is the frequency corresponding to the note currently played by the instrument, while $\Delta\omega_c$ is the frequency sweep and f_{in} is the input to the carrier.
It's output is multiplied by a simple percussive envelope, analogue to the modulator's one, and thanks to a panner we upmix the sound from mono to stereo, and as a lucky bonus we can set the panning. The gain is controlled by the user.

2.1.2 FM synth as an instrument

We have developed an interface to control our FM synth as an instrument. The user can use a midi keyboard as an input method, giving him the possibility to play his favorite tunes with a sound of his own creation. The user can configure the FM parameters of each modulator and of the carrier thanks to some knobs and he can choose the topology too (algorithm), thanks to some buttons.
The first four knobs' columns of the GUI each control one modulator ratio, attack, release and index, where attack and release are simple sound envelopes parameters while the ratio $(C : M)^{-1}$ and the index β have already been defined in this paper. The fifth

knobs column concerns the master (carrier), and it's where we can find the master volume, which controls the general volume of the sound, and the master panning, which controls the general panning.
Every time the user changes the topology, a visual feedback is provided by a small display in the bottom right corner of the GUI.
In the top right corner of the GUI the user can find five different buttons:

1. Scope: allows the user to see the scope when pressed;
2. Freqscope: allows the user to see the frequency scope when pressed;
3. LOAD: allows the user to load some predefined preset (saved in a CSV file) into the current SC session;
4. SAVE: allows the user to save the values of every knob and the topography of the FM synth in a selected CSV file;
5. Play demo - stop demo: lets the user play a Demo that was created by the developers.

2.1.3 Sequencer demo

The FM synthesis gives the user endless different sounds possibilities and to prove it our team has created a demo song. The three voices (bass, kick and hi-hat) have been created just using two modulators and a carrier inside a Task function:

1. Bass task: a midi note sequence array is played punctuated by a duration array;
2. Kick task: the kick sound is played every 0.5 seconds;
3. HH task: the hi-hats sound is played every second.

Then these task functions are played together as soon as the 'play demo – stop demo' is pressed.