10610731 – 10619659 – 10806125 – 10868569 - 10574372

Homework 1 – Assignment 2
Virtual instrument with FM synthesis

Link GitHub: https://github.com/polimi-cmls-22/group12-hw-SC-Ratatouille

Project Target

The purpose of this project is to create a virtual instrument capable using FM synthesis and an interface for controlling it.

FM synthesis theory

Frequency modulation synthesis (or FM synthesis) is a form of sound synthesis whereby the frequency of a periodic signal, called *carrier*, is changed by modulating its frequency with other signals, called *modulators*. This apparently simple method gives the possibility to the user of creating several patches and timbres of a sound without the use of filters.

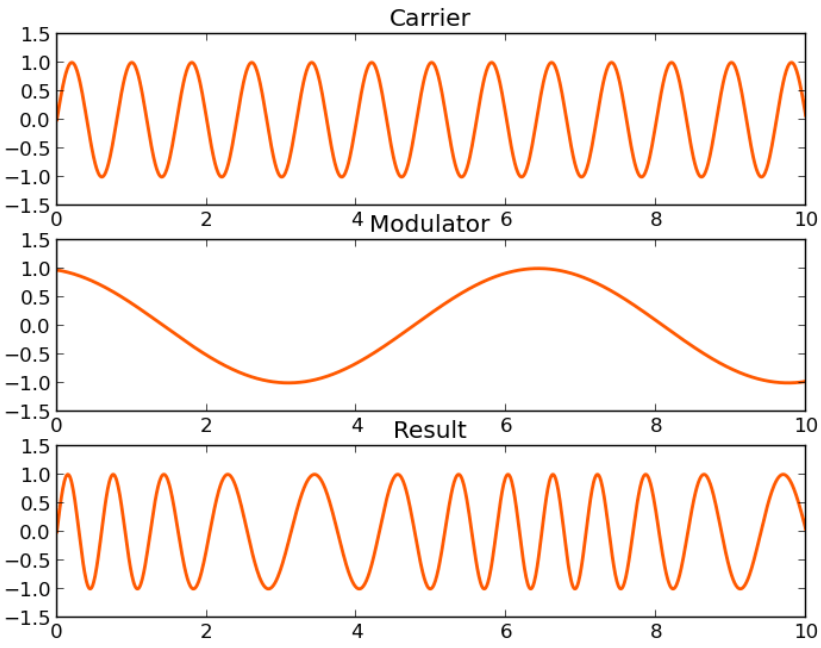


Figure 1 FM operating principle

Increasing the modulator frequency, instead of looking like a cyclical 'squeezing' and 'stretching' of the carrier waveform, the modulation will generate sidebands frequency, perceived as a distortion of the carrier waveform. The relation between the carrier and modulator frequencies is described by a quantity called Modulation Index. Let's call f_c the carrier frequency, f_m the modulator frequency and Δf the peak frequency deviation of the carrier. The modulation index is defined as follows:

$$I = \frac{\Delta f}{f_m}$$

When $\Delta f \neq 0$, sidebands appear above and below the carrier frequency, at multiples of $\pm f_m$.

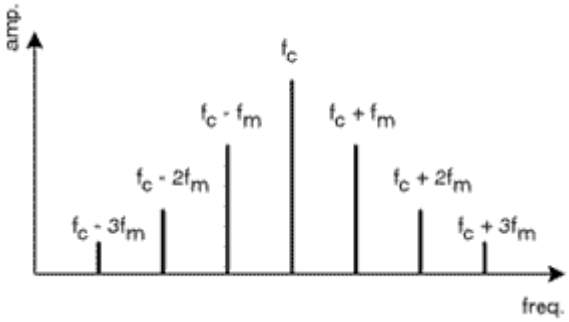


Figure 2 Output spectrum sidebands

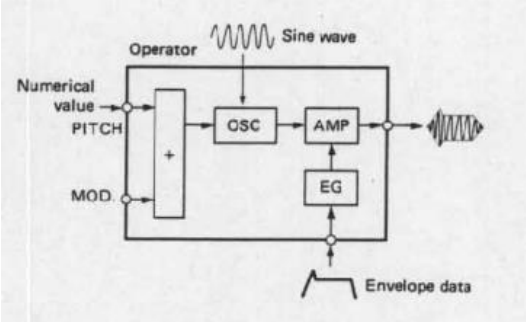


Figure 3 Components of an operator, input parameters and output

The basic structure of every FM synthesizer is a series of identical components called operators. Each operator is the digital representation of an oscillator, envelope generator and amplifier. The operator acts either as carrier (makes a sound) or a modulator (which modulates another operator). The carrier signal determines the pitch of the note produced, and the modulator determines the shape or timbre. The operators are grouped together into preset algorithms, which allow the creation of a great variety of effects. In addition to this, every operator can be independently switched on and off, significantly tweaking the final output.

Every operator consists of different parameters which allow to control its pitch and dynamic behavior.

WOODY FM synthesizer implementation

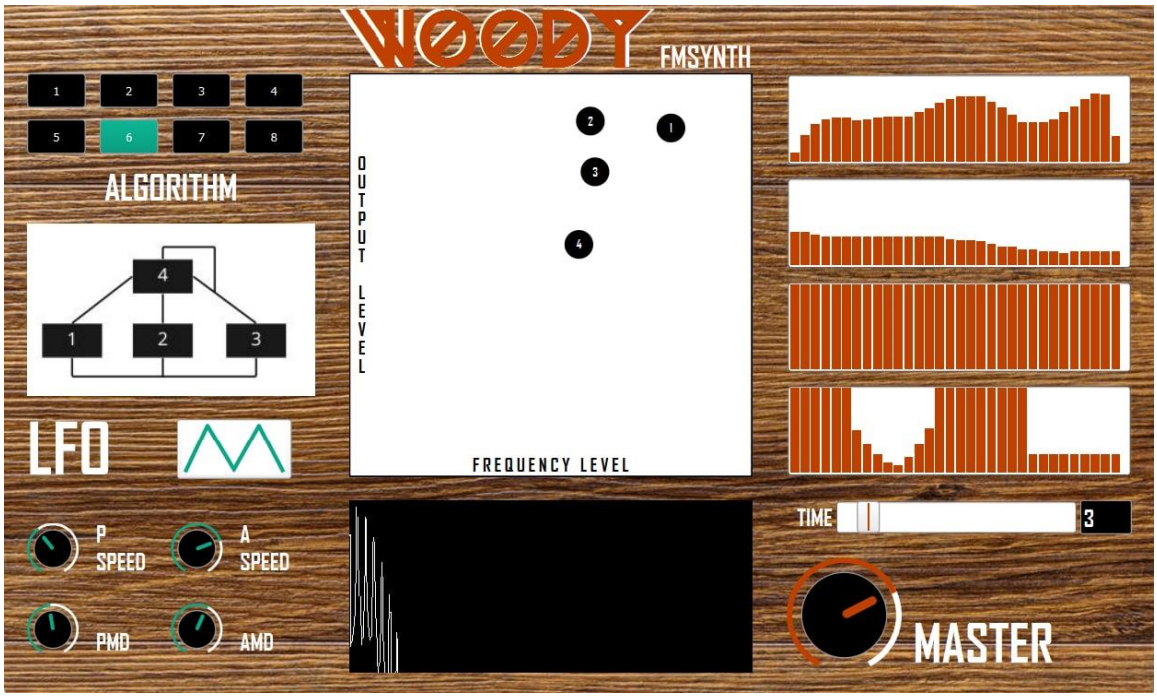


Figure 4 WOODY GUI

Modulation Index formula

For our project, we took as reference the Yamaha DX7, which debuted in 1983 and became the first mainstream commercial FM synthesizer.

Because of the DX7 ultimate computational complexity, we significantly reduced the number of components and parameters. In addition to this, the DX7 exploits both frequency and phase modulation, while our goal was to implement the characteristic features of a pure FM synthesizer.

The key FM formula we used to implement WOODY modulation is the following one:

$$y(t) = A\sin(2\pi f_c t + I\sin(2\pi f_m t))$$

In this formula, I represents the modulation index. The next step was finding a relation between the modulation index and the output level of the modulators (set by the user).

Since we referred to the DX7 as a model, we defined the variables *~outLevelToTL* and *~outLevelToIndex*, in order to explicit the mathematical relation between the two quantities.

For DX7, we had $I = \Pi \times 2^x$ and $x = (33/16) - TL/8$,

where the TL values represent an intermediate step between the output levels (enclosed in the Level Tables matrix) and the modulation indexes.

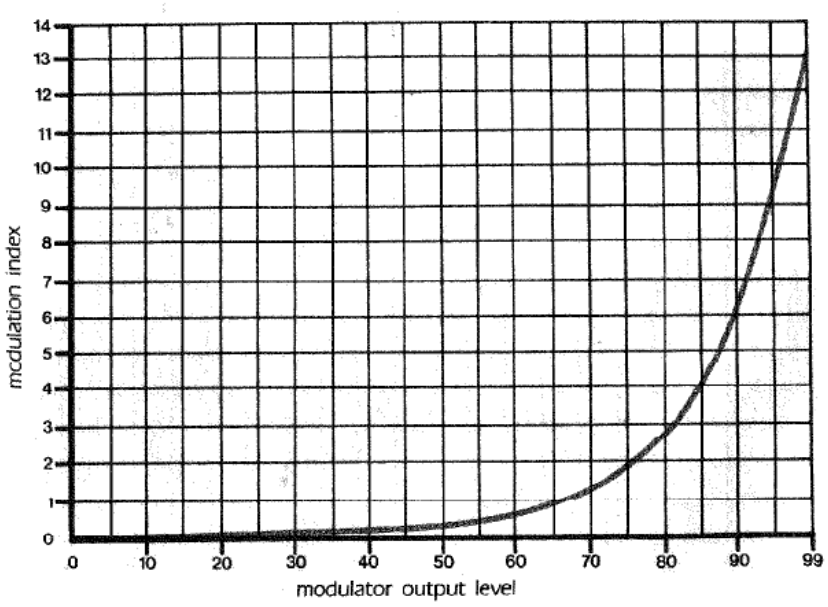


Figure 5 DX7 – Index vs. Output Level

Algorithms

The term “algorithm” refers to the connection between different operators. In FM jargon, the bottom operators on each stack are carriers and the ones above it are modulators. As a result, if you select an algorithm and switch off the assigned carries, you’ll get no sound. Another important concept regarding algorithms is layering.

We implemented the algorithms shown in the image below:

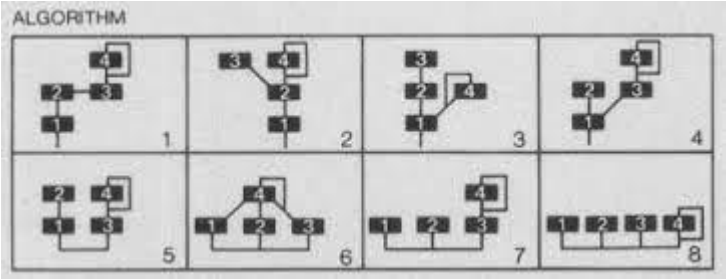


Figure 6 Four operators' algorithms

Using an FM synth, it's possible to select both parallel and cascade operator disposition. The first ones involve multiple carriers at the same time, mixed at the output; the second ones allow to obtain multiple stack modulations, which require an ordinate modulating action reflected on the same carrier.

We implemented a SynthDef for each algorithm. Each SynthDef differs from the others for the combination (in terms of frequency, ratio and modulation index) and number of carriers and modulators.

The connection between the GUI code and the correspondent SynthDef is provided by the variable currentAlg which contains a reference to the current algorithm.

As regards the graphical user interface, we wanted to blend a classical appearance (provided by the wooded background and the canonical disposition of commands) and a more creative interaction with the user. Starting from the top left corner, eight numbered keys allow the selection of the desired algorithm, which is schematically shown in the window below.

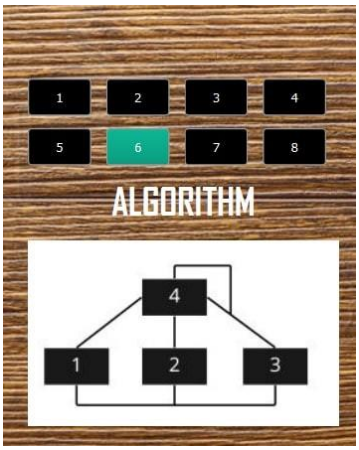


Figure 7 WOODY GUI: algorithms

Operators' control panel

In WOODY all the operators are set in the so-called ratio mode. If the operator (due to the selected algorithm) acts as a carrier, the oscillator follows the note being played. In this case, regulating the frequency means multiplying the fundamental pitch (determined by the MIDI note) by a precise factor, from 0.5 (one octave down) through 1, 2, 3 etc. For example, 1.5 means multiply the frequency by 1.5, which gives a pitch a perfect fifth higher. The output level simply refers to the volume.

Whereas, if the operator is acting as a modulator, changing the frequency means setting the c:m scale, establishing a relation between the carrier and the modulator frequency. The value of this ratio has huge implications on the final output since it affects the numbers of harmonics. For example, a fractional number, generates "inharmonic" information that will likely make the sound discordant.

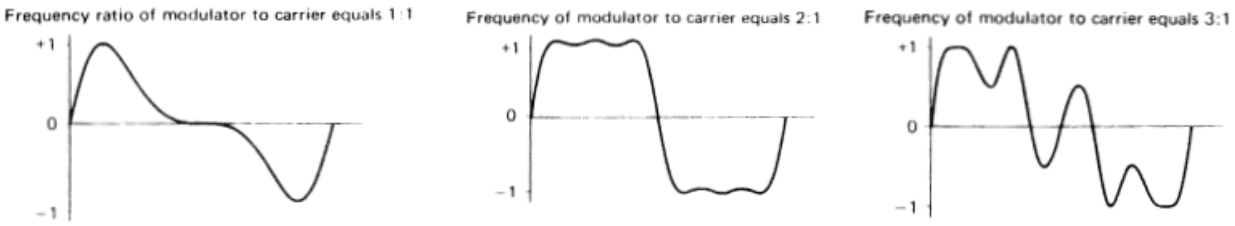


Figure 8: Examples of output based on different frequency ratios

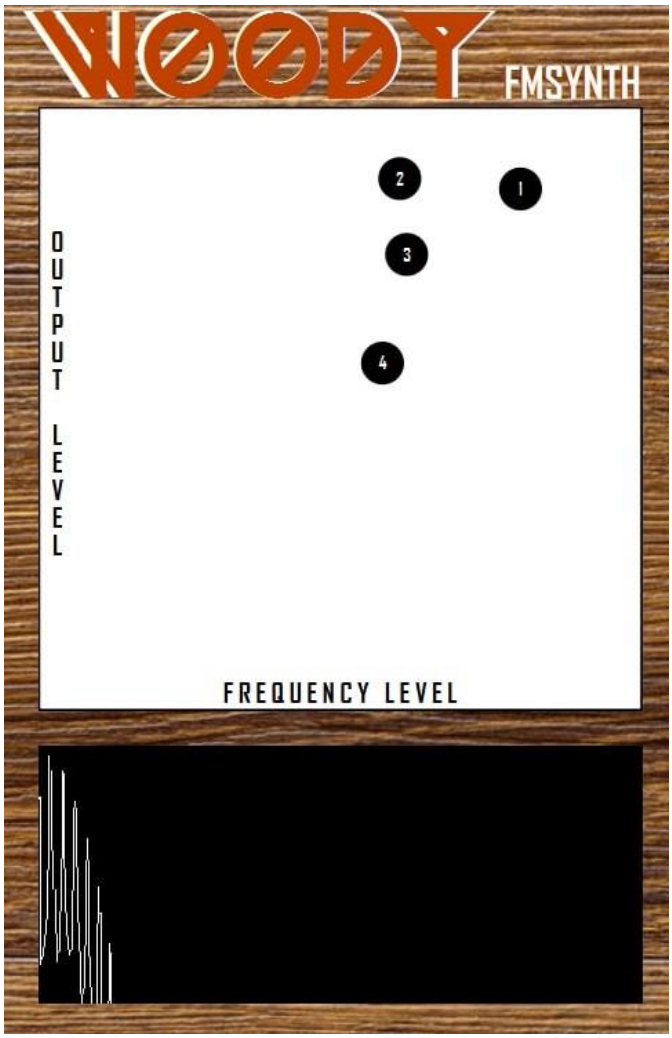


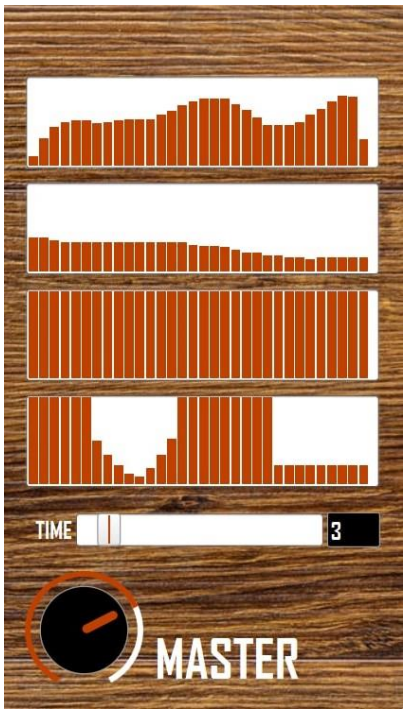
Figure 9 WOODY GUI: operators' control pannel

In WOODY, the central section is divided into two different parts: the top one shows a two-dimensional diagram which allows the control of output and frequency levels, while the bottom one displays an oscilloscope of the output signal.

The location of each point on the plane can be edited by the user; depending on the algorithm, output and frequency level will automatically change their aim with respect to the final output.

Dynamic controls

The right column of WOODY's interface contains the envelopes. Each operator's trend over time can be computed using four envelopes' levels and the corresponding envelopes rates (following the ADSR model). The levels indicate the target the envelope is supposed to reach during the attack, decay, sustain or release segment of the note. The rates indicate how fast it travels from where it ended the last segment (or 0, at the start of the note) to this target level.



If an operator is acting as a carrier, its envelope determines the volume of the sound in the usual way. If it's acting as a modulator, though, its envelope affects how the harmonic character it adds to the sound varies in intensity over time.

We implemented four histogram-like windows (one for each operator), where the envelope trend is shown. Instead of using control knobs for rates and levels, the user can freely draw the envelope directly with cursor movements. The horizontal extension of the envelope is also variable and gives the user the possibility to select from 1 to 16 samples and consequently different accuracy levels.

The slider below controls the total duration of the signal, from 0.1 to 30 sec, while the knob regulates the master volume.

Figure 10 WOODY GUI: envelopes, time and volume controllers

LFO section

The oscillator of an LFO is of low frequency, to be set between 0.1 to 20 Hz, which can't be heard but can influence other parameters of the synthesizer sound, to give life and movement to the sound that is heard.

WOODY's bottom left section is entirely dedicated to the LFO. The basic controls for the LFO are its waveform, speed and depth. In addition, the user has the possibility to apply the LFO both to the amplitude (AMD) and to the pitch (PMD) of the output signal, obtaining respectively the tremolo and vibrato effects.

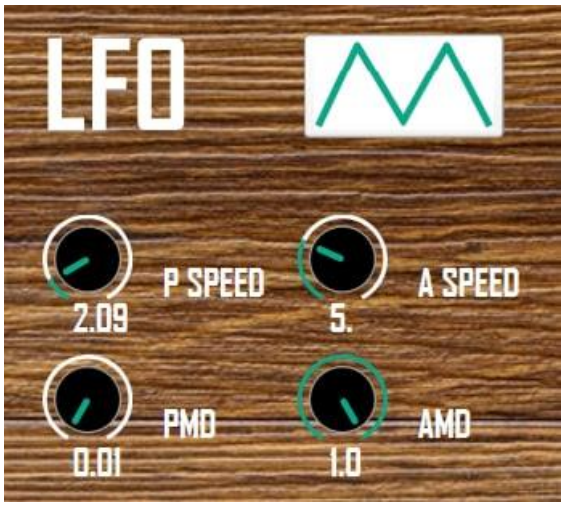


Figure 11 WOODY GUI: LFO

Conclusions

WOODY is an essential FM synthesizer, capable of modulating sinewaves modifying their instantaneous phase. Despite all the main functions for frequency modulation which have been implemented, there's still room for improvement.

During the implementation process, we tried to implement a unique SynthDef, capable of displaying all the algorithms. Later, due to SuperCollider impossibility to host arrays as arguments for the SynthDef, we had to implement a separate SynthDef for each algorithm, at the expense of the code optimization.

One of the reasons why we didn't set a precise grid for output and frequency modulation is to encourage the user to develop a more creative approach to sound synthesis. The lake of reference points allows freer operator movements: in this way it's possible for the user to create a unique association between visual patterns and precise sounds.