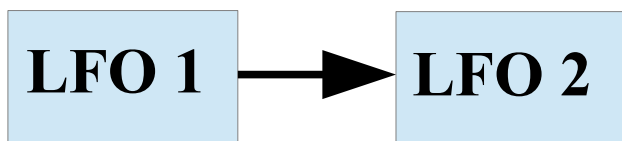
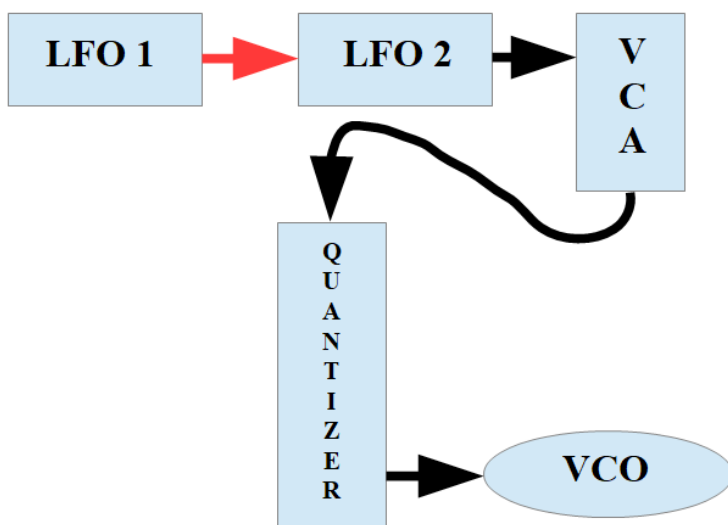


I leave the field of pure additive LFO networks now and turn to LFOs in series. Here one LFO modulates the frequency of the other one:



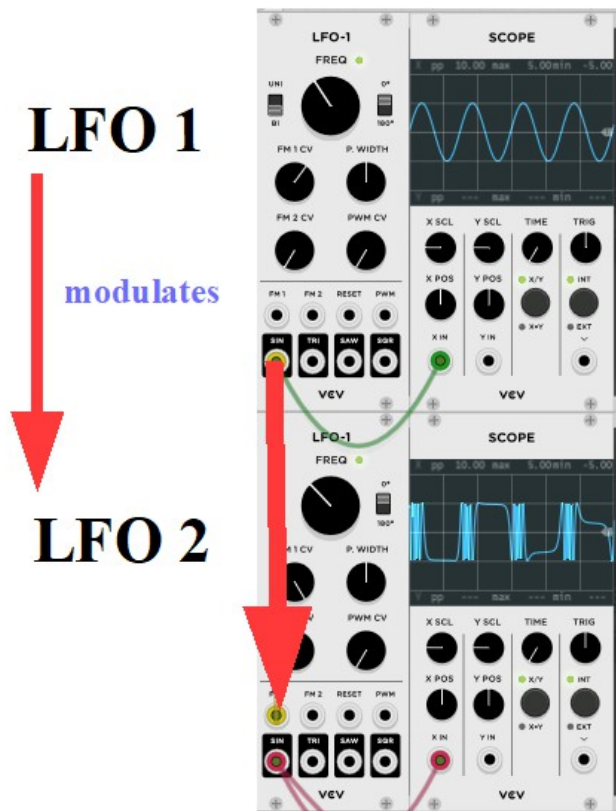
LFO 2 can be patched into a quantizer then, and from the quantizer the CV goes to the 1 Volt/octave input of an VCO. At first only the output of the second LFO shall be attenuated to set the frequency range of the VCO:



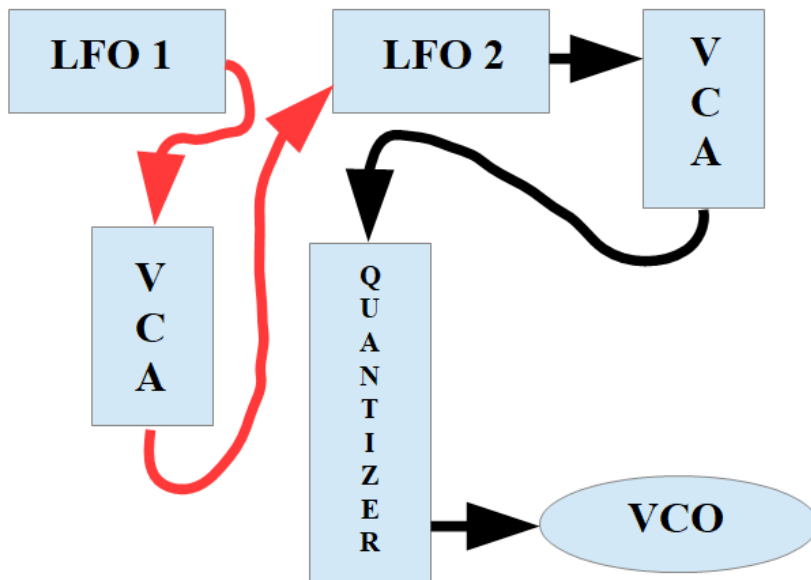
The result is quite predictable: LFO 2 lets the melody go up and down the scale. Without the modulation by LFO 1 and in case we are using only sine waves, the melody would develop slower at its bottom turn as well as at its top turn, because there the value differences of the sine wave are quite small, and therefore the quantizer will give out the equivalent pitches for a longer time.

But there is the modulating LFO 1, which modulates the overall time the circles of LFO 2 need to be completed. Therefore the melody develops slower than “natural” sometimes, and sometimes it runs quite fast.

The cycles of LFOs 1 and 2 may look like this (just an example):



Patching a VCA between the two LFOs – or using the “FM CV” knob of our LFO, if it has one – we can adjust the strength of the modulation.



The frequency of LFO 1 (modulating frequency) determines **how fast** the changes in the “speed” of the melody happens, whereas the strength of the modulation by LFO 1 determines **how much** the speed of the melody changes.

If the frequency of LFO 1, the modulating frequency is higher than the frequency of LFO 2, the modulated frequency, then we notice the effect, that the modulating LFO 1 determines the development of the melody more than LFO 2. which directly modulates the Quantiser-VCO connection, because the changes of the length of a circle of LFO 2 are going faster than on wave (circle) would “naturally” last.



The video behind the link demonstrates this, and the preset “two_in_series.vcv” (comes with the book) serves as a starting point for experiments of your own.

<https://youtu.be/Ocfe6qrP5G8>

Well, there don't seem to be many interesting aspects in this simple in-series network, at least not as far as generative music is concerned. Let's go a step further therefore, and this step is: feedback!

And let's use only one single LFO at the beginning. One LFO with its output patched back in its frequency/rate CV input. As the vast majority of LFO modules deliver more than only one wave shape, and most of them

are sine, triangle, saw and square we have to investigate all possible pairs of modulated vs. modulating shapes. Appendix A contains a comprehensive collection of graphs showing the result of these feedback combinations, with different strengths of modulation. And the video behind the following link shows how I have gained them.

<https://youtu.be/unc7Rh1Ty7Q>

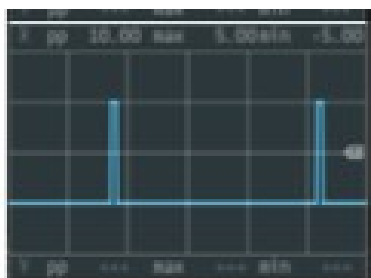
As far as generative music is concerned we are interested only in different and changing **developments** of the CV, which the feedback loop delivers to VCOs and other modulation targets. Therefore we won't use the pure internal feedback "output to frequency/rate CV input". Instead we will patch an external VCA between the output of the LFO and its frequency/rate input, a VCA, which we can modulate.

Let's go for some examples.

I combine (additive) the following constellations:

LFO 1:

square modulated by saw at a frequency of 0.102 Hz, unipolar, modulation strength 100%, signal share 23.5%. The corresponding graph (see Appendix A) looks like this:



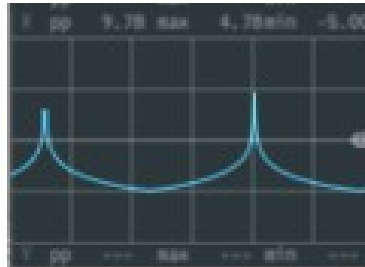
LFO 2:

sine modulated by saw at a frequency of 0.447 Hz, bipolar, modulation strength 100%, signal share 43.9%. The corresponding graph (see Appendix A) looks like this:



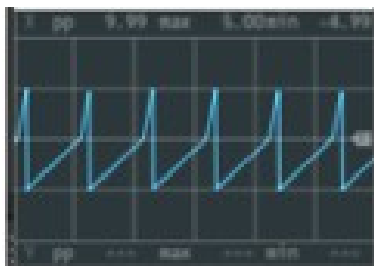
LFO 3:

triangle modulated by triangle at a frequency of 0.033 Hz, unipolar, modulation strength 100%, signal share 70.45%. The corresponding graph (see Appendix A) looks like this:

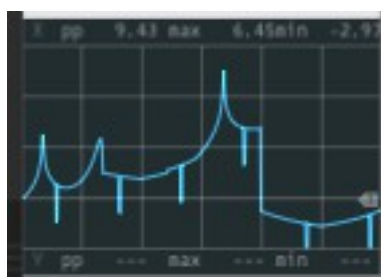
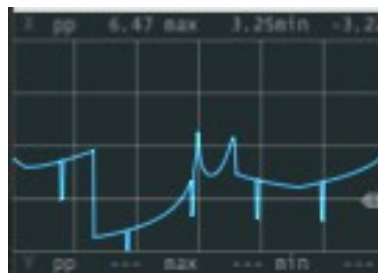
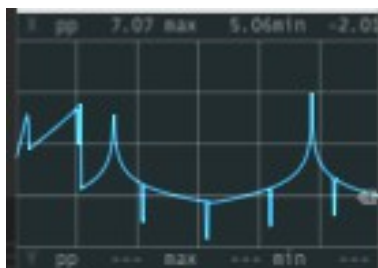


LFO 4:

saw modulated by square at a frequency of 0.145 Hz, bipolar, modulation strength 25%, signal share 53.35%. The corresponding graph (see Appendix A) looks like this:



The seemingly infinitely changing development of the summed CVs look e.g. like this (only some snapshots of the ever changing graph):

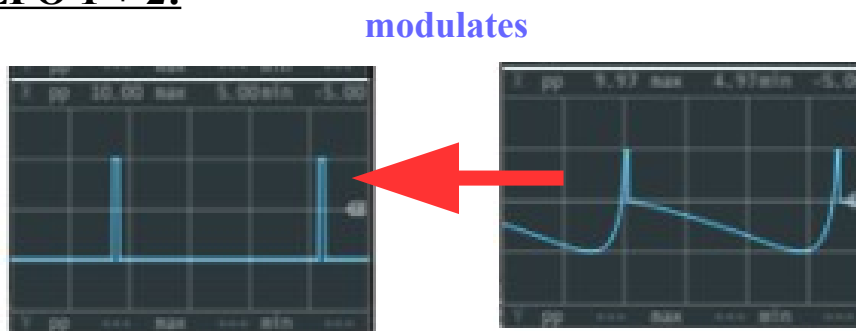


The preset “FB_example_1” is set up like that. Just go and experiment with different frequency relations of the LFOs and different VCA settings. The following link leads to a video, which shows me doing so.

<https://youtu.be/OfzdDK5mm94>

My second example (preset “FB_example_2”) uses the same 4 LFO constellations, only that they are paired two-and-two, meaning, that LFO 2 modulates the rate of LFO 1, and LFO 4 modulates the rate of LFO 3. All self-modulations stay active. LFOs 1 and 3 are modulated by two different sources therefore: by one other LFO as well as by themselves. Using the CV graphs from before the situation can be described by:

LFO 1 + 2:

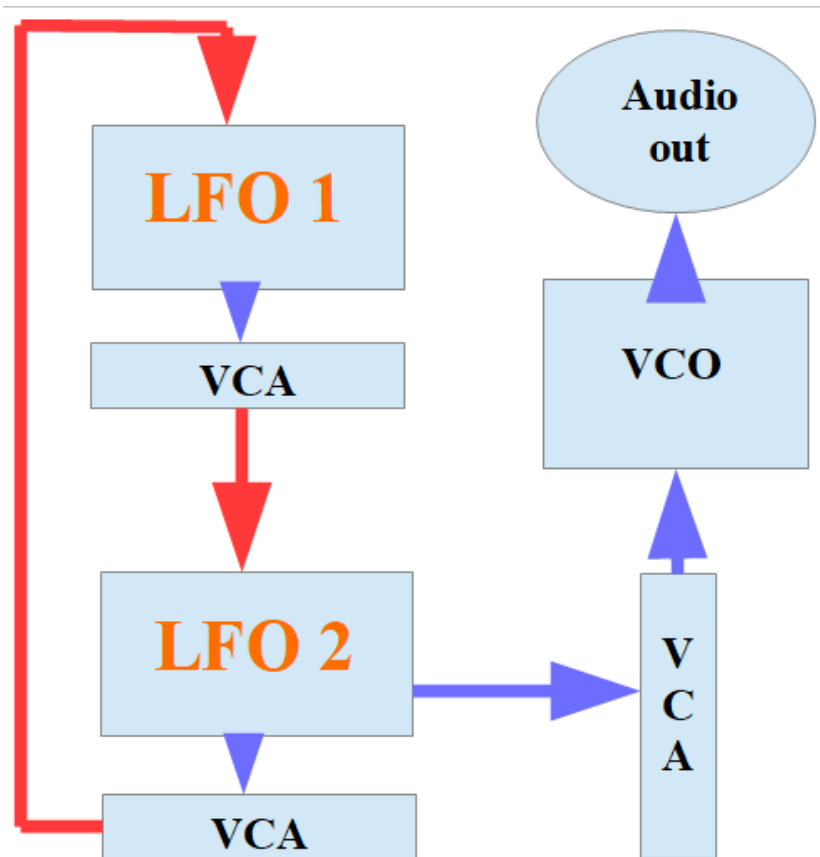


And there is a video to show the situation (of course there is one; just click the following link).

<https://youtu.be/BBp0rlqLNoo>

With this example we have already set one foot into the next level: feedback loops containing more than only on single LFO. Only one foot, because we don't have any feedback between the two LFOs. None of the 4 feedback loops leaves its own LFO. Let me change this now. And for reasons of compatibility I don't use the internal functionality to regulate the strength of the modulations and/or the feedbacks, but patch all CV signals through external VCAs.

The simplest of all possible patches of this category looks like this:



The upper VCA in the feedback loop determines how much the rate of LFO 2 is modulated by LFO 1, and the lower VCA in the feedback loop sets the strength of the feedback modulation (LFO 2 modulates LFO 1). The right VCA between LFO 2 and the VCO determines the strength of the overall pitch modulation of the VCO by the LFO network.

The preset called “FB_twoLFOs_1.vcv” is set up like this. Just go and mess around with it if you like. And the following link leads to a video demonstrating the patch, and showing the different CV developments and resulting modulation shapes.

<https://youtu.be/y29slGjHRfg>

And even if I explain the patch in the video it might be a good idea to describe some general principles here in the written part now.

The data of this example are:

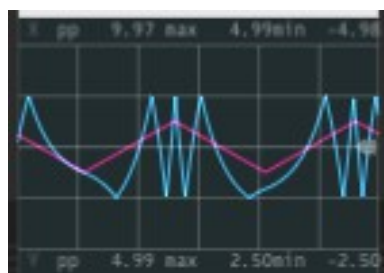
LFO 1: frequency/rate: 0.25 Hz
 wave: triangle
 output level of the VCA: 50%

LFO2: frequency/rate: 0.5 Hz

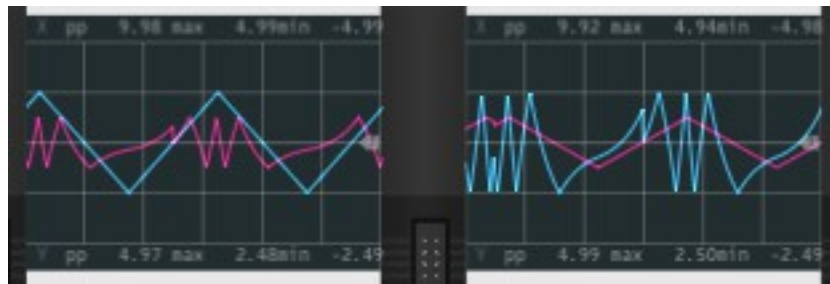
wave: triangle
output level of the VCA: 50%

The CV development WITHOUT the feedback from LFO 2 to LFO1 looks like shown in the following image. The blue curve is the output of LFO 2, the pink curve is the output of LFO 1. We have the situation: LFO 1 modulates LFO 2. No feedback so far. The blue curve (the output of LFO 2) is the result of the modulation of LFO 2 by LFO 1, the pink curve is the modulating signal. The original signal of LFO 2 is not shown – well, it's simply a triangle wave of double the frequency of the pink curve.

We see, that always when the **amplitude** of the modulating pink signal increases the **frequency** of the blue curve increases too.

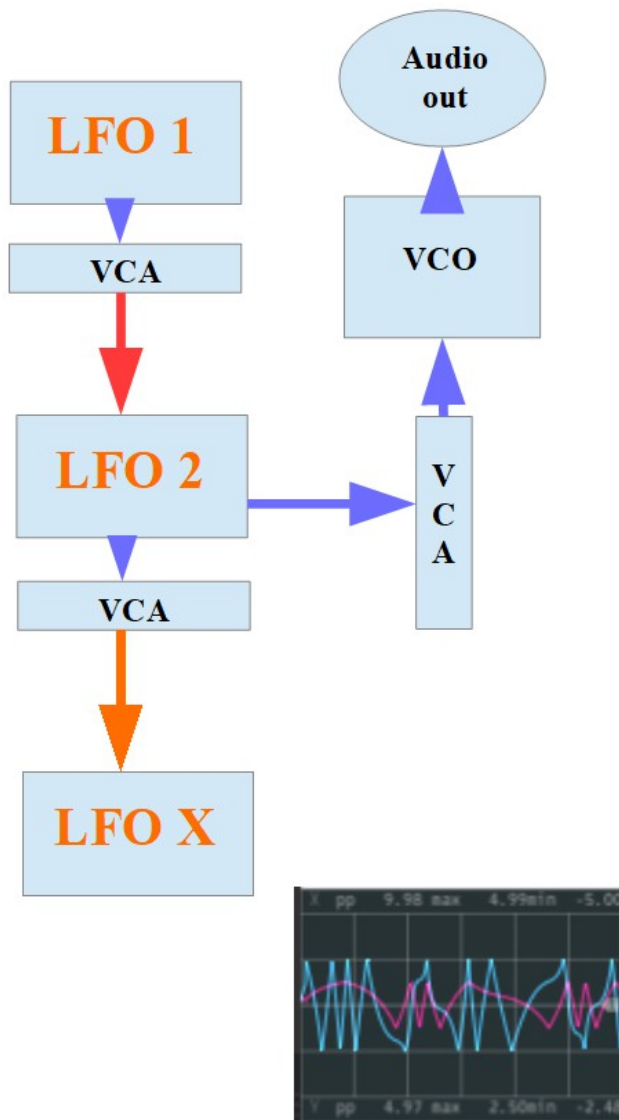


I've patched a scope to both LFO outputs. Let's have a look at these two scopes first, before I patch the feedback loop:



The left graph(s) show the actual output of LFO 1, the modulating LFO, and the right graph(s) show the actual output of LFO 2, the modulated LFO.

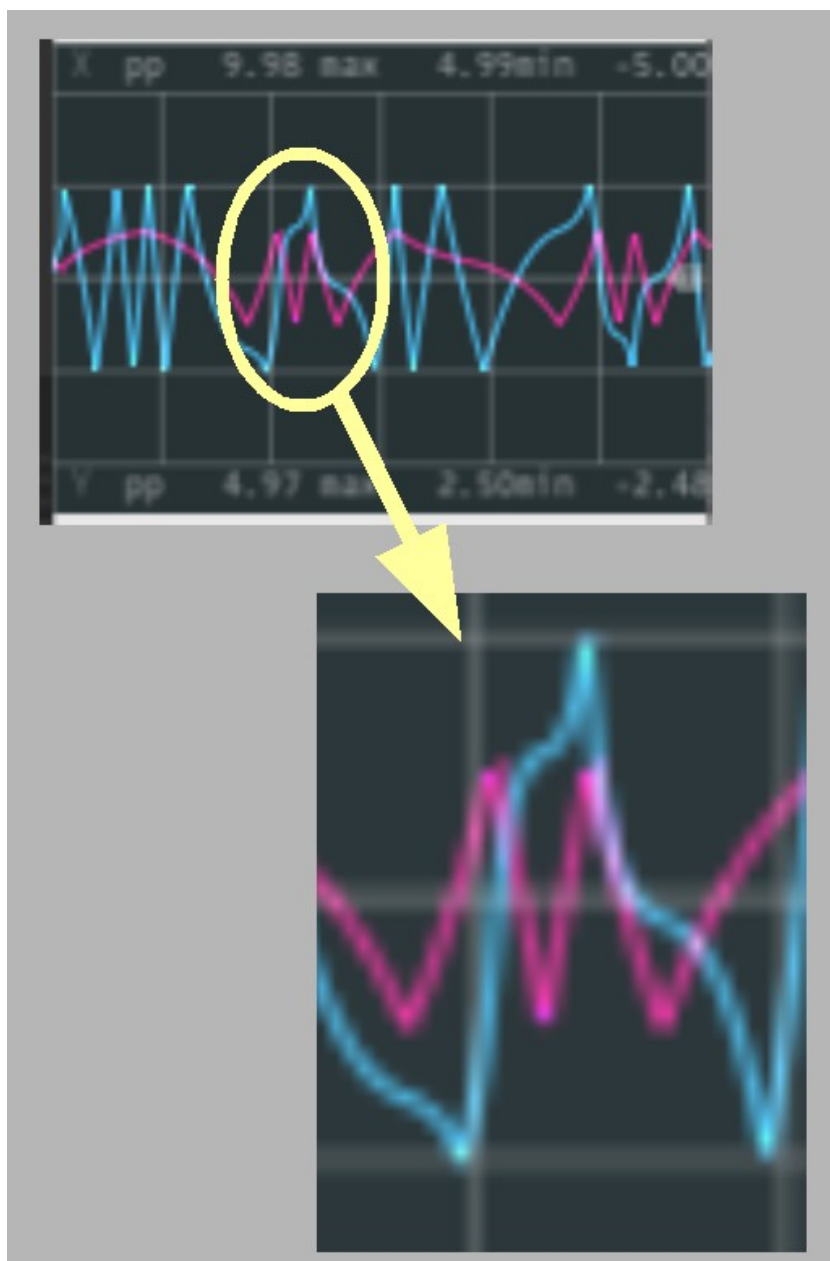
And again the pink graph is the output of the OTHER LFO: on the left scope we see the blue CV graph of LFO 1 and the pink CV graph of LFO 2, whereas the right scope image shows the blue graph of LFO 2 and the pink graph of LFO 1. The pink graph on the left side and the blue graph on the right side are the same therefore. (As well as are the blue graph on the



left and the pink graph on the right.) To understand the next step, the feedback loop, I insert a third LFO, let me call it “LFO X”, and modulate its frequency by the output of LFO 2 as we have it so far.

We see, that the blue graph – the result of the modulation of LFO X by LFO 2 and the pink graph – the output of LFO 2 – perfectly relate to each other. Always when the amplitude of the output of LFO 2 increases, the frequency of LFO X increases as well and so on.

There is one very interesting point in the graph (meaning: there is one very interesting situation in the development of the Cvs). I’m talking about the two quite fast rising and falling amplitudes of the (pink) output of LFO 2.

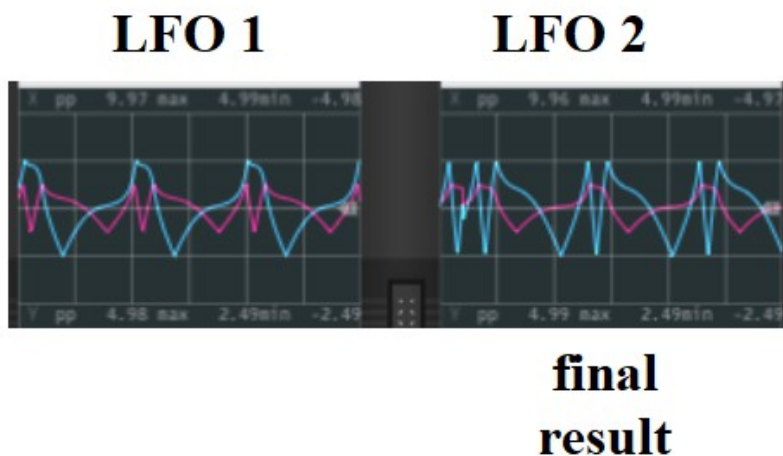


The amplitude of the modulating (pink) CV signal goes up, and the modulated (blue) CV signal tries to increase its frequency, but before it can complete its circle it is caught by the again falling of the amplitude of the modulating (pink) CV signal. Therefore the modulated (blue) CV signal decreases its frequency (= enlarges the length of its cycle and the graph builds this little saddle), before the next rise of the amplitude of the modulating (pink) CV forces the blue graph again to start increasing its frequency (= shortening the length of its cycle).

It is important to understand these goings on. Otherwise we would not grasp what's going on, when I now replace LFO X again and patch the complete feedback loop (use the preset "threeLFOs_beforeFB.vcv" for experiments of your own).

When the feedback loop is active, the blue curve of LFO 2 depicts the CV, which is going to modulate LFO 1, which – in the first step – should (and does!) generate something like the blue curve of LFO X, which then will modulate LFO 2 again, which then (re)modulates LFO 1 again and so on.

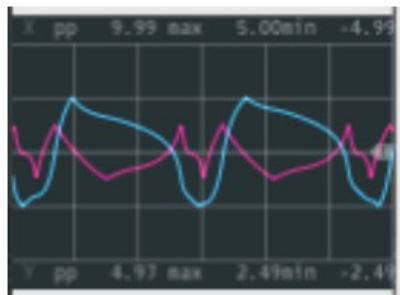
Like in any feedback loop a certain wave shape stabilises, which in our case looks like this:



There is still one step missing: How does the CV development look like, that is generated by LFO 2 modulating LFO 1 in the aforementioned first step?

Well, inserting another LFO-Scope combination in the patch, and patching the output of LFO X to this new combination of – let's say LFO Y + Scope gives the answer (because LFO 1 is still producing simple triangle waves as long as it is not part of a feedback loop. So, the result of the described

first missing step is this:



blue curve: result (LFO Y)
pink curve: from LFO X

Well, this is the result of the feedback from LFO 2 to LFO 1. The only thing left to show now is the result of this new LFO 1 – curve modulating back LFO 2, but, this is – of course – the final result of the whole feedback loop, as I have shown before (well there are some cycles needed to stabilise though). While interpreting the last graph, please don't forget, that an amplitude rise of the modulating pink curve means a positive stretching/lengthening of the cycle of the blue curve, and an amplitude fall means the opposite: the blue curve gets compressed, its cycle starts getting shorter. Rise and fall of the pink (modulating) curve does not at all cause a rise and a fall of the blue (modulated) curve, but elongation and contraction of the blue curve.

All in all, when you look at the overall resulting curve of this two-LFO-Feedback loop, you see: it is far too regular to serve as a CV source for generative developments **on its own**. Changing the frequency relations to high integer numbers (e.g. 0.44 to 0.41) increases the length of the overall cycle, but we need more – and we will get more. Let's just finish talking about LFO networks. And there are some very interesting ones to come.

But after investing so much time in explaining (and after you've made such efforts in understanding) I think it's time for a bit of beauty. So, in the video behind the following link you see and hear a nice way to implement the things you've just learnt.

<https://youtu.be/H40gzzDfpto>

... to be continued