

# **Harmor**

## **Additive**

## **Subtractive**

# **Synthesis Manual**

**From FLStudio 12.4 and onwards**

**“Addictive Attractive Synthesis”**

**February 25, 2017**

## Table of Contents

Foreword .....	11
Chapter 1. Harmor Introduction .....	14
The synths landscape .....	14
Volume .....	14
Harmor Timbres .....	20
Timbre 1 harmonic level .....	22
Harmonic phase .....	22
Timbre 2 harmonic level .....	23
Timbre 1 & 2 mix .....	25
Jean-Baptiste Joseph Fourier .....	26
Timbre blending modes .....	26
Phase randomness .....	28
Chapter 2. Additive/Subtractive synthesis .....	31
Pitch frequency .....	33
Pitch .....	34
Panning .....	34
Harmonic detuning multiplier .....	34
Oct mode versus Hertz mode .....	35
Chapter 3. Unison .....	37
Unison Panning .....	37
Unison pitch thickness .....	38
Unison phase .....	38
Unison distribution .....	41
Unison alternate distribution .....	41
Chapter 4. Subs, clipping, volumes, gain etc .....	43
Sub harmonics 1, 2 and 3 .....	43
Harmonic protection .....	44
Harmonic clipping threshold .....	44
Harmonic clipping mode .....	44
Effects dry / wet mix .....	44
Auto-Gain .....	45
Time domain volume, frequency domain volume .....	45

Velocity switches .....	45
Chapter 5. Basic filtering .....	48
Filter 1 & 2 mix (parallel – serial) .....	48
Filter shapes.....	48
Filter frequency .....	49
Keyboard tracking, key to filter frequency .....	50
Filter width.....	50
Filter resonance amount .....	52
Filter resonance width .....	53
Filter resonance type .....	53
Filter resonance adaptive width / free noise.....	54
Filter resonance offset .....	56
Filter resonance self-oscillation .....	56
Filter oct / Hz switches.....	56
Chapter 6. Blur .....	57
Harmonic blur amount .....	57
Top and bottom tension.....	57
Blur attack and delay .....	57
Chapter 7. Pluck and Phaser .....	59
Pluck.....	59
Pluck amount .....	59
Pluck replaces blur decay.....	60
Timbre blending mode: Pluck .....	60
Phaser.....	61
Phaser type .....	62
Phaser mix .....	63
Phaser width .....	63
Phaser offset .....	63
Phaser offset motion speed .....	64
Key to phaser offset.....	65
Phaser scale .....	66
Chapter 8. Prism .....	67
Prism amount.....	67

Prism add / multiply mode .....	67
Harmonic level to prism amount .....	69
Chapter 9. Harmonizer .....	70
Harmonizer mix.....	70
Harmonizer width.....	70
Harmonizer Shift Step .....	71
Harmonizer Gap Step .....	72
Chapter 10. Tremolo.....	74
Tremolo speed.....	74
Tremolo gap .....	74
Tremolo depth.....	75
Chapter 11. Vibrato .....	76
Pitch vibrato depth.....	76
Pitch vibrato speed .....	76
Chapter 12. Legato and Portamento .....	76
Legato.....	77
Legato mode switch .....	77
Legato time.....	77
Linear / Logarithmic legato .....	78
Fixed / Variable legato time .....	78
Legato limit.....	78
Velocity / Release to Legato time .....	79
Portamento .....	79
Portamento mode switch.....	80
Portamento time .....	80
Linear / Logarithmic portamento .....	81
Fixed / Variable legato time .....	81
Portamento pitch limit.....	81
Velocity / release to portamento time .....	82
Chapter 13. Strum.....	83
Strum time .....	83
Strum mode.....	83
Strum tension .....	84



Chapter 14. Part A/B .....	85
Part selector .....	85
Part enable switches .....	85
Part A/B mix .....	85
Instant replication of the user's changes .....	85
Chapter 15. Distortion.....	86
Waveshaping theory .....	86
Unipolar / bipolar mode in WaveShaper .....	87
Pre amp in WaveShaper .....	87
Mix level in WaveShaper.....	87
Post gain in WaveShaper.....	87
Distortion type .....	87
Distortion amount .....	87
Distortion asymmetry / extra .....	87
Distortion wet volume .....	88
Distortion mix .....	88
Distortion high cut .....	88
Distortion illustration project .....	88
Harmor Classic Distortion.....	90
Harmor Hill Distortion .....	90
Harmor Flat Hat Distortion.....	90
Harmor Cube Distortion .....	90
Harmor Log Distortion .....	91
Harmor Softsat Distortion.....	91
Harmor Rubber Distortion.....	91
Harmor Ribbon Distortion .....	91
Harmor Sincrush Distortion .....	92
Harmor Bitcrush Distortion .....	92
Harmor Bitjam Distortion .....	92
Chapter 16. Chorus.....	93
Fruity Flanger.....	93
Fruity Flangus .....	96
Harmor Chorus.....	98

Chorus presets .....	99
Chapter 17. Delay .....	100
Enable delay .....	100
Delay input volume and delay input panning .....	100
Delay feedback mode: Normal, Inverted and Ping Pong .....	100
Delay feedback level .....	101
Delay time.....	101
Low-cut, High-cut, feedback and feedback damping .....	102
Delay time stereo offset .....	104
Chapter 18. Reverb - Reverberation.....	106
Reverb color.....	106
Reverb input filter: Hight cut and Low cut .....	107
Reverb pre-delay .....	107
Reverb room size .....	108
Reverb decay .....	108
Reverb high damping .....	108
Reverb diffusion .....	108
Reverb wet volume .....	109
Reverb options .....	110
Chapter 19. Compression .....	111
Compression type.....	111
Compression amount.....	112
Compression band: Low, mid, and high.....	112
Chapter 20. Line editor .....	113
Point modes.....	113
Basic line operations.....	113
Line segment types.....	114
Arpeggiator points .....	114
Chapter 21. Target basics.....	115
Editor Target drop-down menu .....	116
Combining mode: Unipolar product.....	117
Combining mode: Bipolar product.....	118
Combining mode: Bipolar sum .....	118

Articulator part: Envelope.....	118
Envelope window option menu: Make all loop.....	119
Envelope window option menu: Insert points at markers.....	119
Envelope window option menu: Flip vertically.....	119
Envelope window option menu: Scale levels... ..	120
Envelope window option menu: Normalize levels.....	120
Envelope window option menu: Decimate points.....	120
Envelope window option menu: Filter.....	120
Envelope window option menu: Smooth up.....	121
Envelope window option menu: Smooth up abrupt changes .....	122
Envelope window option menu: Turn all points smooth.....	122
Envelope window options menu: Create sequence.....	123
Articulator part: Low Frequency Oscillator, LFO.....	126
Articulator part: Keyboard mapping.....	127
Articulator part: Velocity mapping.....	127
Articulator part: Modulation X, Y and Z mapping.....	127
Articulator part: Random mapping.....	127
Articulator part: Unison index mapping .....	128
Articulator part: Held index mapping .....	128
Articulator part: Mod X speed mapping.....	128
Articulator output smoothing.....	128
Chapter 22. Image synthesis / resynthesis .....	129
View mode: Plane(s) to work on .....	130
Image fine speed.....	130
Image course speed .....	131
Image time offset .....	131
Image time offset smoothing .....	131
Looping mode .....	131
Image sharpening.....	132
Level section.....	132
Image gain pixel scale .....	132
Image gain mix .....	132
Frequency section.....	132

Image frequency pixel scale .....	133
Image frequency mode .....	133
Image formant shift .....	133
Image formant shift mix .....	134
Image copy, Image paste .....	134
Image frequency interpolation curve and Image gain interpolation curve .....	135
Display scale .....	137
ADV tab image / resynthesis settings: quality mode .....	137
ADV tab image / resynthesis settings: denoise .....	137
ADV tab image / resynthesis settings: Side processing / invert right channel - stereo .....	137
Image options menu: Basics .....	138
Image options menu: Map audio markers / regions to keys .....	138
Image options menu: Prepare time envelope for drum loop .....	139
Image options menu: Prepare image time envelope .....	140
Image options menu: Create bouncing loop time envelope .....	140
Image options menu: Slow down within loop region .....	141
Image options menu: Tune pitch for time offset .....	141
Image options menu: Set viewed zone as loop region .....	141
Image options menu: Set last played zone as loop region .....	142
Image options menu: Map time offset to last hit key .....	142
Image options menu: Generate random cloud .....	143
Image options menu: Degrade horizontally .....	143
Image options menu: Degrade horizontally .....	143
Image options menu: Crossfade 10%, 25% and 50% .....	143
Image options menu: Crop top .....	143
Image options menu: Invert .....	144
Image options menu: Flip vertically .....	144
Clipboard monitoring .....	144
Edit image .....	144
Image selected column .....	144
Image freq local mask .....	145
Chapter 23, Advanced targets .....	146
Filter envelope amount .....	146

Global envelope ADSR settings.....	147
Global envelope LFO settings.....	147
Global EQ.....	148
Local EQ.....	149
Harmonic randomness .....	150
Harmonic clipping .....	150
Harmonic prism.....	151
Harmonic unison pitch .....	151
Image formant mutation.....	152
Filter shape .....	153
Resonance shape.....	153
Pluck shape.....	153
Phaser shape.....	154
Harmonizer distribution .....	154
Filter mask .....	155
Filter resonance mask .....	156
Phaser mask.....	157
Harmonizer local mask.....	157
Blur local mask.....	157
Chapter 24, ADV Tab.....	158
Predelay .....	158
Envelope release polyphonic scale.....	158
Ramping time.....	159
Declick.....	160
Unit Order .....	161
Effects order .....	162
Auto-gain mode.....	163
Invert arp .....	163
Smooth mod .....	163
Monophonic .....	163
Randomness .....	164
Performance versus Quality.....	164
Chapter 25, Global menu.....	166

Chapter 26, Presets .....	167
The default preset sections.....	168
How to decompose presets .....	169
Template, tutorials, and tricks presets.....	174
Create your own presets.....	175
Create presets for others.....	177
Chapter 27, Examples .....	183
Make white noise.....	183
Chapter 28, Open questions.....	185

## Foreword

I had hoped that someone had written this book for me, but no one did. So here I am, writing it myself. It is difficult and it takes much time. Being a developer myself I totally understand how frustrating the developers at Image-Line can feel when they must write a manual. It feels like a waste of time. Probably it is a waste of time. Actually, I would rather have them create new features for me than waste their time. So here I am doing this in my spare time.

The idea has struck me that it is not the people who knows everything about a subject that should write books about it. It is the people that that can sympathize with the beginner that should write the book. So, this is me. Now when I write this introduction to this book I am indeed totally clueless about Harmor and I decided that I will write the book that takes the reader and myself all the way through Harmor.

My approach to Harmor is breadth first. First, we go over Harmor skipping most of the drop-down menus and envelopes. In the second round, we open and discuss all drop-down menus.

... seven months later...

People tell me that I should sell this book but I don't think they understood why I made this book so I will explain that here. I made this book because I am enthusiastic about Harmor and how things are working in Harmor and I like to convey this knowledge to others. Selling books has nothing to do with sound design. Besides, if I were to sell this as a book then perhaps I had to do marketing and the price itself of the book would be a barrier for people to read this. I don't need that and I already got a daytime job as a software developer at one of the fastest growing translation companies in the world so I am fine.

When I got all that explained people don't give up but they tell me that I should tell Image-Line to give me money for writing this book but I don't think they understood my relation to Image-Line so let me explain that here. I have nothing to do with Image-Line except that I bought their product. Of course, I don't want them to be irritated on me but I like to be able to be myself and when I feel that a feature is not good enough then I will write that here. If Image-Line starts paying me then things gets unnecessary complicated.

If you still want to send money for this book I suggest that you instead donate a little sum to a proper rheumatology charity in your country and that you tweet about your deed with the hashtag #JensHarmorBook. I would be delighted, especially since I suffer from a bit of arthritis myself.

If you wish to print this book, then it should be printed by yourself for your personal use. You may not publish this book in parts or in any electronic form. You may download and view this electronically for your own personal use. This book is fully protected by copyright laws.

Here is the link to this book: <https://www.malmgren.nl/Harmor.pdf>







## Chapter 1. Harmor Introduction

This tutorial covers the creation of synths using the FLStudio Harmor plugin from within FLStudio. The version used here is Harmor from FLStudio 12.3 and onwards.

So, what is Harmor? It is a flagship synth plugin of Image-Line for use in FLStudio or as a VST in other DAWs. It was released in September 2011.

My explanation of Harmor starts in the upper left corner of the user interface. The sound produced in Harmor is controlled with the timbre section at the top left corner of Harmor. We could go straight into how Harmor produce sound but before we do that we will take a detour over a couple of other simpler synths.

### The synths landscape

If we look at the “entry level” oscillation based synths in FLStudio we find for example:

- 3xOsc, it has 3 oscillators.
- Wasp, it has 2 + 1 oscillator. 2 low frequency oscillators. 6 filters.
- Wasp XT, it also has 2 + 1 oscillator. 2 low frequency oscillators. 6 filters. Mod envelopes.

These synths feature different variations of sound production with help of a couple of oscillators. The sound from the oscillators can be mixed and the result can be filtered with or without help of LFO - a low frequency oscillator. These synths contain regular filters such as high pass, low pass etc. What LFO is as well as high and low pass filtering will be covered later in this book.

The logical step upwards from 3 oscillators is something with 6 oscillators and one thousand knobs. Yes, that is Sytrus! Inside Sytrus the oscillator controls are called operators because they are doing so much more each of them that it would not be fair to call them just oscillators.

### Volume

Common waveforms in these synths are usually named like “Sinus”, “Saw”, “Triangle” and “Square”. The name reflects how the waveform looks like in an oscilloscope.

The louder a sound is played the more volume it has. This is the same as saying that the amplitude of the waveform is higher. Unfortunately, I have not really found any good antonym for amplitude so instead I talk about making sound quiet or lowering the amplitude. In Harmor it is simple though, you just turn the volume knob down.



You could be fooled to believe that oscillators are everything and that the oscilloscope is the best way to analyze sound in the synth creation world – but that is not the case. Take for example the 6 operators in Sytrus. They are clever things that you can reroute in amazing patterns. You can even gradually morph between all the standard waveforms in Sytrus.



Image 1. The timbre windows.

So how many oscillators got Harmor? Clearly you can see two timbre windows! It is true that they influence the sound in Harmor. If you are new to Harmor or sound design I suggest that you ignore timbre 2 to begin with, we will come back to timbre 2 later.

In Sytrus you can right click the waveform and select “Convert to sine harmonics” to open “the interior workings” of the waveform formula. When doing this, it is possible to let every one of the operators of Sytrus produce any sound. Now please consider that if **one** of the operators in Sytrus can be **that** powerful then why do we need 5 more of them? Is that not a little bit too much?

Exactly what the timbre is and how it works is something you could easily get lost in as a beginner so I will explain it in small digestible chunks.

We are going to explain the oscillator in Harmor with a practical thoughts experiment. Sound can be explained in many ways. The oscilloscope is not the best way of understanding sound and yet this is the first view method we get confronted with in many synths. The oscilloscope can explain how a cone of a speaker is supposed to move in or out but there it ends.

FLStudio comes with a Wave Candy tool that can visualize sounds in various ways. Here is a link to an FLStudio project set up to analyze various synths with Wave Candy:

<https://www.malmgren.nl/Harmor/Harmor-01.flp>.

First of all, I will show how a saw waveform looks like in 3xOsc.



Image 2. Playing the note C5 with 3xOsc. The blue Wave Candy window displays the oscilloscope image. To the right there is a spectrogram image and at the bottom Fruity parametric EQ 2.

The saw wave form produced by 3xOsc is not especially “clean”. There are plenty of artifacts. The entire frequency spectrum is produced all the way up to about 18kHz. Next we will have a look at saw waveform produced with Wasp.



Image 3. Saw waveform C5 note saw waveform played with Wasp. The spectrum is much cleaner than 3xOsc but the frequencies above 18kHz is fading out. The 3xOsc has an “N” shaped waveform and wasp the other way around but it still a saw.

And then we look at a saw waveform produced by Sytrus.



Image 4. Here is a C5 note plying saw waveform produced with Sytrus.

Sytrus produce a clean waveform and it uses the spectrum all the way up to 20 kHz.

I prefer the spectrum diagram in Wave Candy to analyze sound. The spectrum diagram is much more informative than the oscilloscope. If we play the standard waveforms and analyze them with Wave Candy it can look like this, see Image 5.



Image 5. The classic waveforms produced with Sytrus playing a C5 note. This is produced with this FLP <https://www.malmgren.nl/Harmor/Harmor-02.flp>.

We notice that

- The sine waveform has one line at the bottom. It is the fundamental frequency or the fundamental tone. The rest of the spectrum is silent.
- The triangle waveform has one loud frequency at the bottom and then much less loud frequencies over the rest of the spectrum.
- The square waveform has the same number of active frequencies compared to the triangle waveform but all the frequencies has a higher amplitude. They are louder.
- The saw waveform has the double active frequencies compared to the square waveform.

Of these waveforms, we can see that the sine curve is not so interesting to work with when it comes to sound design in a synthesizer because there is just one frequency doing anything. The triangle is better but the two best basic waveform types that is the saw and the square waveforms. They have plenty of frequencies to work with and I think this is the reason why



they got such a prominent place in Harmor, namely the default setting of Timbre 1 is a saw and Timbre 2 is a square.

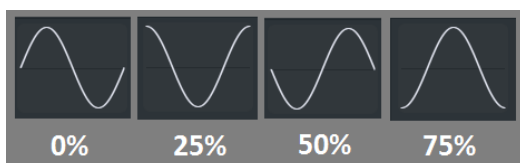
Each of these frequencies in the spectrum diagram are themselves sine curves, only. We can show that by filtering everything except one frequency.



Image 6. We filter out all sound except the frequencies at 1577 kHz. What wave form is that producing? This is illustrated in <https://www.malmgren.nl/Harmor/Harmor-03.flp>. Here we placed the Fruity parametric EQ2 before the Wave Candy so that the only one frequency is displayed. There is some leakage of sounds on other frequencies but it still works quite well.

To create a specific waveform there are three components needed: Specific frequencies of sine waveforms and amplitude. But that is not all, also the starting point in the oscillation for each frequency is something that also contributes to the characteristics of the sound. The starting point is called the phase.

It had been nice if we could see the phase of the oscillation for each frequency but there is no tool for that in FLStudio Wave Candy. The phase is also the most elusive component of the creation of a waveform. Another way of illustrating the phase can be done by producing a waveform in Sytrus where we can see the phase and then play around with that. We will try this in a moment but first let's have a look at various phases of a sine curve.



The phase is not only elusive it also has a “linguistic confusion” to it as well. In help texts phase is described as the starting point of the oscillation but what is that? In Sytrus the phase is given as a percentage of the whole cycle. You

could believe that by introducing phase we introduce silence before we play the waveform. This could be true for recorded sound mixed together with other sounds. In Harmor and Sytrus and most synths the phase is defined to be the (offset) point into the oscillation waveform that the oscillator will **start** play from.

Harmors timbre 1 window displays the oscilloscope with the current waveform. For a moment, we don't care about timbre 2. The waveform of timbre 1 is produced with help of the combination of timbre levels and the harmonic phase data. The timbre levels are all about how many of the oscillators should play and at what amplitude - how loud, each oscillator should play. There are 516 partial oscillators to play with in Harmor, and they all play only sine waves. A specific waveform can use several of these oscillators but sometimes not all of them are needed. Image 7 shows how to see how many partials are active.

If you for example play a sine waveform, then only one is at work. The harmonic phase data is the individual phase of each of the oscillators at work. In total, we have three elements that make up a single waveform in Harmor:

- The number of active partials and the frequency of them.
- The individual amplitude of each of the working partial sine waves.
- The phase position of each working active partial.

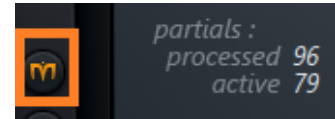


Image 7. At the left edge of Harmor the About Harmor knob. In the window that appears you can see how many

All these three elements combined in a clever way can create any waveform imaginable. If you missed the previous sentence, then you could go back and read it again and again because it is important. Thanks.

Look how Sytrus presents a triangle waveform and the sine harmonics of it:

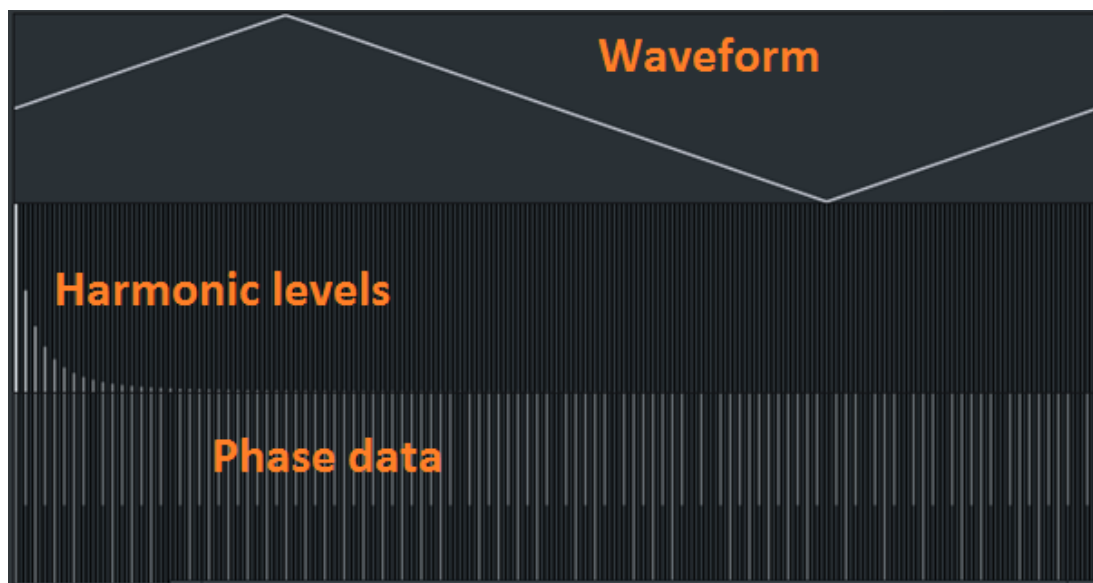


Image 8. A triangle waveform in Sytrus.

Harmonic phase data without any active partial is not contributing anything to the waveform.

### Harmor Timbres

The timbre 1 window has a built-in filter. It is called a "Brownian noise filter". It is reducing the amplitude gradually for higher frequencies. When working inside Harmor it is just there



and you cannot really “see it”. It is when we start comparing Harmor with other synths that we will notice the Brownian noise filter but we will come to that in a moment. This filter is there because higher frequencies comes with much higher energy and the Brownian filter compensates for that. If you look at Image 5 (the image with the spectral analysis of the four standard waveforms) you notice that the higher frequencies are denser than the lower frequencies. This makes it easier to create traditional good sounding waveforms. For normal sound design, it is not necessary to make the high frequencies extremely loud so we get them automatically reduced with this built-in filter.

Sytrus got no built-in Brownian filter. If we create a waveform in Sytrus with the sine harmonics window it can look like Image 9 here below.



Image 9. Here is a waveform created inside Sytrus in the sine harmonics editor.

The upper part is the timbre levels, the amplitude of each individual sine. The bottom half is the phase data of each sine. Please notice that there is a little bug in this version of Sytrus so that the phase is hidden below the scrollbar.

I made stairs in the phase data of Image 9. The first step is 10%, and then 20% all the way to 70%. Then above 74% the phase is displayed below the scrollbar unfortunately.

Sytrus got 255 partials in total. When exporting a single-cycle waveform from Sytrus and then importing it into Harmor only the first 255 partials will be set by the import in Harmor because it has 516 partials. Another thing we will notice is that Harmor shifts the phase. The result is that we get the same waveform shape but shifted one half cycle. The outer ends of the waveform in Sytrus will be the new middle of the waveform in Harmor.



Image 10. Waveform imported into Harmor. To open the harmonic levels window click on the diagram.

### Timbre 1 harmonic level

When you click the timbre 1 window then at the bottom of Harmor you can see the harmonic level data of the imported waveform.

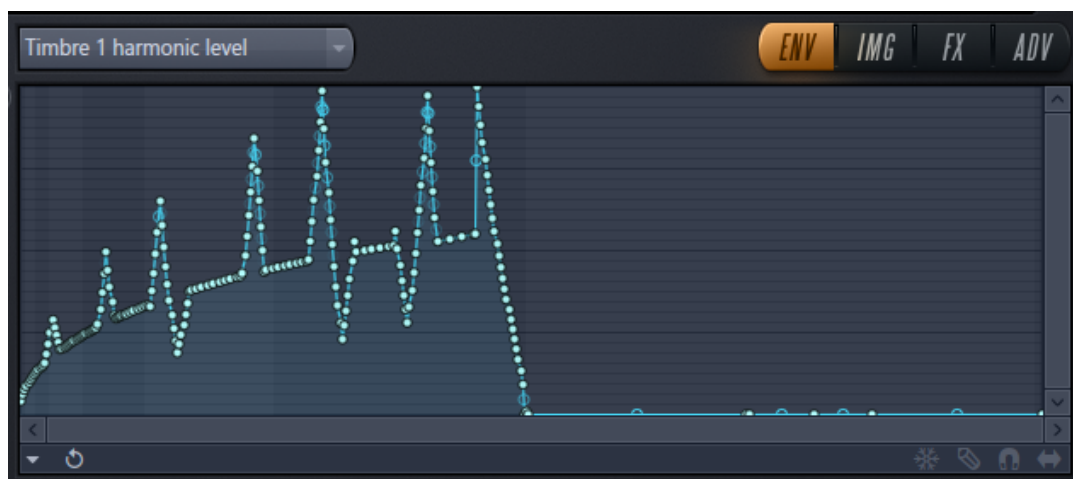


Image 11. Harmonic levels data of the imported waveform.

Here you can see that the levels of the Brownian filter have been compensated for in the import. In Sytrus we had a flat line with some triangular shaped bumps and cavities. In Harmor the line goes up and up. At 255 partials, it ends. The triangular shapes are pointier because of the Brownian filter.

### Harmonic phase



Image 12. Harmonic phase window. The phase is shifted a half cycle. To open the harmonic phase window select that in the drop-down menu above the envelope window.

The harmonic phase window is phase shifted. Here you can see the stairs in the phase starting at 10%. In Harmor the tooltip says +0.10. At +0.50 Harmor jumps down to -0.50 and then continues up until -0.09. That is the part of the diagram that Sytrus had hidden under the scrollbar. For the rest of the diagram above partial 255 Harmor sets the phase to +0.24?! Whenever a partial was not used in Sytrus, then after import in Harmor it is set to +0.24 or -0.25. I think this is “feature” because in Sytrus if the harmonic level is zero then the phase is not displayed. Internally it is set to a random value. In Harmor it becomes visible.

Here below is an experiment that will help us understand the timbre of Harmor. This curve here below is created in Sytrus. It looks like a “W”.



Image 13. Here is a "W" waveform created with Sytrus.

I exported this single-cycle waveform from Sytrus and imported it into Harmor. The upper part will be imported into the harmonic data of timbre 1. The lower part is imported into the harmonic phase. During import the data has been compensated for the Brownian filter.

### Timbre 2 harmonic level

Up until now I have not talked about timbre 2. Now we have all the knowledge in place to start talk about timbre 2!

Timbre 2 has its own harmonic data but it shares the harmonic phase data together with timbre 1. When we analyze a waveform by right clicking timbre 1 and selecting “Analyze

single-cycle waveform...” then the phase data is imported as well and it is shared with timbre 2.

The default harmonic levels for timbre 2 in Harmor looks like a sine curve.



Image 14. The default harmonic levels for Timbre 2. Every even harmonic is turned off.

As you recall the Brownian filter removes more and more of the amplitude for higher frequencies. If we were to reverse engineer timbre 2 inside Sytrus it would look like something here below.



Image 15. Here we reverse engineer the timbre 2 window of Harmor from within Sytrus.

As you can see I made the harmonic magnitude (as it is called in Sytrus) to decline approximately as I think that the Brownian filter in Harmor is working. The phase is identical to the “W” waveform.

Here below are the two timbre windows in Harmor! As you can see I am close.



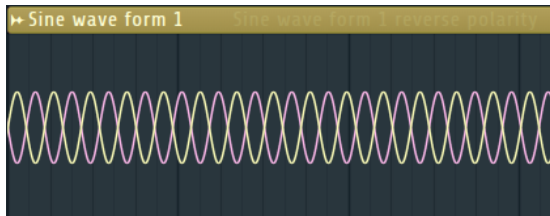
In the sample project <https://www.malmgren.nl/Harmor/Harmor-04.flp> you find the “W” waveform in Sytrus at OP3 that I exported into a single-cycle waveform. In OP4 you find the simulation of how I thought that Harmor would create timbre 2.

The “W” shape sounds a bit like a church organ!

### Timbre 1 & 2 mix

The mix between the timbres is done with the knob between the two timbre windows. All the way to the left is 0% and only timbre 1 can be heard. In the middle, we have 50% and you can hear both timbre 1 and 2. All the way to the right is 100% and only timbre 2 can be heard. The reason to let the two timbre windows share the same phase is that this way you can mix between two different sounds without the need to care about phase cancellation. Although this is fantastically clever, this with two timbre windows sharing the same phase is hard to explain. What is phase cancellation? This phenomenon is best explained with help of the oscilloscope.

Normally when mixing sounds, you will come across a phenomenon called phase cancellation.



A regular waveform, for example a sine wave, has a positive and negative value.

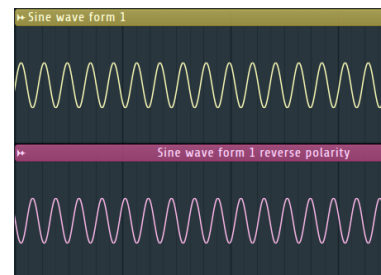


Image 16. Sine waveform 1 and sine waveform with reverse polarity.

When we send such sound to a speaker the speaker cone moves out and in. When we mix together two identical waveforms of opposite phase they are cancelling each other completely. When one sound is moving the speaker cone out, the other sound is doing the exact opposite motion so that the result is a speaker cone that is not vibrating at all.

In Image 17 we are mixing together two waveforms. The yellow is the first sine waveform and the

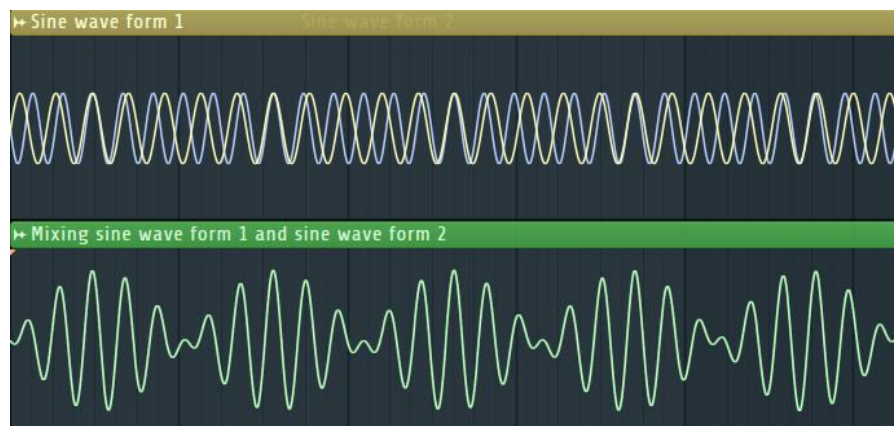


Image 17. Mixing two sinewaves of different frequency.

blue is second sine waveform. When they cancel each other, there is almost silence and when they match the sound doubles in amplitude. The green waveform is the result of the two. When the phase cancellation is causing this, it is called a rhythmic cancellation.

### Jean-Baptiste Joseph Fourier

It was 1807 that Mr Fourier presented the proof of that any waveform could be decomposed in a series of simpler sine waveforms, partials. Then 204 years later, 2011, Image-Line released Harmor, based on this idea, additive synthesis.

### Timbre blending modes

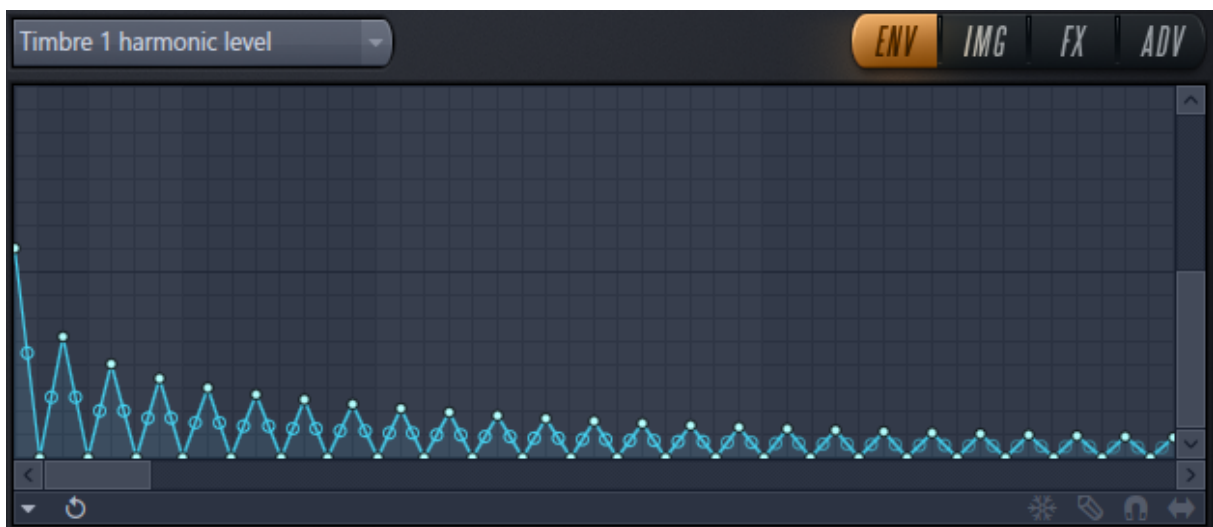
Below the timbre windows you find the blend modes. The default blend mode is fade. It is doing what we explained up until now in this book.

The blend modes are doing arithmetic operations on the harmonic levels of timbre 1 and timbre 2.

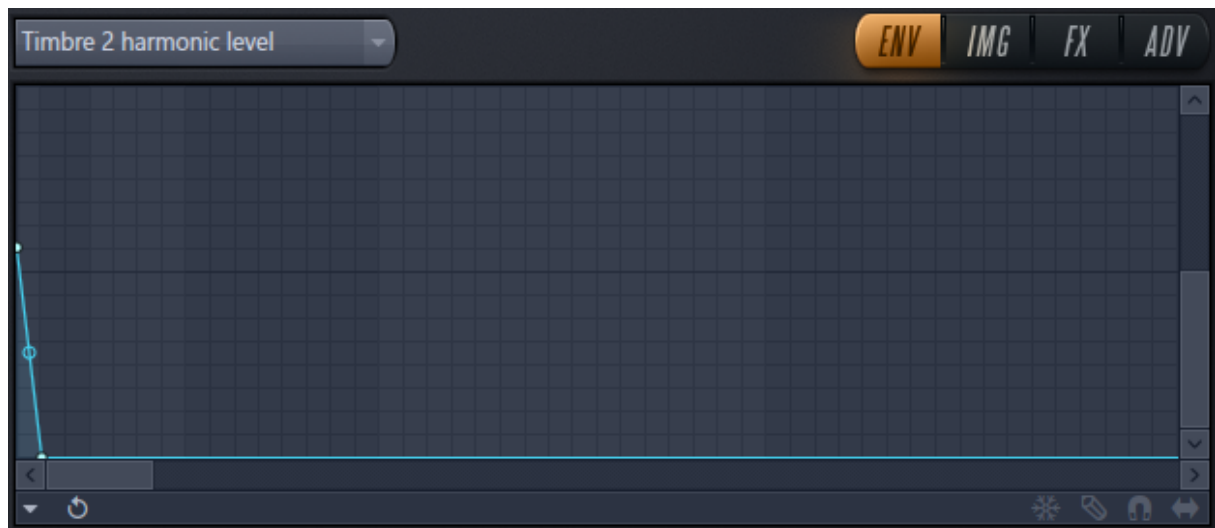
If you set blend mode to fade and then analyze a triangle waveform in the left window, timbre 1. In the right window, timbre 2, analyze a sine curve.



This is how the harmonic levels of the triangle wave looks like.



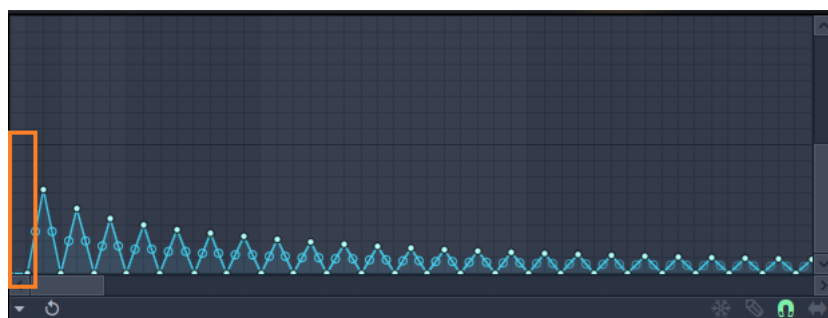
Here is timbre 2, the sine waveform.



This is what we get if we set blend mode to subtract. Timbre 2 is replaced by the result of the subtract operation. Inside Harmor the old Timbre 2 is still there it is just that we see the result of the Subtract operation in the Timbre 2 window.



What could make this? It is a triangle waveform missing its first point. If we go into the harmonic levels of timbre 1 and do that manually we get this result.



Now we have identical graphs so this looks like a proof but what did just happen because if we just changed timbre 1 then why would not the timbre 2 window change to something new? It should do with that the sine curve harmonic data is +1 while our triangle harmonic data point one is 0 and arithmetically subtracting 1 from 0 would yield -1 but there is no negative amplitude here so the result is identical.



And another thing, how could the sine wave display so nicely on top of the phase data of a triangle waveform? This is because the sine wave only has the first harmonic active. All others are zero so for those the phase data has no influence.

In all my experiments, I can only see that timbre 2 is subtracted from timbre 1. The documentation says the opposite.

The multiplication mode has a behavior that we perhaps can explain easier. For example, multiplying with 0 gives 0. Multiplying with a value less than 1 gives a smaller value. I think it is strictly timbre 2 multiplied by timbre 1 but it does not matter so much with multiplication.

Here are the options of the timbre blending modes drop-down menu:

- Fade - Standard Timbre 1 to Timbre 2 fading.
- Subtract – The harmonic levels of timbre 2 is subtracted from the harmonics levels of timbre 1.
- Multiply - In this mode the harmonic levels of timbre 2 is multiplied with timbre 1. If any has zero level timbre levels, then the result is zero.
- Maximum - The maximum harmonic level is selected from Timbre 1 and 2.
- Minimum - The minimum harmonic level is selected from Timbre 1 and 2.
- Pluck – blending of Timbre 1 and Timbre 2. The graph in the timbre windows represent the waveform as it looks at the very beginning of playing the note. That could we interpret as if Pluck blending just copy Timbre 1 into Timbre 2 but there is more going on here. We will explore this mode further when we come to the pluck functionality in *Chapter 7. Pluck and Phaser* and there I will explain what this mode does.

## Phase randomness

In the upper left corner, you find the phase control.



Image 18. The phase controls.

The start button sets the global phase of the waveforms to a value from 0 to 359 degrees. This is a starting phase for all partials. Like a global start value. The default value for the starting phase is 180 degrees. When using image synthesis this button jumps automatically to 0 – I don't know why. For the rest, I cannot see it is used so much.

The randomization is the same for both timbres. It is not so that the different timbres have different random phase value (although that had been interesting as well).



The rand button is interesting. When turning it to the left each partial phase are individually randomized. That means that the shape of the waveform is changed and that is a logical consequence of that phase is an ingredient making up the waveform. The more the button is turned to the left the bigger the variations of each random value.

When turning the button to the right the partials are randomized in a free running mode. In this mode, the starting phase of the waveform is set to a random value. The more the button is turned to the right the bigger the variation of the random value.

In the project <https://www.malmgren.nl/Harmor/Harmor-06.flp> we look in depth what is happening with phase and phase randomness.

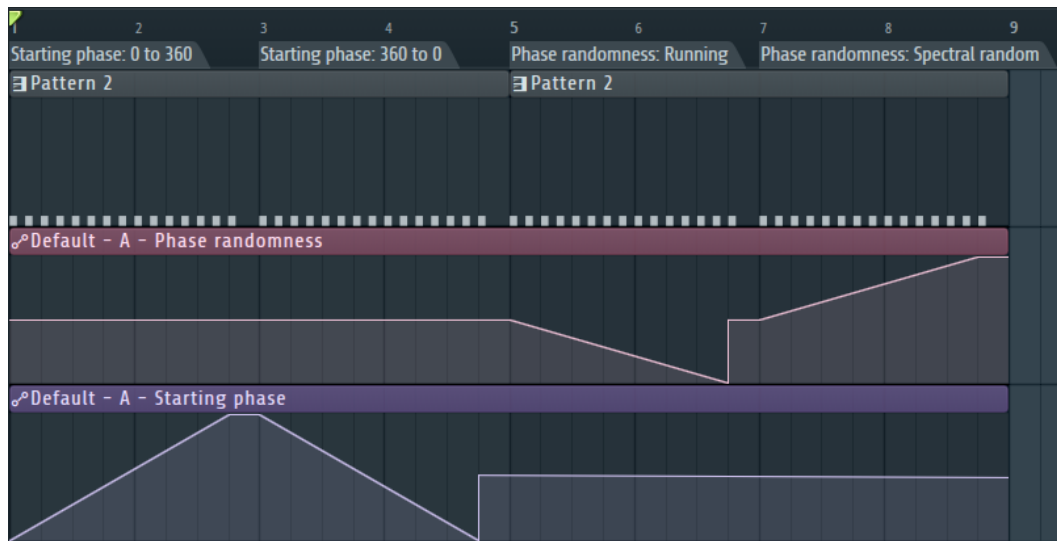


Image 19. Illustration of what phase and phase randomness gives to a saw waveform. Pattern 2 plays a note of C4 of Harmor plain and simple. Then I connected the two knobs Phase and Phase Randomness to automation clips moving the knobs while playing the “song”. Now we can record this and analyze the result.

This is the result in Edison when zoomed out.

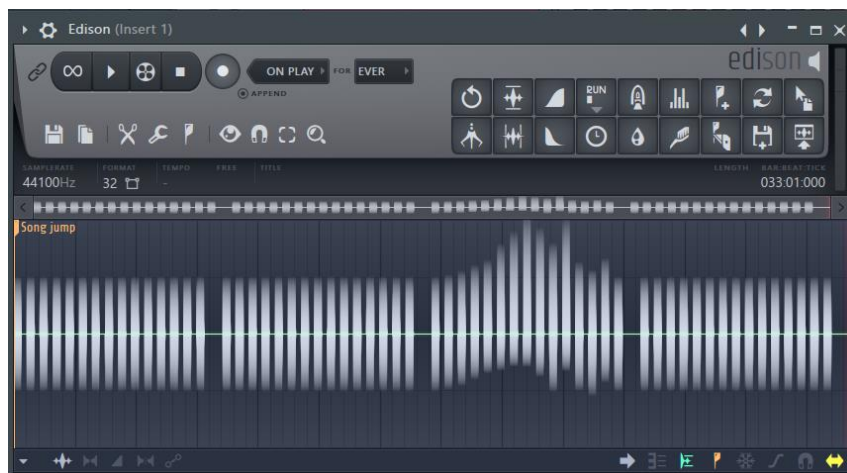
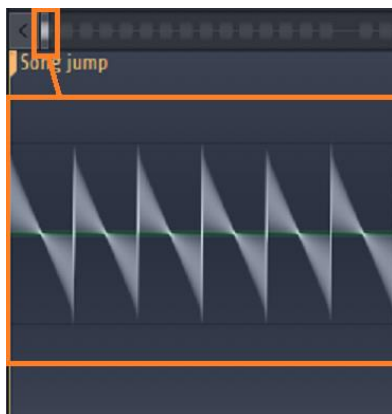
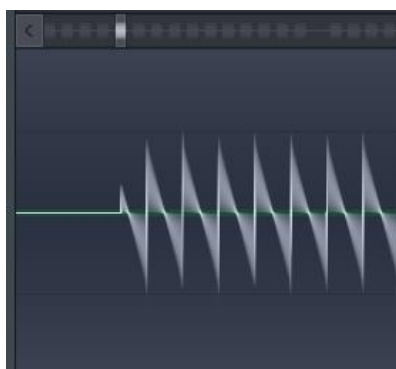


Image 20. The project recorded with edison.

Here we zoom in to the beginning of the sound.



Here is the saw starting at the beginning as we are used to. When scrolling forward we can see how the phase changes.



In the section 0 to 360 the beginning of the waveform is skipped more and more. In the section 360 to 0 the reverse happens.

The next section is phase randomness – running. It destroys the waveform.



The last section is spectral random. What we see is that the further the randomness is increased the more the starting phase jumps back and forth between each note.

## Chapter 2. Additive/Subtractive synthesis

Now we have come to the point that we learned so much that I can explain what makes Harmor unique compared to other synthesizers.

In a regular synthesizer, the sound is generated somehow. It can be with oscillators or wavetables and then after that usually a large section of the synthesizer is normally devoted into manipulating the sound by removing frequencies and sounds with help of various filters. Hence the term subtractive synthesis. The removal of sounds usually takes place on the audio signal. When processing the audio signal in step after step it is possible that each step introduces various artifacts. Some artifacts we want and some are not wanted. Probably some of the artifacts are so common that we don't pay attention to them.

The sounds in Harmor on the other hand is put together with a maximum of 516 different partials: Their amplitude, frequency and phase makes up the final sound. Instead of mixing the sounds and then doing the rest of the processing as other traditional synthesizers Harmor has a highly configurable filter processing workflow that works on the partials before the sound is assembled into a final audio signal. This makes it possible to process the sound in a lossless way that is entirely new and unique to modern digital sound design.



Slightly over-generalized you could think of the upper part that it is devoted to the “lossless” workflow of the partials. The sound is created and gets much of its characteristics in the upper part. Then it is assembled to a regular audio signal and is further processed with help of the settings in the lower part, especially the effects. Even the lower section has partial manipulation features.

Harmor is working on the sound in the form of partials. Image-Line recreated many of the classic filters to work with the partials as well. This fusion of both worlds they called additive/subtractive synthesis. If you need to say this very quick and often it is better to say additive/attractive synthesis, don't you think?



At the end of the lossless filter workflow it is possible to visualize the workings of Harmor. On the right side of the user interface there is button next to an eye symbol. When you click that button the visual feedback window displays. Click on it two times to make it maximum size. The visual feedback window gives a lot of information about what is going on inside Harmor.

There is a significant difference between the built-in feedback window in Harmor and what we see outside Harmor when we analyze the signal with our regular set of WaveShaper windows.



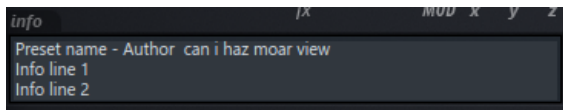
Image 21. Here I play two notes with the default saw waveform to show various feedback windows.

In Image 21 you see various feedback monitors. The master monitor, A, in the toolbar displays a small oscilloscope and a peak meter. The oscilloscope is like the bigger blue WaveCandy window, F, also set to Oscilloscope display. The red WaveCandy window, E, is set to a vector scope. This is a good tool to analyze the stereo signal. The vector scope, D, is a spectrum meter and this is like the Fruity parametric EQ 2, C, at the upper right corner and the visual feedback inside Harmor, B.

In the sample project <https://www.malmgren.nl/Harmor/Harmor-05.flp> I set up these displays as shown in Image 21.

The visual feedback inside Harmor happily ignores that I play two notes at the same time. It just displays the analysis of one note in mono.

When hovering over the built-in feedback monitor the tooltip window displays the frequency that the cursor is hovering over.



If you would like to have the visual feedback extra big then you can write the text “can i haz moar view” in the tooltip window.

## Pitch frequency

There are three ways to adjust the pitch when playing notes with Harmor. The first place to adjust the pitch is determined by the root note setting. This is a setting available for most synthesizers in FLStudio under the wrench icon.



Image 22. Miscellaneous functions tab. Right click a key in the piano to set the root note.

Setting the root note to C4 makes Harmor produce the notes you expect so that when you press C5 on the keyboard or send that note to Harmor via a note in the piano roll then Harmor also produce a note C5. If you set the root note to C5 then Harmor produce a pitch that is one octave higher than the played note. For now, consider using C4 as a root note.

The second place to adjust the pitch is inside Harmor at the Pitch label.



Image 23. Pitch frequency inside Harmor.



The frequency control makes it possible to adjust the pitch relative to the root note. The frequency control displays a fraction. When the nominator, N, is increased to 2, Harmor produce a note that is one octave higher than normal. When increasing to 4 Harmor produce a note 3 octaves higher than normal and so on.

When the nominator is set to one and the denominator, D, is increased then the produced note is lower than normal compared to the root note.

## Pitch

Below the pitch label you find two radio buttons oct and Hz. Hertz is the regular way of producing sound. All we learned so far about how partials are additively building waveforms works as you expected in this mode. If we had a saw waveform and then we turn on the oct mode all the partials are evenly distributed from the root note and upwards. The sound is no longer that of a typical saw waveform. It is only the sinus waveform that sounds the same while switching between oct and Hz mode.

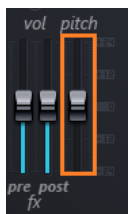


Image 24. Main pitch slider.

The third place to adjust the pitch is the main pitch slider. It has the range of -2400 cents to +2400 cents. That is two octaves up or down.

## Panning



At the top of the window you find the panning knob. This sets the panning of the output of Harmor.

## Harmonic detuning multiplier

The detuning control is also a fraction with a nominator and a denominator part. The default setting is 1/1 and that gives no detune. When increasing the nominator then the fundamental frequency stays where it should be but the rest of the partials are moving away from the fundamental frequency. In Image 25 you can see how a saw waveform playing a C4 note how “reacts” to detuning.

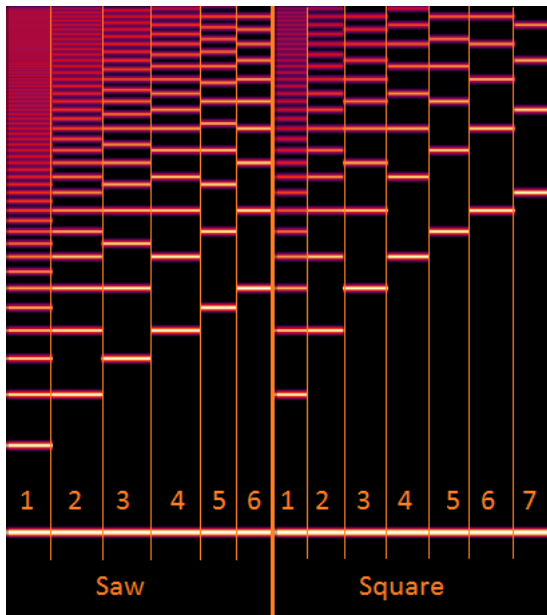
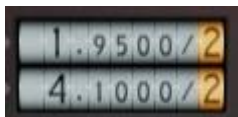


Image 25. Detuned saw and square. The horizontal line at the bottom is the fundamental frequency. Numbers indicate the nominator value.

When increasing the detune denominator the frequencies are moving towards the fundamental frequency.



Look in this video by SeamlessR if you would like to see an advanced application of detuning to create interesting phase cancellation:

<https://youtu.be/-JUnNTFSekE>. Every time when SeamlessR talks about Harmor he says it has 512 partials. Why is that? Harmor manual says it has 516?

### Oct mode versus Hertz mode

In the Pitch section of Harmor there is also two buttons oct and Hz. What is that?



Throughout Harmor in various places you find the option of switching between oct mode and Hz mode. In most places the default setting is oct mode. These two modes refer to the distribution of frequencies of partials.

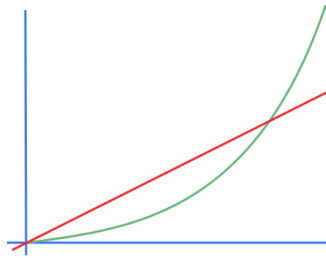


Image 26. Logarithmic versus linear. The curved line is logarithmic and the straight line is linear.

In general, oct mode refers to the frequency scale as we are used to. In oct mode partials are organized logarithmically. If you recall the images until now in this book you have seen that for example a saw wave has many more higher frequencies compared to lower frequencies.

In Hertz mode, the partials are organized linearly, the frequencies are evenly distributed along the spectrum. I find that Hertz mode often creates “outer space” results while oct mode is down to earth and intuitive.

Many times, oct mode is the normal setting of that part of Harmor and Hertz the alternative method. Many times, (but not always)

Hertz mode makes the frequencies more wide-spread.



## Chapter 3. Unison

Unison is an effect that is making the sound – sound thicker. This works very much the same way as with traditional instruments or singers. For example, violinists playing in an orchestra or singers singing in a quire. Often the idea is to play or sing the same notes but just slightly different. This also what Unison is doing. In a computer synth, this is even more important because computers are predictable by nature unless we deliberately introduce some randomness or difference and that is possible with Unison.



Image 27. Unison

The Unison effect is created by generating several copies of the synth and then introducing differences between each sub voice and playing these concurrently. The controls for Unison are placed right in the middle

of the user interface of Harmor.

When there is no number in the field besides the label order then Unison is turned off. You can increase the number in this field. It can be set to a value between 2 and 9 indicating how many sub voice copies will be made. You can try this by hovering of the square and see an up-down cursor and then click and drag to change the number.

The built in visual feedback window will not display anything different when changing the Unison settings of Harmor. Outside Harmor there are more going on in various displays.

Unison introduces several differences between the voices with various methods: Panning, Pitch and Phase.

### Unison Panning

This gives each sub voice its own panning setting from left to right. The slider can be set to a value of 0 to 100%. The result of this effect is visible in the vector scope monitor.

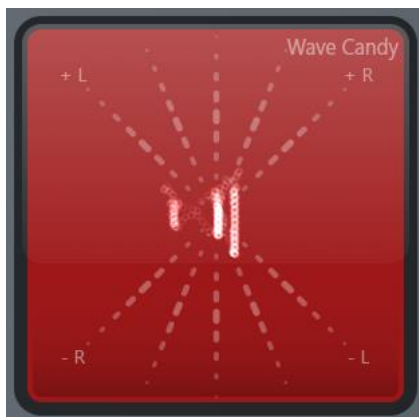


Image 29. Saw tooth waveform with unison set to order 4 and panning to 75%.

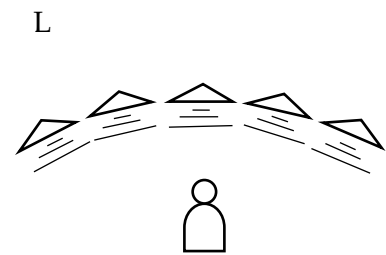


Image 28. Unison of order 5 you can see like five different Harmor patches playing like an orchestra.

When increasing the panning slider, the white lines are spreading out further, the stereo gets wider. When decreasing, they are getting closer to the middle the sound gets more mono.

### Unison pitch thickness

It is possible to introduce a variation of pitch between the different sub voices with the unison pitch thickness slider. This slider can be set to a value between 0 and 100%. At 100% the difference in pitch is so big it sounds detuned.

### Unison phase

The phase is at what point into the oscillation that the oscillation of each sub voice should be played. It is possible to offset the phase of each sub voice with two different modes: When moving the slider up you indicate how much each sub voice phase will jump ahead. When moving the slider downwards you indicate how much equal spreading each sub voice phase will be given. There is a special mode when the slider is all the way up. This is the full blur mode.

It is hard to tell what the jump ahead mode means. To find out I set up an experiment that you can find in <https://www.malmgren.nl/Harmor/Harmor-07.flp>. I created notes of saw tooth in a piano roll, pattern 1, with silence in-between. This is because I want to see how the beginning of each note is behaving regarding phase. Then I created an automation clip for the unison phase slider.

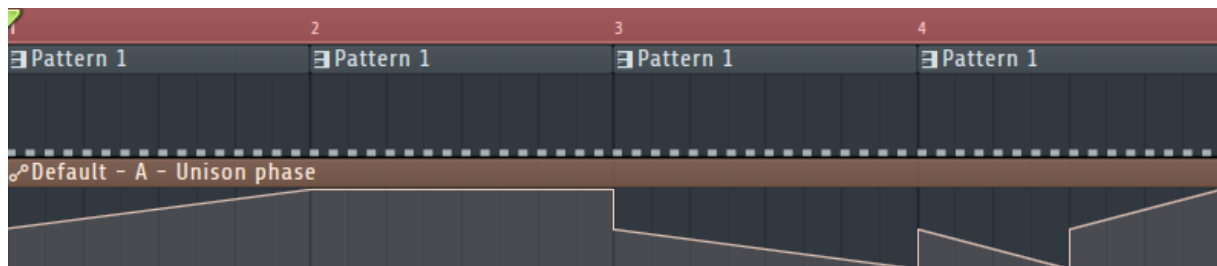


Image 30. Experiment designed to gain understanding about various positions of the phase slider of Unison in Harmor.

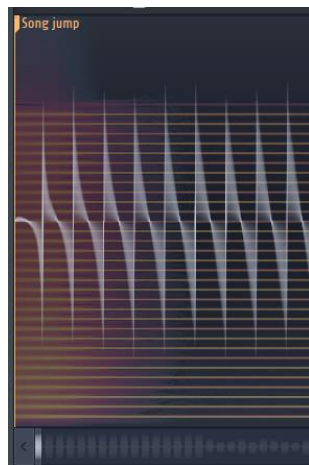
The first bar starts with the slider at 50% going upwards. This is the jump ahead mode. At 100% it stays in the blur mode during the second bar. Then at the beginning of the third bar it jumps down to 50% goes downwards, the equal spreading mode. It goes down to 0%. Then the fourth half bar it repeats the spreading mode and the other half is repeating the jump ahead mode.

I run this with order of 3, thus there are three sub voices of unison. This is recorded and we see the result in Image 31.

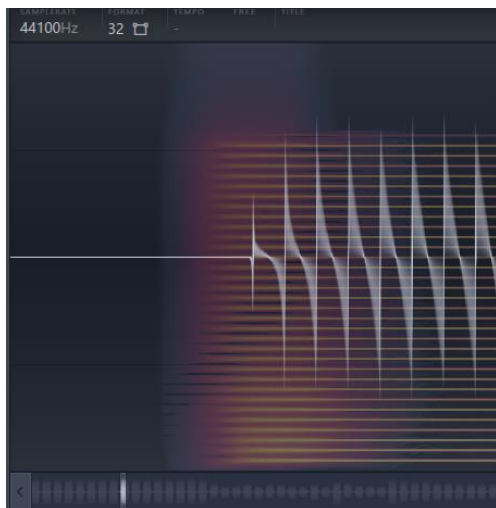


Image 31. Analyzing Unison phase. Recording displayed in Edison.

Luckily the different bars give different appearance in the spectrum analysis. The first bar is the most mysterious part. It is the jump ahead mode. Is it doing anything?



To see what is going on, I turn on dual view mode in Edison. That gives us both a spectrum as well as an oscilloscope view of the sound. At the beginning, at the song jump, the waveform starts downwards.



As we move forward the starting point of the waveform is changed but all sub voices are moved forward an equal amount. I cannot hear any spectacular effect in this mode! A higher order in the jump ahead mode gives an increased amplitude because the different waveforms are in sync and thus amplifying each other, but this is also the only measurable effect of the jump ahead mode.

At the second bar, the full blur mode, the phase of each partial is changed randomly.



Image 32. Unison, phase full blur mode.

When changing the partials within the waveform then the entire waveform becomes irregular.



Image 33. Phase with equal spread mode.

In the third bar, we are analyzing the “spread evenly” mode. We can see that the base waveform is replicated by the number of order. This means that the phase configuration of all partials is given an extra offset but each sub voice is keeping the internal phase configuration, otherwise the shape of the waveforms had changed. When the phase slider reaches the button the sub voices reached the maximum of spreading. With an order of two and a saw wave form the pitch thereby doubles. Then for each increased order the pitch increase accordingly. With a square wave form, all even orders the sub voices are cancelling out each other giving complete silence.

### Unison distribution

The manual says this unison distribution provides variations in pitch and panning. Most of the time I find variations in pitch. I find very little or no variation in panning and phase. To investigate this further I created <https://www.malmgren.nl/Harmor/Harmor-08.flp>. See Image 34 for a screenshot of the result.

This drop-down menu has 5 choices: Classic, Uniform, Blurred, Random and Hz (Special). Classic looks like it has a wider distribution of the partials compared to the other distribution modes. Uniform has the most uniform distribution. Random and Blurred got almost the same distribution like Uniform but they have different amplitude of the partials. In some cases, Blurred and Random are the most regular repetitive modes unlike what the name indicates. Hz is a completely different mode. It looks like it is distributing partials evenly from each sub voice. In this case, you could also think of the Hz mode as if they turned the distribution of the frequencies “upside down”. The fundamental frequency is the same but the 4 frequencies of the fundamental tone (when the order is 4) they got the biggest spread. Then for each frequency the spread between the frequencies get smaller and smaller.

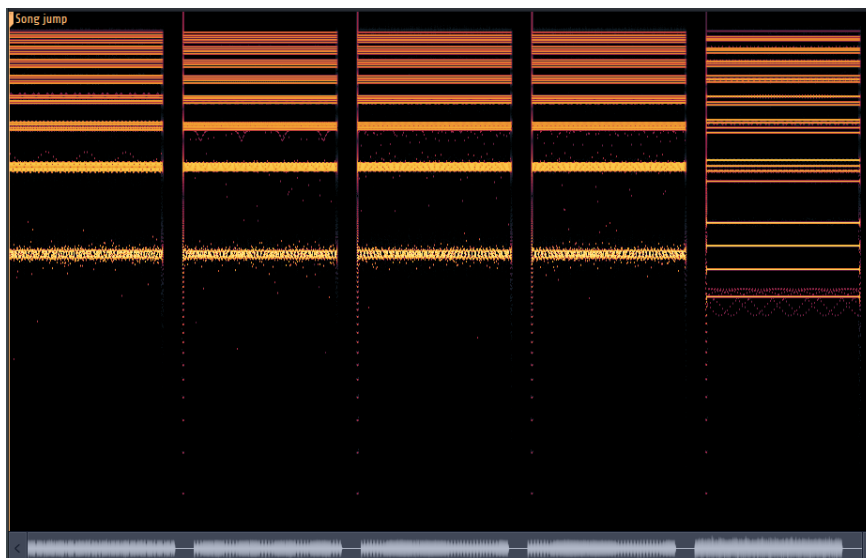


Image 34. To find the difference between the unison distribution modes I play the same note with the different modes and carefully try to find out the differences.

### Unison alternate distribution

Below the unison distribution drop-down menus you find a little knob “alt”. This button provides phase inversion, a stereo enhancing technique that by adding some of the right

signal to the left channel out of phase and some of the left signal to the right channel out of phase. Obviously, you can only make efficient use of this effect if you added a bit of panning first. This explanation here is my guess because the manual is somewhat vague on this point.



Harmor got two parts marked as A and B and we will use this for an experiment. A/B is actually two different Harmors “named” A and B. Suppose you got the project Harmor-08 open. By clicking the button below the B we can turn on the second Harmor. Now we will investigate the alternate distribution button of Unison. Right click the pan slider of Unison in A and select “copy to B”. Do the same for Order, the pitch thickness and phase sliders. Turn on Unison *alt* in both A and B. Now when we play the track part A will have the unison distribution drop-down automated but B don’t. When B is set to Classic and Blurred the two Harmors completely phase cancel each other. This is normal with this phase inversion. When B is set to Random there is no complete phase cancellation. You could make the conclusion that indeed there is some sort of randomness going on so that the distribution mode *Random* cannot completely phase cancel. Now comes the surprise!!! When we set B to Uniform then both Uniform and Random phase cancels! There IS a connection between these two modes. I make the conclusion that these two are identical somehow!



## Chapter 4. Subs, clipping, volumes, gain etc

In this chapter I will finish explaining the timbre section, the part marked with a dotted line in Image 35.



Image 35. Discrete dotted lines mark the different parts of Harmor. Here in this image they are marked with orange lines. In this chapter, we will talk about the part in the dotted orange line.

### Sub harmonics 1, 2 and 3

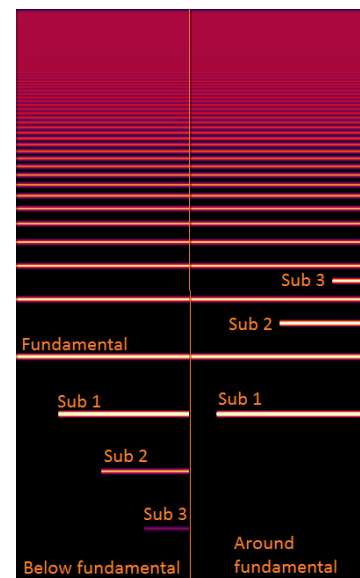


Above the label “sub” you find three sliders. They operate on sub 1, sub 2 and sub 3 from left to right.

There is also drop-down menu. These sliders and the menu is used to enforce the fundamental frequency in two different ways:

- Around fundamental or
- Below fundamental.

In Image 36 the sub sliders are set to 100% one after another in the two different modes. It is of course possible to turn on the sliders just a little to make different effect.



## Harmonic protection

As you already noticed the fundamental frequency is essential for the tone. Further on we will see various filters manipulating the partials and it could happen that we by accident weaken the fundamental frequency. For such event the slider marked *prot* when increased will protect the fundamental frequency from “destruction”.

## Harmonic clipping threshold

The clipping slider is the input for the type of clipping selected with the drop-down menu under the *clip* label. When playing around with these clipping methods it is useful to display the visual feedback window and play a note and experiment with the values.

## Harmonic clipping mode

There are five clipping methods:

1. High threshold. When the clip-slider is at the top, clipping is turned off. When moving the slider downwards the clipping takes more effect and it is decreasing the amplitude from the fundamental frequency and up. With the default settings, the protection of the fundamental frequency is ignored with this clipping method. This is because of the default unit order in the ADV tab. In the section *Unit Order* we will talk more about unit order and how this makes it possible to protect the fundamental frequency also from changes of the different clipping methods.
2. Soft low threshold. When the clip-slider is at the top, clipping is set to the maximum meaning the lowest overall amplitude. When moving the slider downwards more partials emerge starting at the lower frequencies and gradually adding higher frequencies. The clipped partials are not abruptly clipped but soft and gently.
3. Sharp low threshold. This method is like Soft low threshold but slightly less gentle.
4. Hard low threshold. This method is like Sharp low threshold but shows an abrupt disable of clipped frequencies.
5. Subtraction. This method is like the previous three clipping methods. The cut off line is not as hard as the Hard-low threshold but also not as soft as the Sharp low threshold- somethings in-between.

Image 36. Sub enforcement. Here is how it looks when turning on the sliders in the two different modes when playing a C5 sawwaveform



## Effects dry / wet mix



On the right side of the clip slider you find the FX dry/wet mix slider. The default position is at 0% in the middle. “FX” refers to the sounds produced in the FX tab that we will talk about in *Chapter 15. Distortion* and the following chapters. With this slider, it is possible to mix the effects sound produced on the FX tab.

At 0% all the effects sounds are sent to the output. At 50% the original signal is mixed with 50% of the effects and 50% of the original signal. At 100% the effects are turned off and only the original signal is sent to the output. At negative values, exactly the same behavior



but the signal is having reverse polarity, it is inverted. If you look at it in an oscilloscope you can see it is turned upside down.

To be able to turn off the effects is useful in an A/B scenario where one part of Harmor, for example part A, make use of effects and where the other part B need no effects. Then for part B set the slider to 100%.

### Auto-Gain



Auto-Gain can “normalize” the amplitude of sound. The effect is more pronounced when the knob is set to higher values. Sounds of evenly spread partials are not affected by Auto-Gain. When the sound is “unbalanced” Auto-Gain comes into effect. If we for example got very few partials engaged, then higher values of Auto-Gain will increase/amplify the volume. On the other hand, if we got many partials engaged in such way that the sound will be loud then Auto-Gain lowers the volume.

When you experiment with the default waveforms of timbre 1 and 2 you will not notice any effect of Auto-Gain. Try making a timbre with just a few partials engaged to see Auto-Gain amplify the volume. When you create a timbre with very many or very loud partials then Auto-Gain will lower the volume.

In section *Auto-gain mode* of chapter 24 we will come back to Auto-Gain when we talk about different Auto-Gain modes.

### Time domain volume, frequency domain volume

The sound volume can be controlled at the stage when the sound is still a list of partials. This is the slider marked with *env*, frequency domain envelope volume. With the visual feedback window, you will notice that the partials will fade out when the slider is lowered to the bottom. The tooltip says -59.2 dB and all at the bottom it says -INFdB. At that point you cannot hear any output and the visual feedback shows no partials. The default value is halfway and where the tooltip says 0 dB. All the way to the top the tooltip says 12 dB.

The slider *vol*, time domain volume, controls the levels after the partials has been converted to a sound. You cannot see this in the visual feedback window but you can hear it.

### Velocity switches

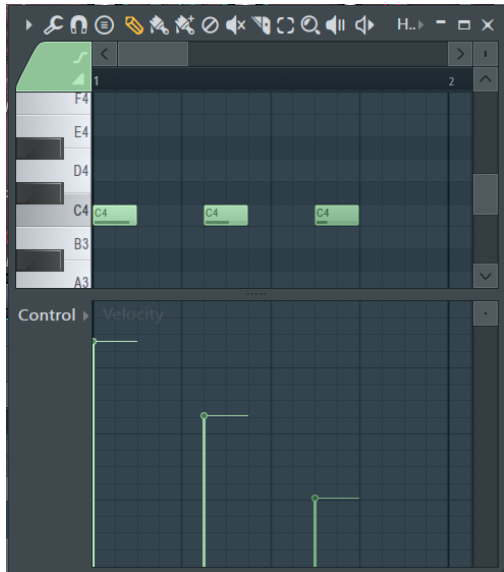
Finally, we will talk about the *vel* buttons. When the topmost is turned on, the amount of velocity will determine how much of the attack is skipped. If the second button is enabled, the velocity determines the speed of the attack. The third button connects the velocity to the release scale. To try this, I created a test project that you can find at <http://www.malmgren.nl/Harmor/Harmor-09.flp>



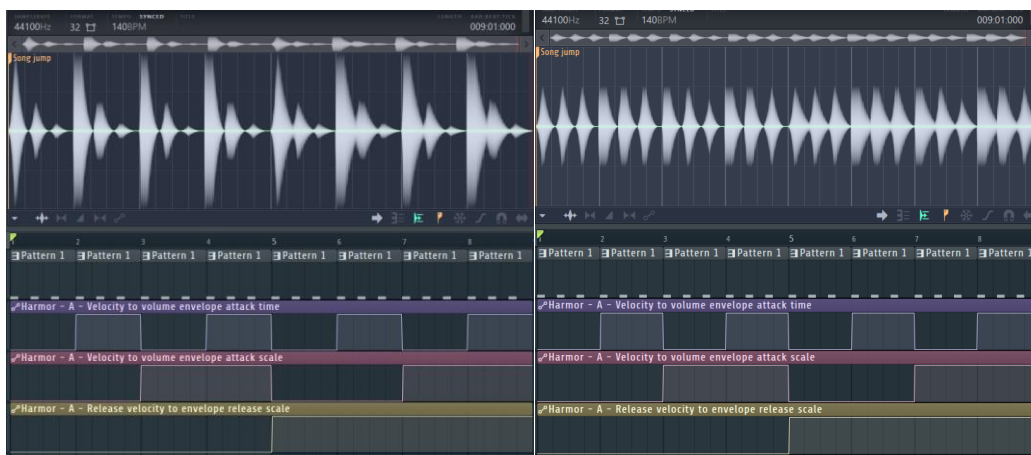
This project plays the default saw waveform in Harmor. It has the Volume envelope turned on and it looks like here below.



The project has a pattern with three notes of C4 with a gradually declining velocity.

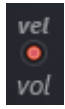


Now comes the clever part. With automation, we turn on and off the three buttons in such a way that we can try out all combinations!



Here to the left is the pattern of the automation set in such way that we can try out all combinations of the three buttons. The first pattern has a slope on the attack and a slope on the release. Clearly when looking at pattern 5, 6, 7 and 8 when “release velocity to envelope scale” is turned on we can see that it increases the release time considerably. Comparing 1 and 3 of “Velocity to volume envelope attack scale”. It makes the release to become quicker with a high velocity but with a lower velocity the second note in the third bar it is almost the

same as the third in 1. While the first note in 1 and 3 are very different. It seems that “Velocity to envelope attack time” makes the attack quick in all combinations with other modes.



In the example, here above to the left, we are using the default setting of “Velocity to volume” in the main area. It instructs the velocity to influence the main volume. If we turn off this and run our experiment again then we will get equal volume on each note and only our *vel* buttons will influence the velocity. That is the image to the right.

## Chapter 5. Basic filtering

In this chapter, we talk about the basics of filtering marked with an orange line in Image 37. This part is symmetrical such as the upper part consists of the controls of filter 1 and the lower part are the controls of filter 2.

So why are we “ignoring” the ENV button and adapt in this chapter? These buttons are not doing anything in the default setting. To make ENV to do something useful we need to connect them to a function and I leave that to the section *Filter envelope amount*.



Image 37. Basic filtering.

### Filter 1 & 2 mix (parallel – serial)

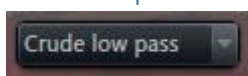
Filter F1 and F2 can operate alone, sequentially or in parallel with each other. The default mix value is 100% and that is the serial mode: first filter 1 and then via filter 2 and out. In the table here below all variation are explained. We start from the default setting, turning the knob counter clockwise.

F1 only via F2	First F1 then also via F2	Only F1, F2 off	Both F1 and F2	Only F2, F1 off
100% serial	50% serial	0%	50% parallel	100% parallel

The filters in Harmor operates on the list of partials before they are converted into a sound. This has the benefit that the sound is not influenced by the filtering process itself other than the intended result, this filtering is “lossless”.

To experiment with the filters, we turn off F2. That is the third column in the table here above. When doing so we are only looking at the result of turning knobs in the upper part of the filter section. Then turn on the visual feedback. Now the experimentation can begin to learn how these filter knobs are working.

### Filter shapes

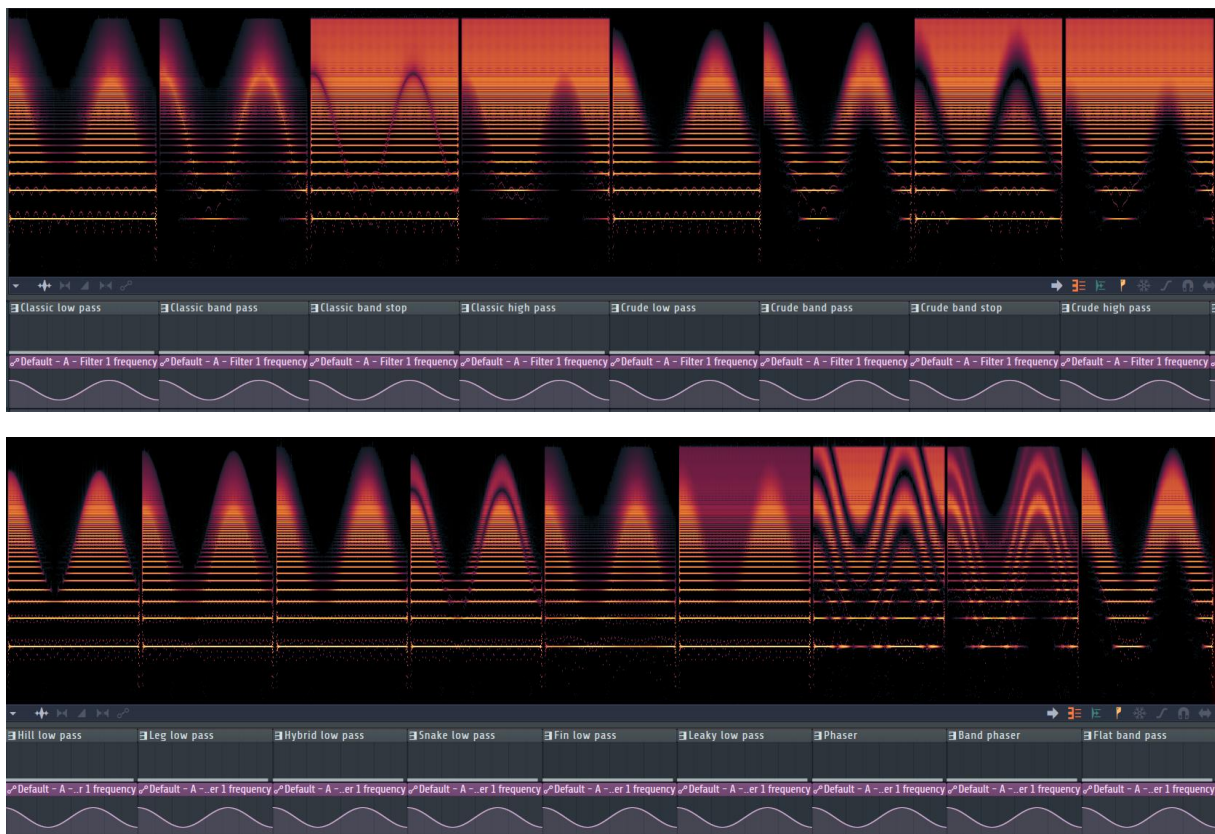


The drop-down menu *Filter 1 shape* at the top says Crude low pass. This drop-down menu has four groups of filters as described here below:

- Pass filters. Pass filters let some frequencies pass through. The other part is often not let through at all. In one case, just a little is let through – “leaky”. There are 12 pass filters.
- Stop filters. They block one part of the frequency spectrum.
- Phaser filters. They block several sections of the frequency spectrum.
- Custom. These filters will be discussed in the advanced filtering in *Filter shape*.

In the case of pass-filters traditionally the frequency spectrum is arranged so that the lowest frequencies are found at the bottom and the highest frequencies at the top of the diagram. This is also how the visual feedback in Harmor is arranged and indeed also the spectral view of WaveCandy. With this orientation, the names of the filters come naturally. Low pass filters let lower frequencies pass through. High pass filters let higher frequencies pass through.

In the project <https://www.malmgren.nl/Harmor/Harmor-10.flp> you find the project of the images here below where we can see the result of all filters (except the custom shapes) in the filter shape drop-down menu.



## Filter frequency



The knob frequency sets the “special point”. What that special point is doing depends on the filter type. For a low pass filter the special point is the point at which higher frequencies are not allowed through.

## Keyboard tracking, key to filter frequency



Because of the fact that the filters are operating on the list of partials it is possible to relate the filter to the fundamental frequency. Of course, we need to experiment with automation to find out how this works! In the project

<http://www.malmgren.nl/Harmor/Harmor-11.flp> you find the keyboard tracking experiment in Image 38. In this project, I selected the crude low pass filter and I have the filter 1 frequency set to 50%.

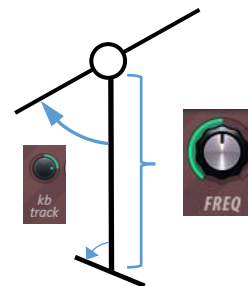
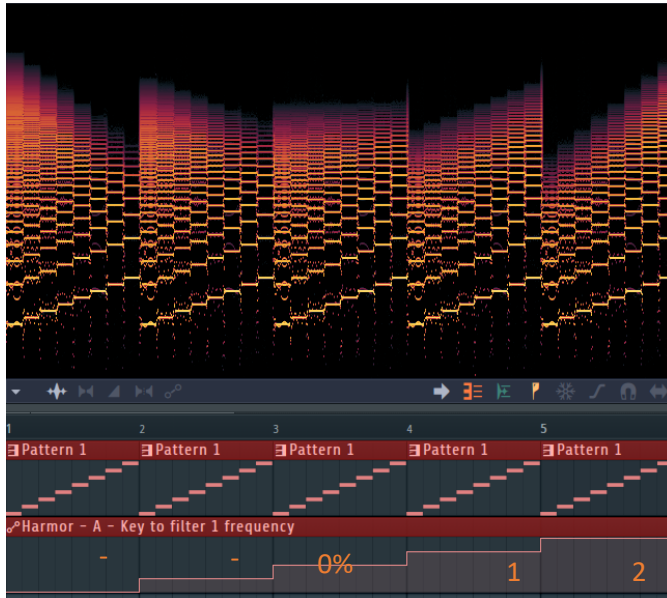


Image 38. Keyboard tracking experiment. Setting different value of the 'key to filter 1 frequency' knob.

Let's first talk about bar 3 in Image 38. In bar 3 the 'key to filter 1 frequency' knob is set to 0%. With 0% the filter 1 frequency has all the influence on the height of all the cut-off frequencies regardless of the fundamental frequency. Here I would like to introduce a metaphor for how the kb track knob works. It is as if the filter 1 frequency influence the height of an imaginary pivot lever. The angle of the lever is influenced by the 'key to filter 1 frequency' knob **and** the fundamental frequency. I tried to figure out the math of this but I gave up on that for now.

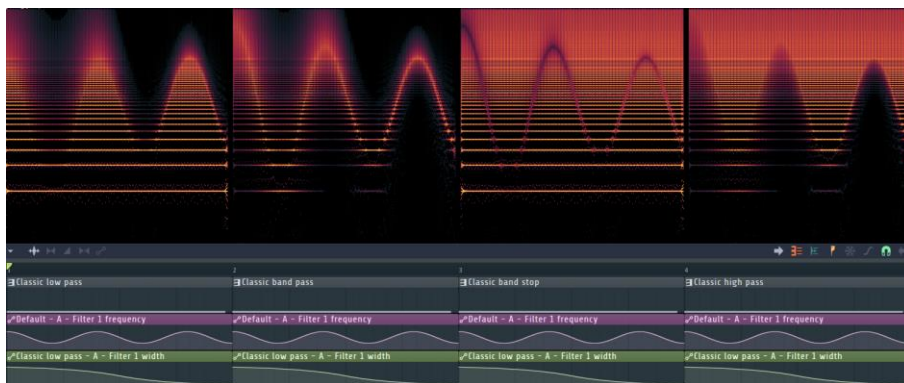
## Filter width



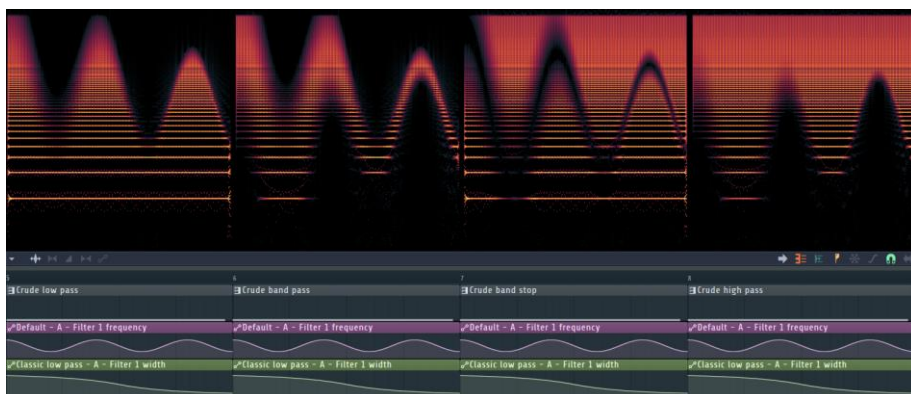
Now we will talk about the filter width. This is luckily very easy to illustrate with a spectrum diagram in WaveCandy! Here in the images below I start with a high width and then decrease the width gradually. The sample project for this is found in:

<https://www.malmgren.nl/Harmor/Harmor-12.flp>

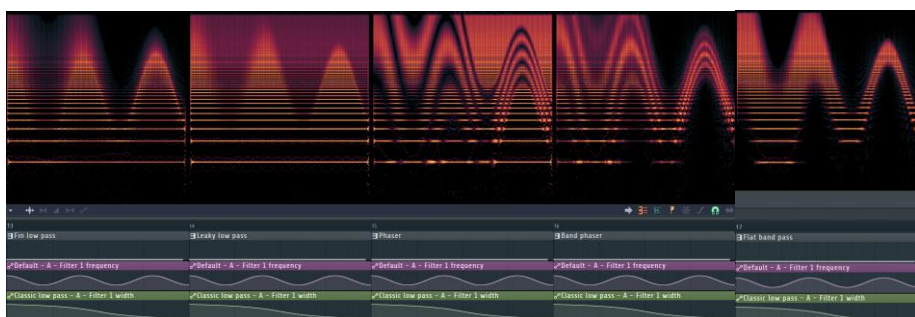
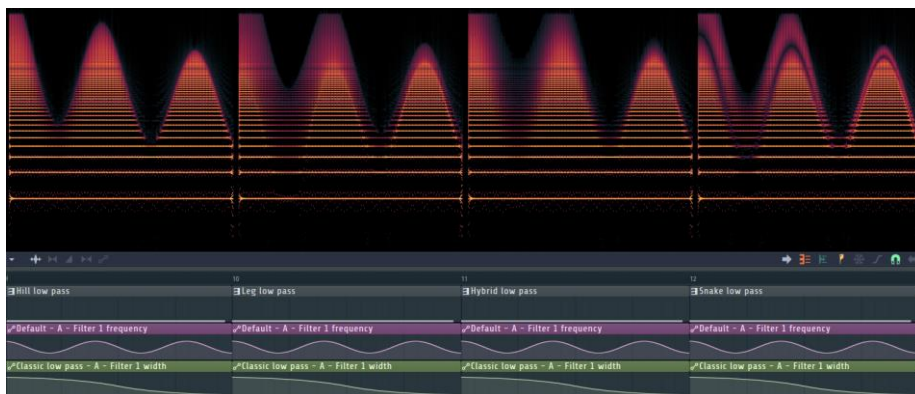




The classic low pass for example starts fuzzy and gets more precise towards the end.



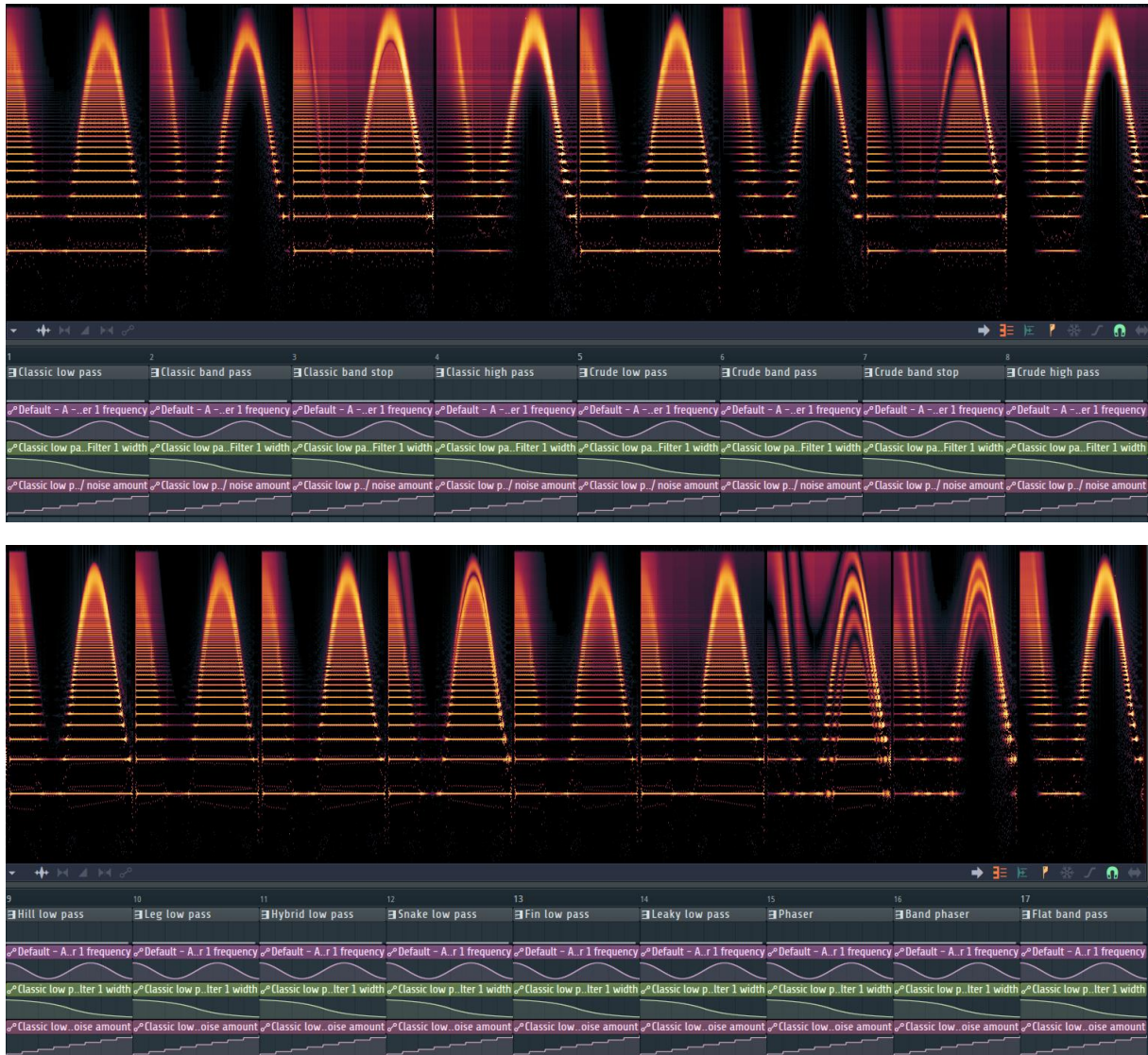
The crude band pass and the band stop both gets narrower towards the end.



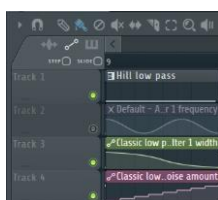
## Filter resonance amount

Next we will see what the *RES* knob is doing. You can find this experiment in <https://www.malmgren.nl/Harmor/Harmor-13.flp>. Normally in traditional filters when resonance is turned on then there is an amplification “bump” located at the cut-off frequency. Here below you can see this in the classic low pass graph.

Here I decided to increase the *RES* knob with the stairs automation graph. Especially in the filter shapes with high pass you can see that there is “noise” that decrease as the Res knob increase.

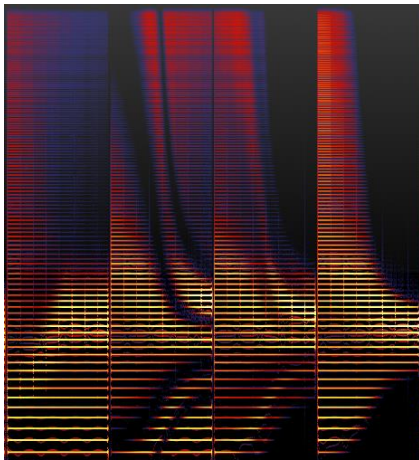


With the experiment project, we can now do much more. For example - turn of automation tracks.





If you click the radio button for track 2 and then go to the mixer ,track one and make WaveCandy visible, then we can notice other shapes!

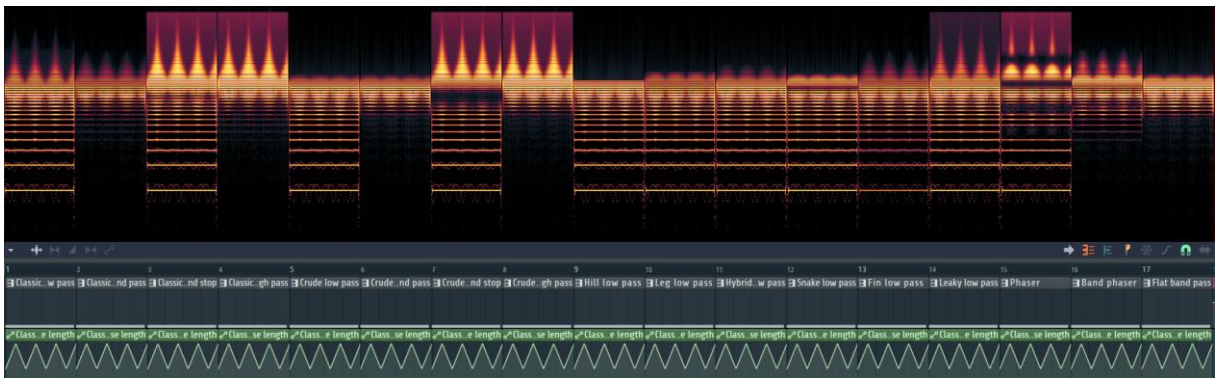


Leaky band pass, Phaser, Band Phaser and Flat Band Pass. Especially Phaser sounds a little like human vowel sounds.

### Filter resonance width



In Harmor the resonance also got its own width. When changing the width of the resonance it is possible to produce much more pronounced vowel sounds. In the project <https://www.malmgren.nl/Harmor/Harmor-14.flp> you see the different filters where the resonance width is automated.



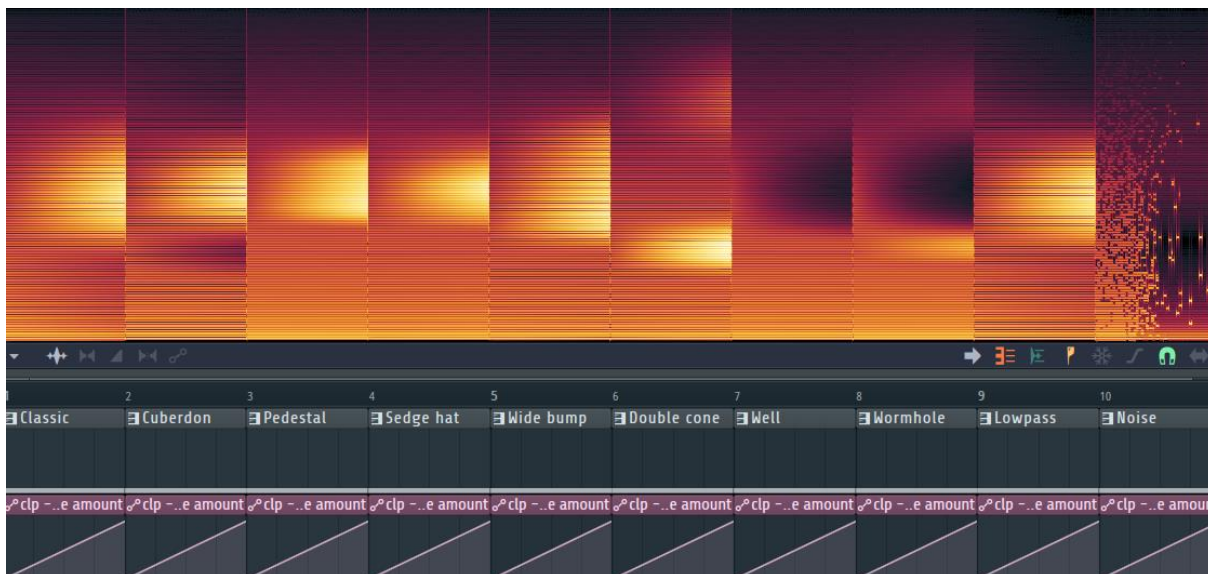
### Filter resonance type



There is a drop-down menu for the resonance type and there exists nine preprogrammed resonance types and two custom types. We will talk about the custom resonance types in the section *Resonance shape*.

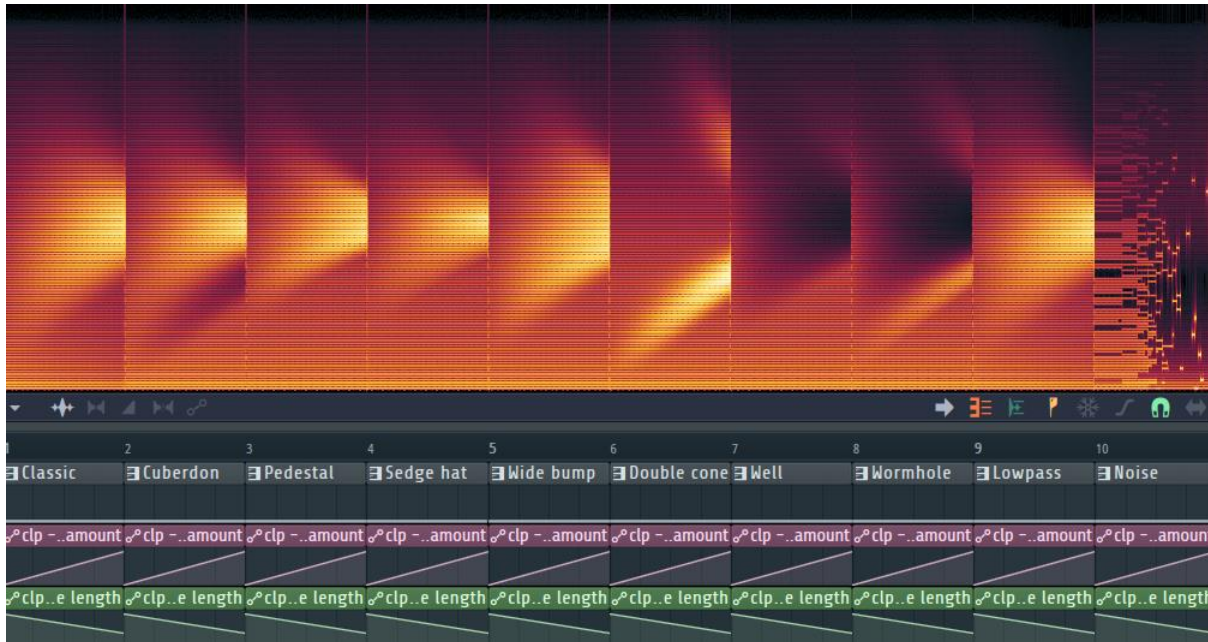
To investigate the resonance types I created a series of projects using only the Crude low pass filter combined with the different resonance types and various variables.

In our first experiment, only the resonance / noise amount knob is automated. The cut off frequency is set to 80%. <https://www.malmgren.nl/Harmor/Harmor-15.flp>



In this graph I turned off natural weight in Edison. Lowpass is the default resonance type. It is specially designed to lower the amplitude of partials above the cutoff frequency. This is useful as a default resonance type because this match quite well how traditional resonance types works.

In our next experiment, we automate the resonance width, in combination with various resonance types. <https://www.malmgren.nl/Harmor/Harmor-16.flp> This experiment describes nicely what the width knob is doing.



### Filter resonance adaptive width / free noise

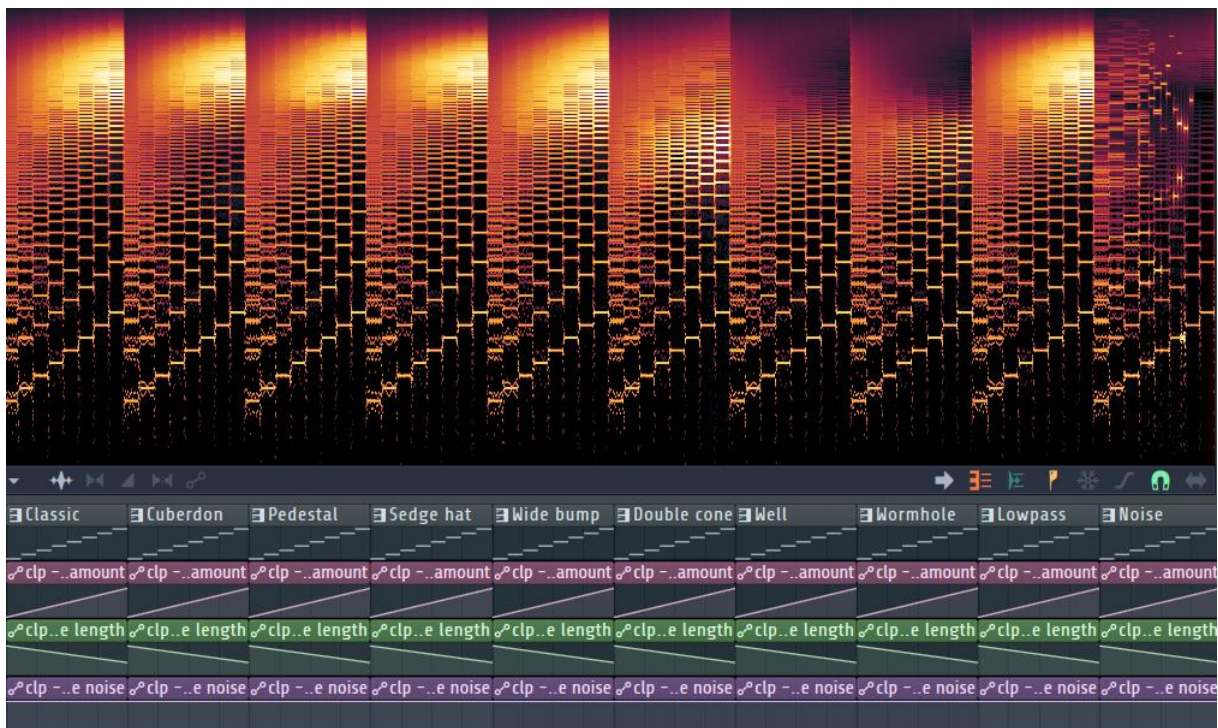
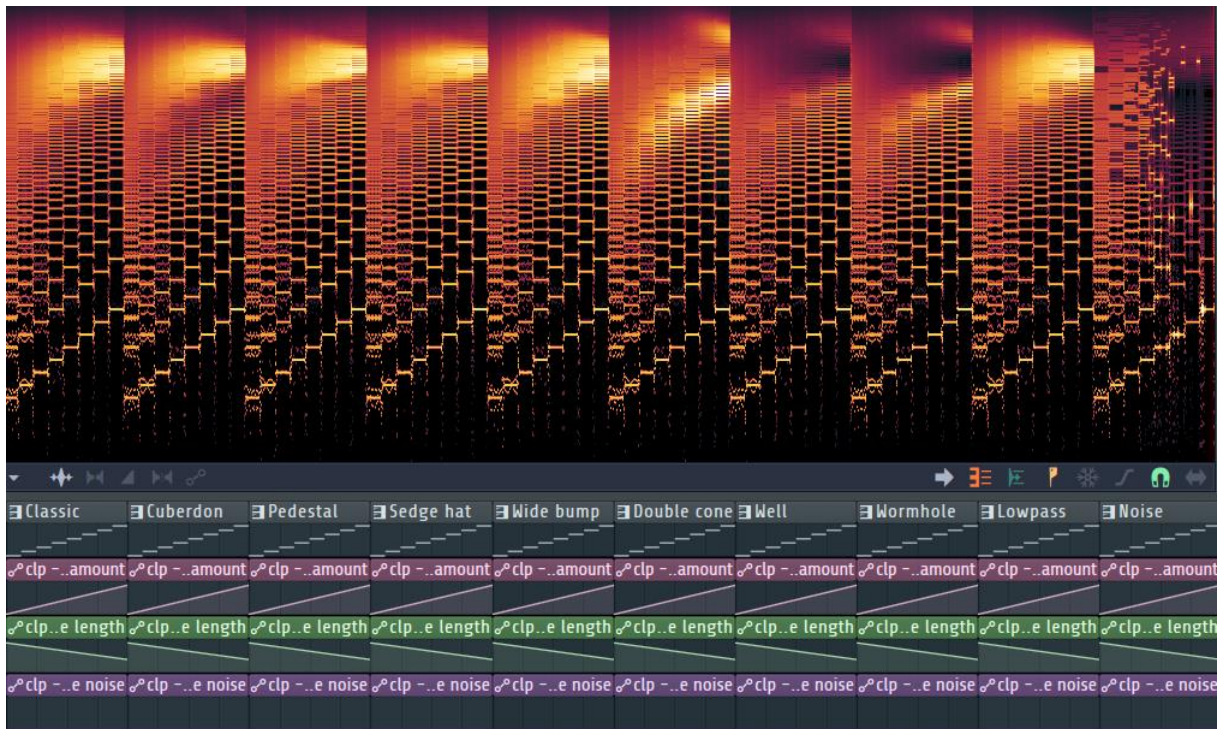


Next to the resonance width knob, *RES*, there is an *adapt* button. These two are connected to each other with a little black line to give us a hint that this button can modify the behaviour of the resonance width. The bandwidth of the filter changes width as a function of fundamental frequency. To experiment with this, I made a



pattern that play ever higher notes and I play these with and without the adapt button to be able to compare the two. This project is <https://www.malmgren.nl/Harmor/Harmor-17.flp>

The first picture is without adapt and the second is with adapt turned on.



What you can see is that the resonance is not pushed up by the higher frequencies when adapt is turned on. This trick can only be carried out with additive synthesis because in classic subtractive synthesis the filter does not have **that** information what the fundamental frequency is, but here the filter knows this and can use it.

### Filter resonance offset

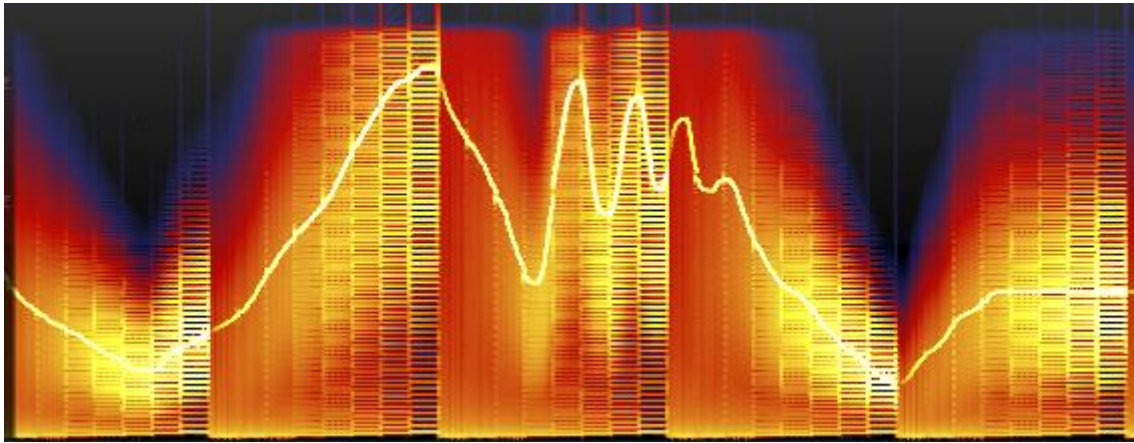


With the *ofs* knob it is possible to insert an offset between the cut off frequency and the resonance by a maximum of +2400 to -2400 cents. Until now when we were talking about the filter effects the resonance has always been connected to the cut off frequency but this makes it possible to move the resonance above or below the cut off frequency.

### Filter resonance self-oscillation



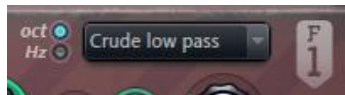
Then we got the resonance self-oscillation effect. The manual talks about that this knob creates a “tonal sound” and that is correct. You can also see it as a visual representation of where the center-point of the resonance is.



### Filter oct / Hz switches



Next to the filter resonance type drop-down menu you find the oct / Hz switch. When changing to Hz mode the resonance is spreading in the frequency spectrum while in oct mode the resonance is more concentrated.



Next to the filter shape you find the filter scale switch. It is not so that when you play around with this switch and fantastic new things happens to your sounds. It looks like the scale is more compressed with Hz mode and that's it.

## Chapter 6. Blur



Blur smears partials horizontally. If you try this effect on a saw waveform there is not so much to be noticed because that waveform already occupies

all partials so there is nothing left to smear horizontally. When switching to a square or sine wave form it all starts to make sense. With the less dense partial waveforms we can mix in partials from the saw wave form, with this blur effect.

### Harmonic blur amount

The default setting is that Blur *mix* is set to 100%. The Harmonic blur amount *harm* is set to 0% so the result is that it is turned off by default. When increasing, the harm knob the effect can be noticed.

### Top and bottom tension

It is possible to balance the effect over high frequencies versus low frequencies. For this purpose, we got two knobs:

- Harmonic blur top tension, 0 to 100%.
- Harmonic blur bottom tension, 0 to 100%

The default setting of the tension buttons is 50%. This is also the setting where they are in balance.

You could think that since they have almost the same name and they are placed next to each other that they work the same way but this is not the case. They work like opposite of each other!

The top tension button suppresses the most top frequencies at 100% and amplify the most top frequencies at 0%.

The bottom tension button suppresses the most frequencies at 0% and amplify the most frequencies at 100%. It looks like the bottom tension is reaching almost the entire frequency spectrum whereas the top tension indeed only modifies the top frequencies.

### Blur attack and delay

There are two knobs left:

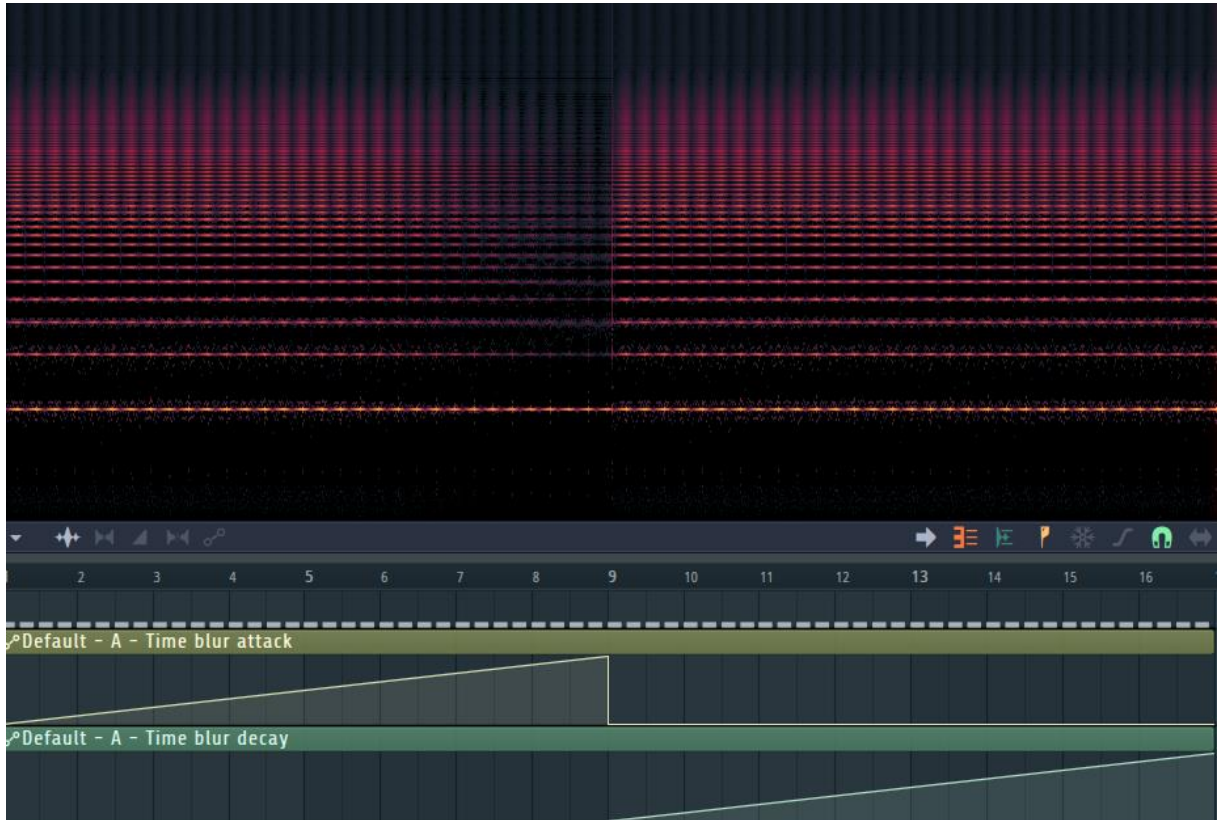
- Time blur attack, 0 to 100%
- Time blur decay, 0 to 100%

The attack knob influence how quick the effect will start. The decay influence how long time it takes for the effect to die out. I have not been able to see any result of the time blur decay knob. I tried it in combination with volume envelope and automation of different sorts. I cannot see that this knob has any effect at all.



The important question is 100% of what? I tried to figure out by creating a long note of 32 bars to find out when the effect peaked. It took almost 32 bars! When creating an automation where I gradually increase the time blur attack it is possible to hear and see the effect.

In the project <https://www.malmgren.nl/Harmor/Harmor-18.flp> you can see how I tested the time knobs.



The wave form I am using is a sine wave. I set the mix to 72 percent so that you easily can see the fundamental frequency of the sine wave and all other frequencies are caused by blur. Harmonic blur amount is set to 100%. Top tension was set to 100% and bottom to 50%. The automation is done over the two time-knobs. At bar 6 you start to notice that the time blur attack is making the frequencies to start declining. That is when automation is at approximately 65%. As I said increasing the Time blur decay do not give any result in this context. This is not to say that it is useless. It could be useful in another combination, for example pluck! Read more about pluck in *Chapter 7. Pluck and Phaser*.

## Chapter 7. Pluck and Phaser

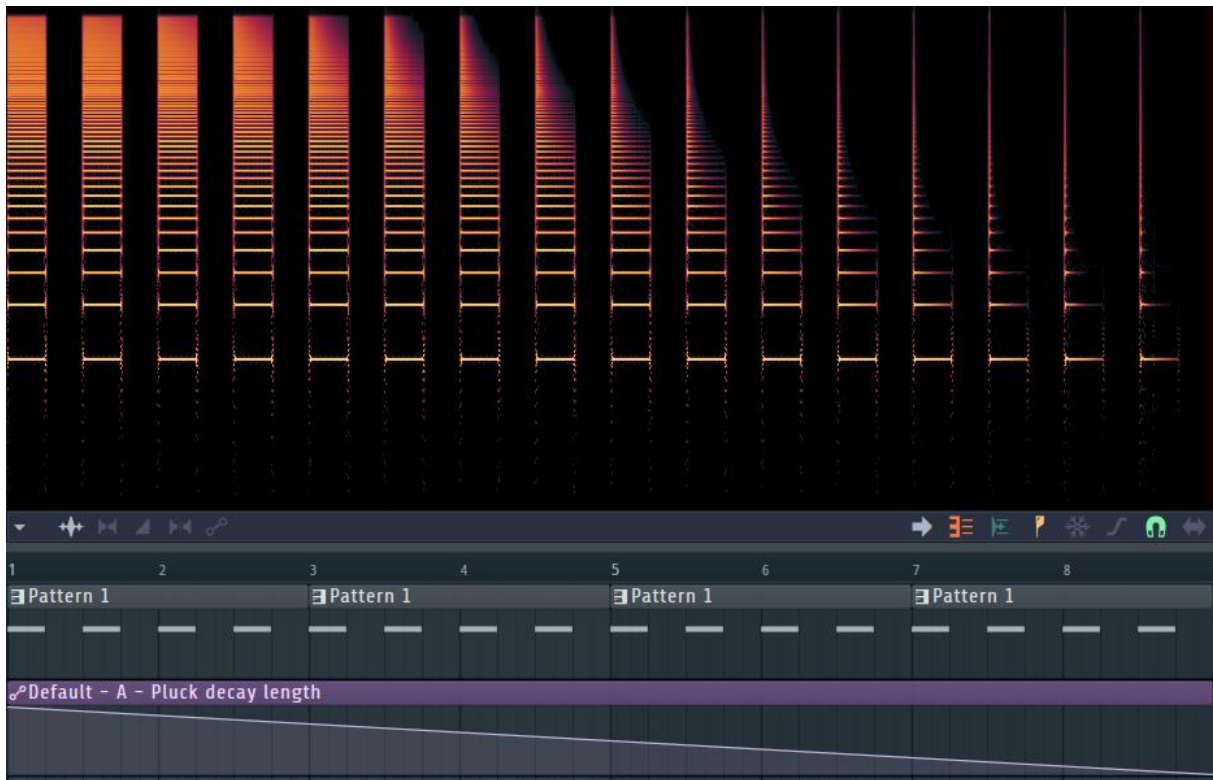
In this chapter, we will talk about the lower part of the filter section. We start with Pluck.

### Pluck

The pluck effect simulates the sound of plucking a string on a string instrument. The sound starts with all available partials engaged and then suppressing the highest frequencies first continuing to lower frequencies.

### Pluck amount

The time it takes for the decay is set by the knob Pluck decay length, *TIME*, also called pluck amount. The default setting is 100% where it is off down to 0% when it has the quickest decay time. Since the fundamental frequency is active throughout the effect the tone height remains stable. It worth noting that when the effect gradually removes all but the fundamental frequency whatever wave form we started with the end sound is a sine wave form. String instruments produce various forms of sine waves unless the sound is distorted as with for example electric guitars. That the effect transform the sound to a sine wave is the reason it sounds so convincing. We will talk more about Pluck in the section *Pluck shape* where we will see that is also possible to modify the shape of the pluck envelope. In the sample project <https://www.malmgren.nl/Harmor/Harmor-19.flp> I applied the pluck effect on saw wave form and gradually decreased the Pluck decay time.



It is not until around 45% that the pluck effect starts to get convincing.

## Pluck replaces blur decay



Next to the *TIME* knob there is a blur button.

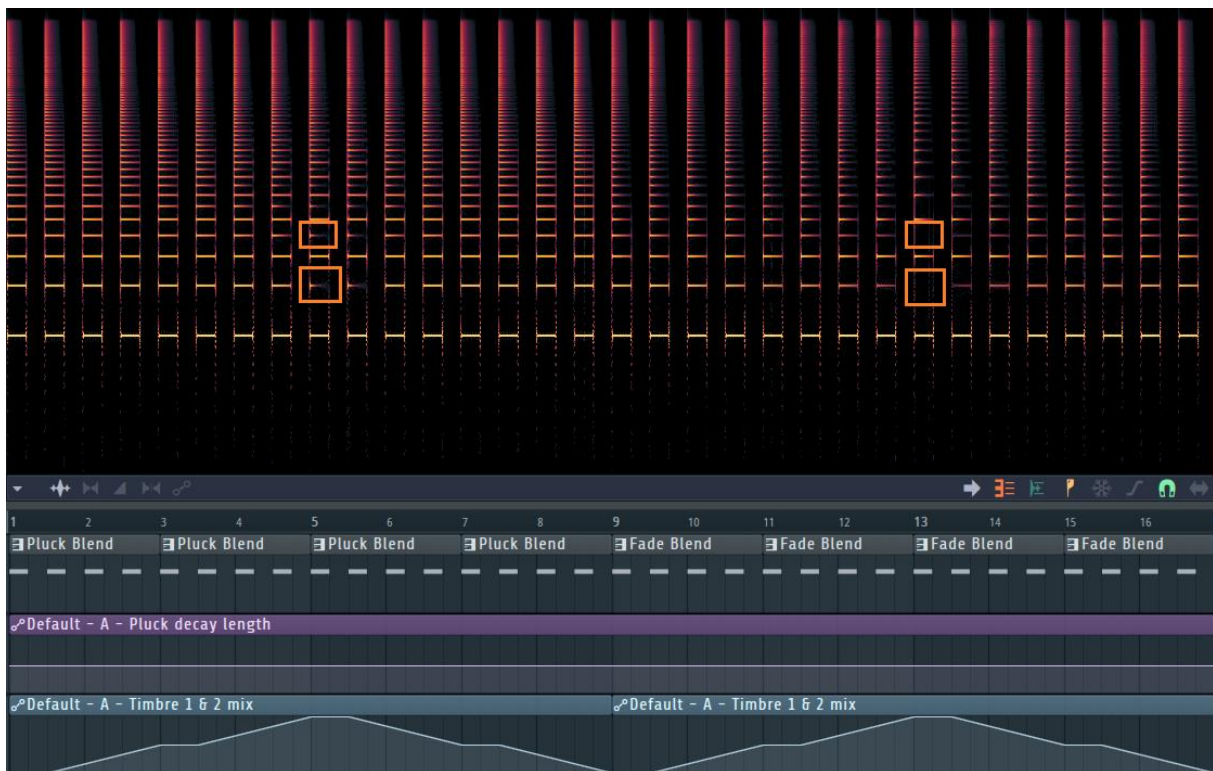
When you turn on this button I could imagine that we would replace the regular pluck shape formula with a blur decay formula. I can see that green ring around the time blur decay knob turns faint green and then indeed the regular pluck shape algorithm is turned off. To try out the effect of turning the time blur decay knob I made automation of various kinds but I cannot hear or see any result.



## Timbre blending mode: Pluck

We talked about pluck in the section *Timbre blending modes*. In that section we left off with a mystery to solve: What is the timbre blend mode pluck doing?

In the experiment project <https://www.malmgren.nl/Harmor/Harmor-20.flp> you will see that I created two patterns Pluck Blend and Fade Blend.

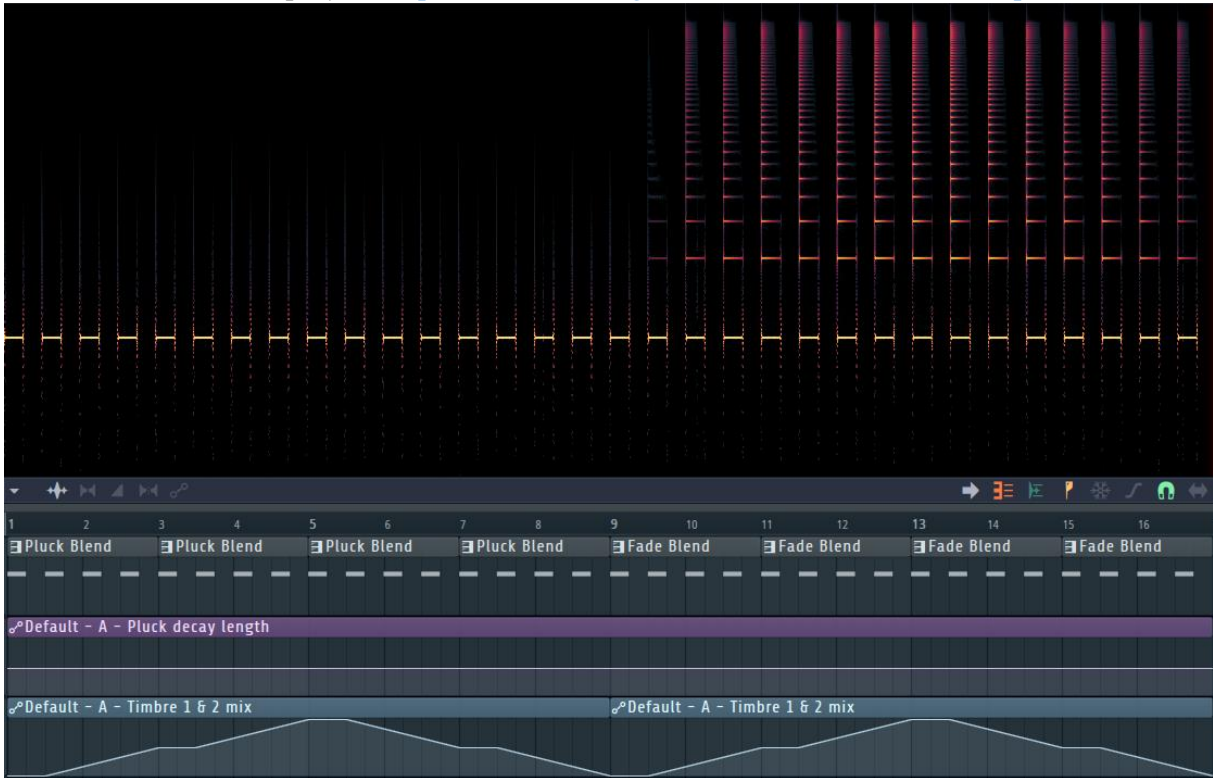


With the automation clip “Timbre 1 & 2 mix” I gradually switch between the two timbres. Both has the default saw and square wave forms. Overall we notice that the fade mode has a lot less partials activity. It is the difference at 100% timbre 2 mode we can see what is happening: The pluck blend mode keeps more partials active at the beginning before it gives way for the waveform. This is also the reason that the two partial windows displays two saw wave forms but this is just the very beginning of the note.

It is not so that if there are no partials that the pluck blend mode will add them like the blur effect is doing. It is the partials between already active partials that will become active. To



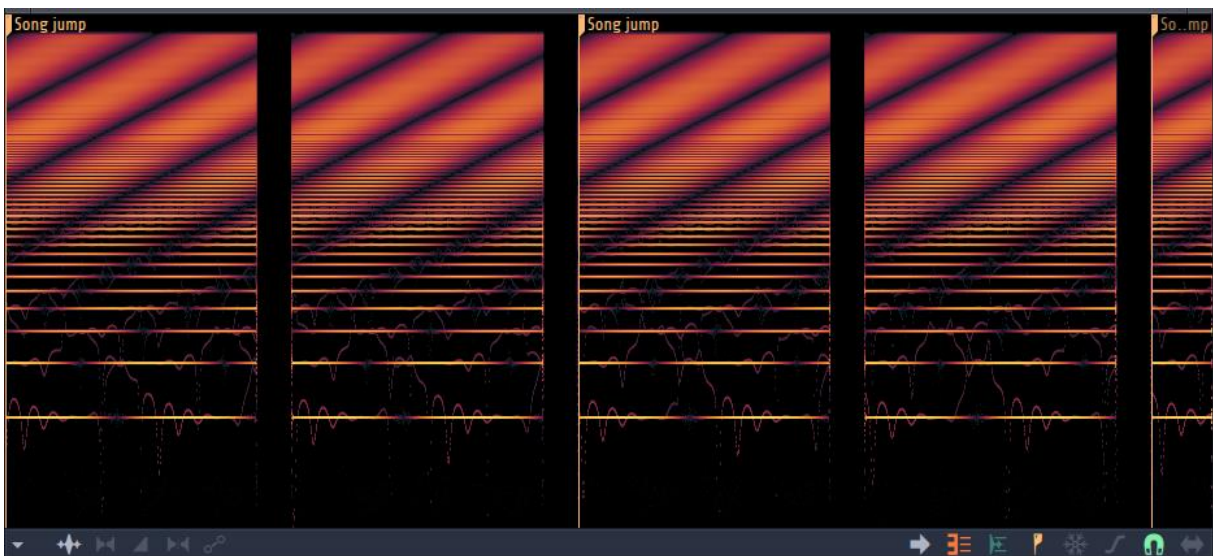
test this I created test project <https://www.malmgren.nl/Harmor/Harmor-21.flp>



A little tip for pluck mode! Try right-clicking a timbre and select randomize. That will create random partials and this makes the pluck sound unique!

## Phaser

In the basic filters chapter, we talked about a filter shape called phaser. The phaser effect here is a variation of the phaser effect we came across in the filter types. The phaser filter shape don't include the time as a variable, once set it will stay. This phaser effect on the other hand applies its filter based on the time. We can best see this by experimenting.

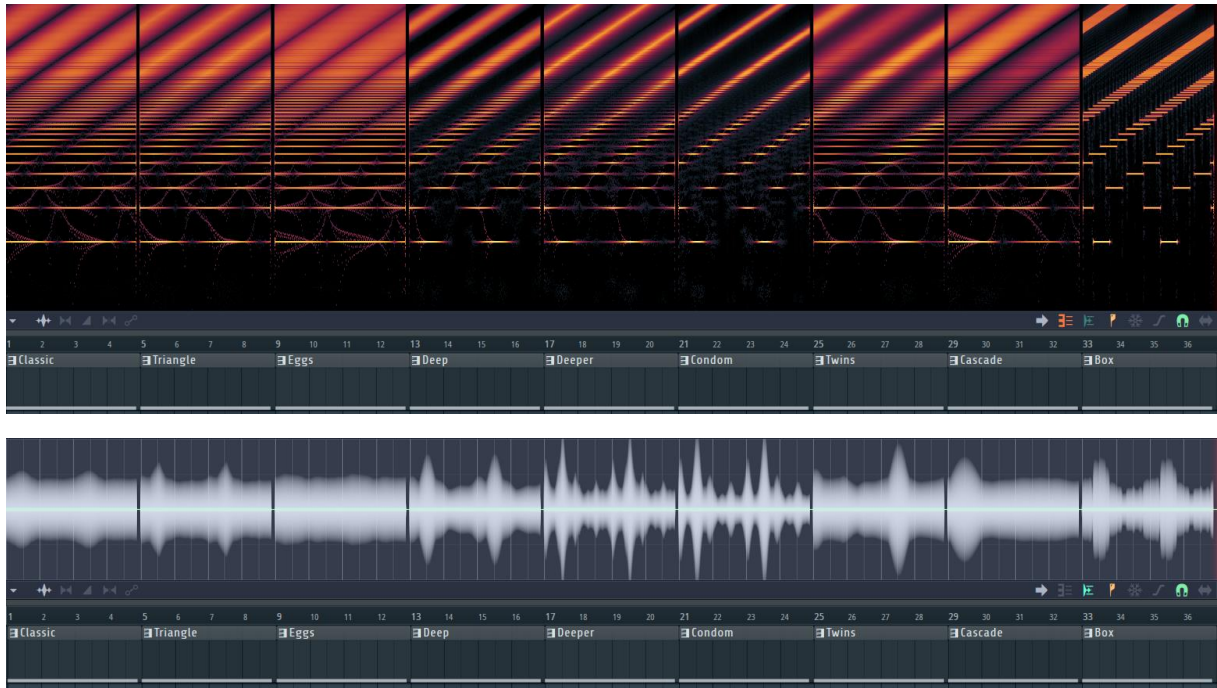


We can see that the effect produce a cancellation of specific frequencies and that this pattern is walking and that offset of the pattern is relative to the clock of the computer. The pattern

continues even if we stop playing a not for a while or we jump to the beginning of the pattern.

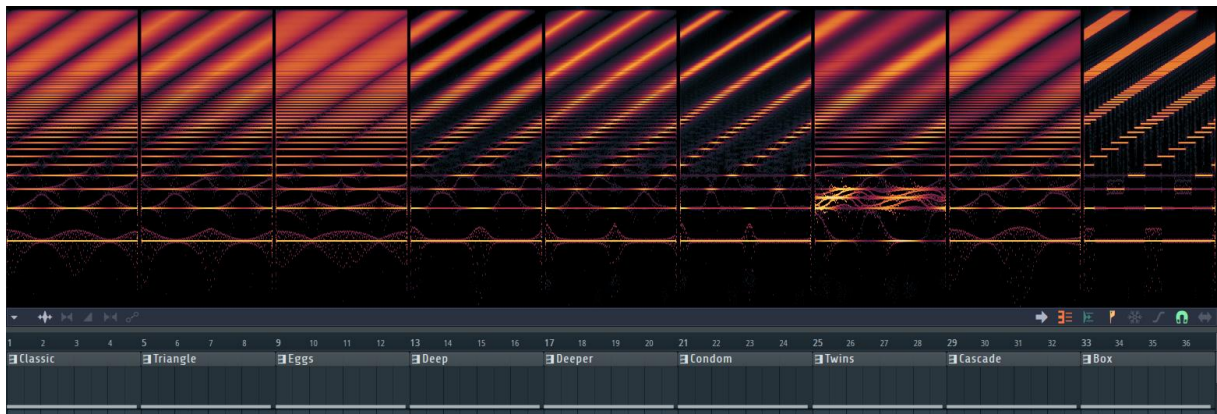
### Phaser type

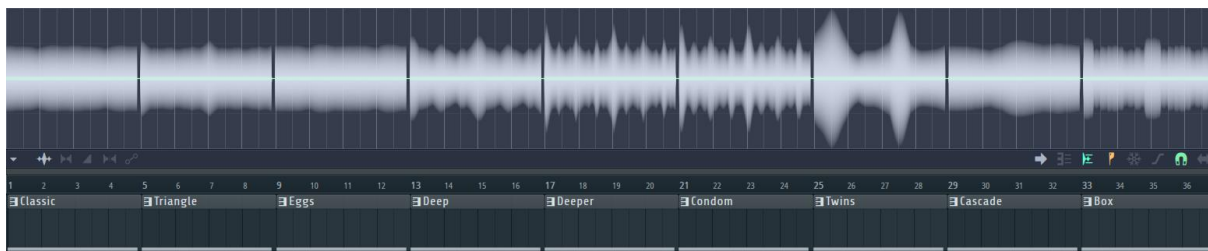
In the project <https://www.malmgren.nl/Harmor/Harmor-23.flp> we can see all the different phaser types in action, except *freq* and *custom*. We will talk about these custom phaser types in *Phaser shape*.



The manual refers to the first three phaser types as traditional types. When we switch Edison to display the wave form view then we clearly see how the amplitude changes throughout the phaser patterns. It is especially when the fundamental frequency is suppressed that volume becomes lower. As you might recall from *Chapter 4. Subs, clipping, volumes, gain etc* it is possible to protect the fundamental frequency.

In the file <https://www.malmgren.nl/Harmor/Harmor-24.flp> you can experiment with protection of the fundamental frequency.





It looks like the detection of the fundamental frequency went a bit out of hand in the twins pattern but for the rest it is doing a great job stabilizing the amplitude and the fundamental frequency.

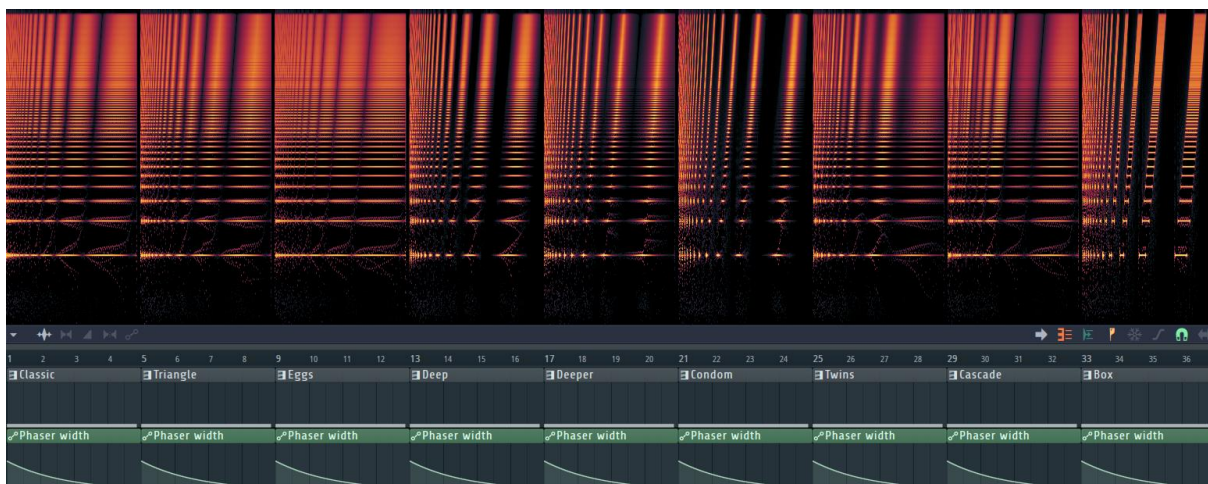
### Phaser mix



The *MIX*, phaser mix, knob makes it possible to adjust the amount of phaser effect that will be applied.

### Phaser width

The phaser width works like the filter width.



I programmed the phaser width to start from 62% going to 0%. The values above 62 percent was much more dense and impossible to see in a screenshot but feel free to open <https://www.malmgren.nl/Harmor/Harmor-25.flp> to experiment with more values of the phaser width automation clip.

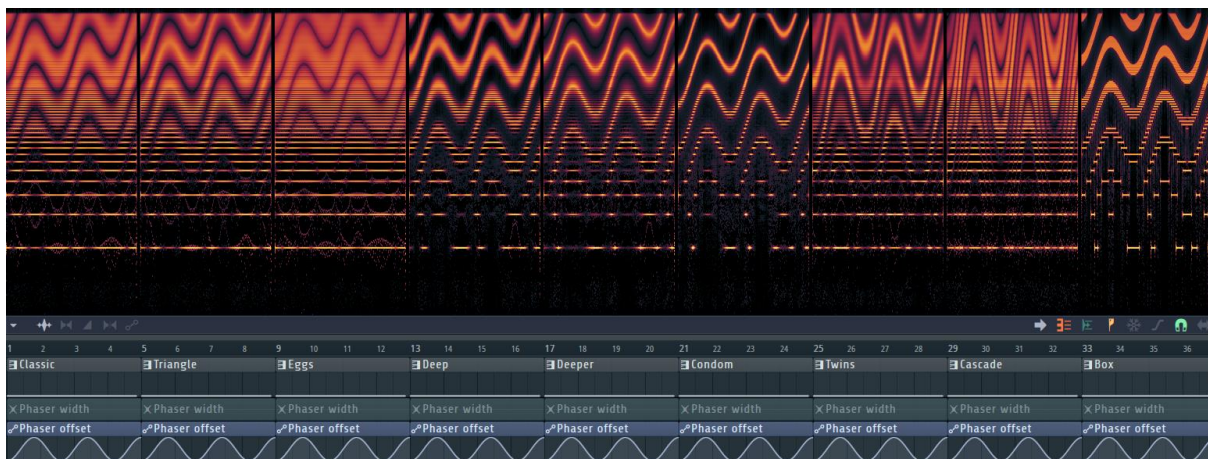
Please look at the cascade pattern. In this configuration, you can see that each “ribbon” of the phaser effect is gradually more amplified. This pattern is four ribbons wide and then it starts over.

### Phaser offset



Normally the phaser effect starts at a point based on the internal clock of the computer. It is of course possible to offset this compared to the computer clock. This is done with the Phaser offset knob marked *ofs*.



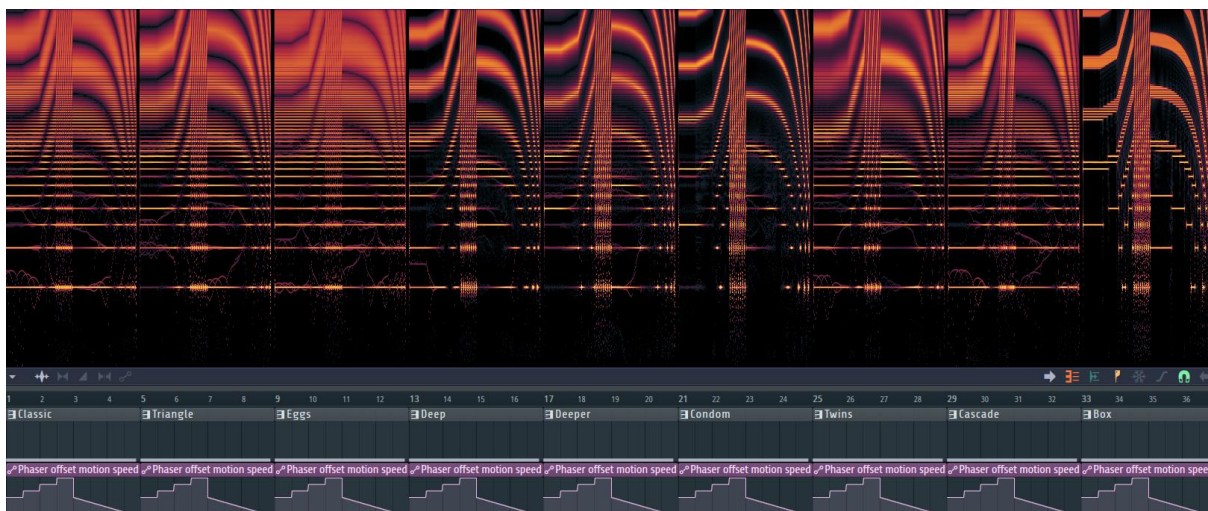


In the project <https://www.malmgren.nl/Harmor/Harmor-26.flp> we experiment with the phaser offset.

### Phaser offset motion speed

The knob *speed* is used to set the speed of which the phaser effect moves across the frequency spectrum. The default value is set to 32:00. When turning the knob to the right the minimum value is 1:00 and turning to the top position the value says “Still”. Right before still the values are increasing exponentially. On the other side the values are labeled the same. All the way to the left the minimum value is 1:00.

For the offset motion speed experiment, I decided to automate the knob and try various values. You find the project here: <https://www.malmgren.nl/Harmor/Harmor-27.flp>

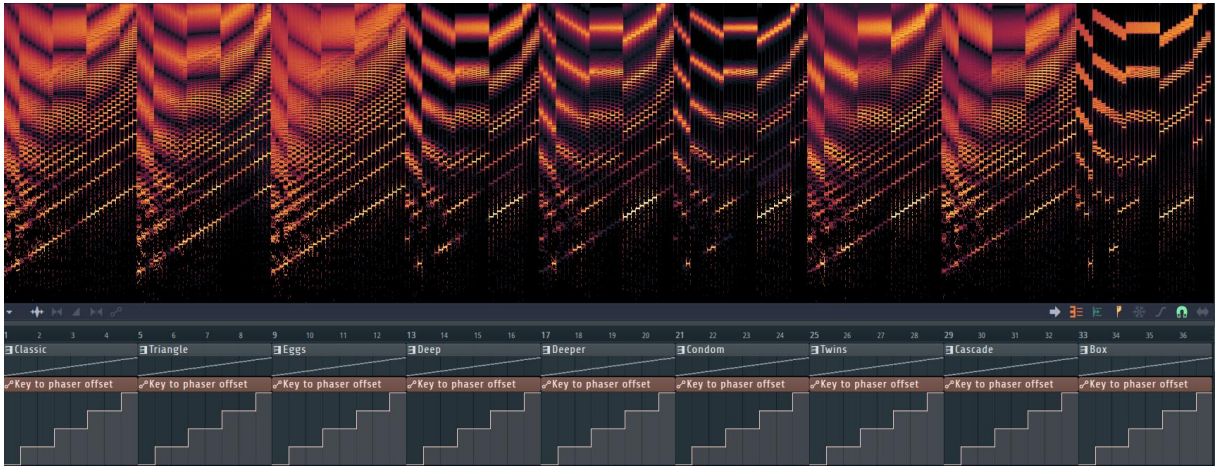


As you can see on the curve of the phaser offset motion speed automation clip start at still 50%. Then in three more steps I turned the knob to the minimum value at the right side. Then I started over at 50% again and went backwards but this time with a straight line.

It is motion speed that we are automating here so near *still* the motion is going slowly. It is when the knob nears the minimums that the motion gets quick. At the right side the motion of the phaser pattern goes up and on the left side the pattern goes down.

## Key to phaser offset

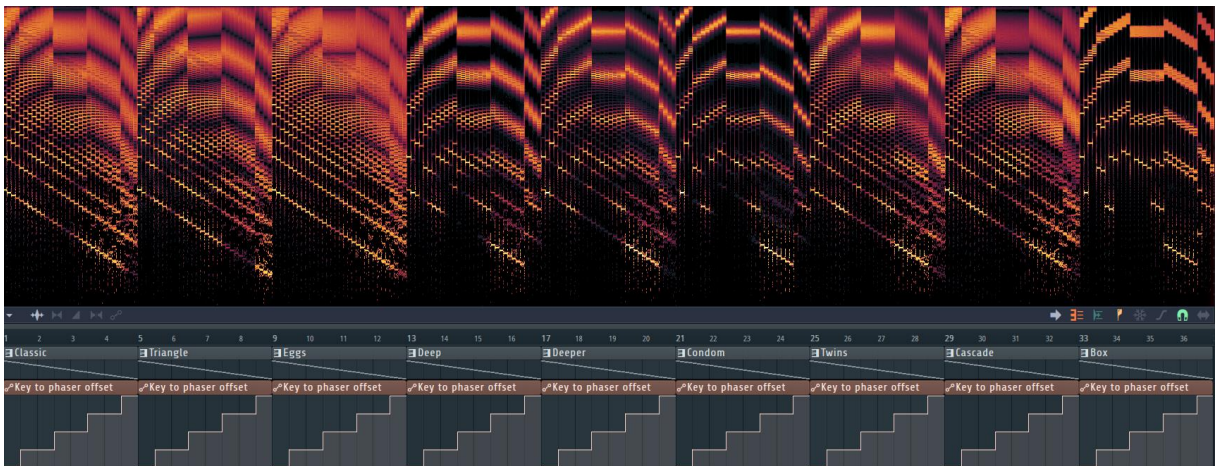
If it is of course possible to make the phaser time offset relate to the fundamental frequency! The keyboard tracking of phaser offset is similar to the keyboard tracking that we talked about in the section *Filter frequency*. When we try this, I decided to turn off the speed. Instead I set all the patterns to still. This way the fundamental frequency is the only modifier of the offset.



In this project <https://www.malmgren.nl/Harmor/Harmor-28.flp> the pattern consists of every note from B2 all the way up to F#5. Then I created an automation clip for the key to phaser offset knob. The automation starts at 0% corresponding to -200 on the keyboard tracking knob. Next step is 25% corresponding to -100, 0% is 0, 75% is 100, 100% is 200 on the keyboard tracking knob.

The slopes of the graph look like the slopes of the graph of the key to filter frequency. When the keyboard tracking is 0 then we get a horizontal slope. If we had speed engaged – but we don't, then we would see only the effect of the speed at that point. At 100% the slope follows the slope of the notes rather nicely. At 200 the slope is much steeper. At all the minus sides the opposite holds.

If this theory is correct we could play the notes from F#5 down to B2 to get the opposite! Luckily I was correct. See project <https://www.malmgren.nl/Harmor/Harmor-29.flp>



### Phaser scale

Octave mode makes the phaser pattern to expand in a logarithmic “curved” way. In Hz mode, the phaser pattern expands “in straight lines” linearly. Harmonic mode is like Hz mode in that the patterns it produces are linear but unlike both oct mode and Hz mode, for the harmonic mode to produce patterns that look like the other modes set width to 2 to 12 percent.



## Chapter 8. Prism

Prism shifts the position of partials from their original position based on the settings of the buttons in the prism section of Harmor. Prism do not shift the position of the fundamental frequency. It is thereby rather useless to apply prism on a pure sine wave.

### Prism amount

Here is <https://www.malmgren.nl/Harmor/Harmor-30.flp>, this project plays a saw and a square tone based on automation of timbre 1 & 2 mix. These notes are played: C3, F3, C4, F4, C5, G5, C6 and E6.

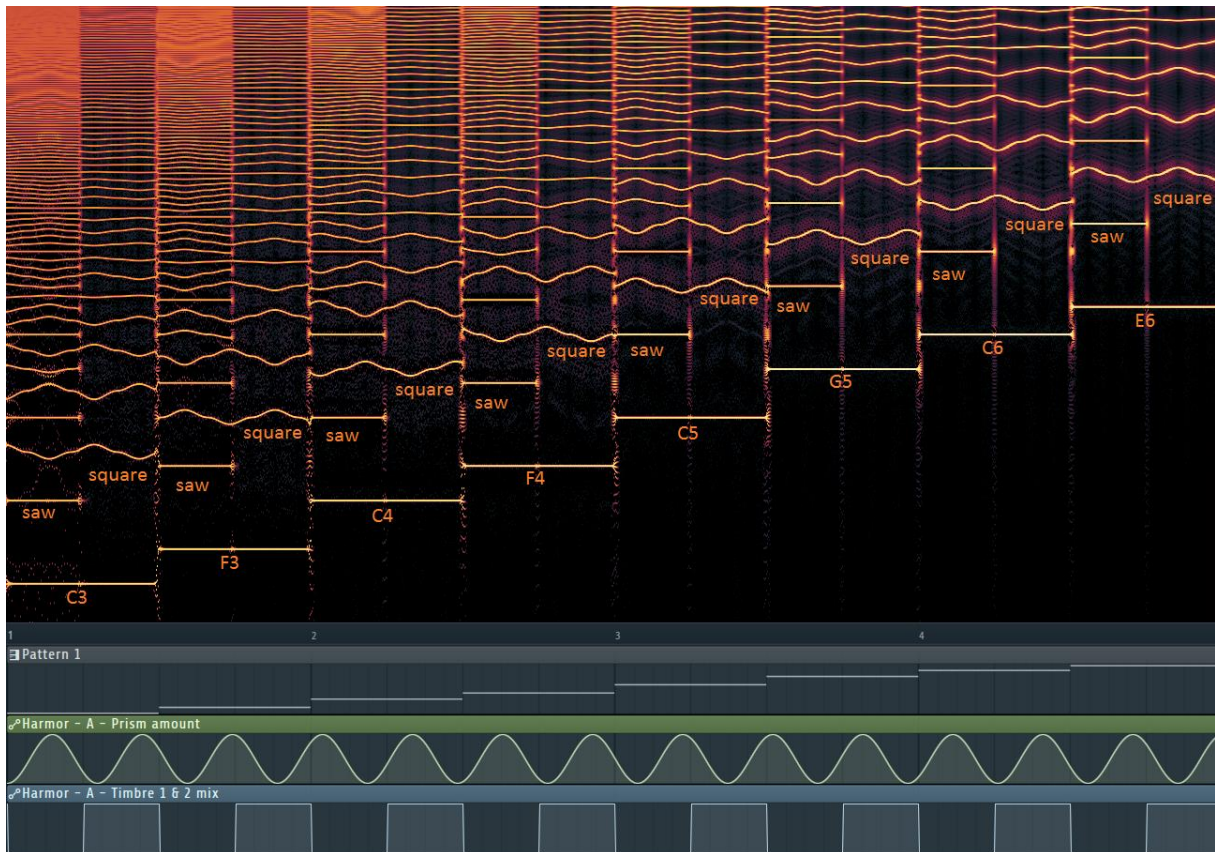


Image 39. Prism effect on saw and square wave form in "Natural scale".

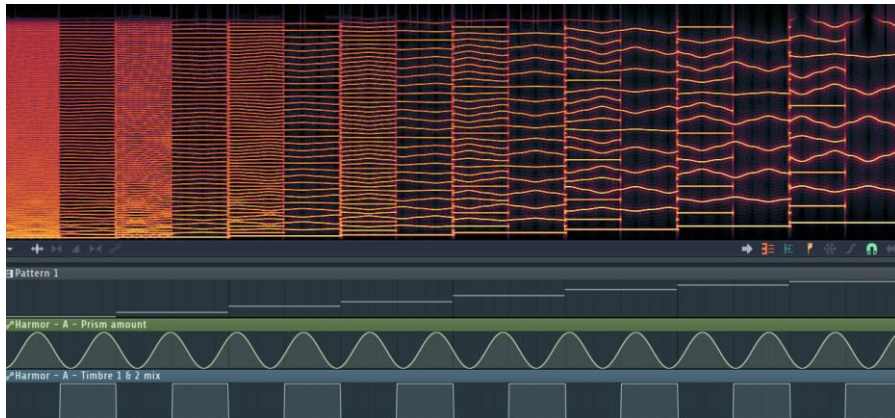
Prism never shifts the fundamental frequency. When the waveform has all partials, such as a saw, it is easy to see that the whole octave partials are also never shifted. The amount of alternation depends on the prism amount knob. It is the partials between the octaves that are alternating in the case of a saw wave. When we play the tone of a square wave then not all partials are playing. The basic principle holds except that the square wave never got any whole octave partials other than the fundamental frequency!

In the experiment here, I automated the prism amount. This means that when the automation curve pass through 0 amount (the middle) prism is doing no shifting.

### Prism add / multiply mode

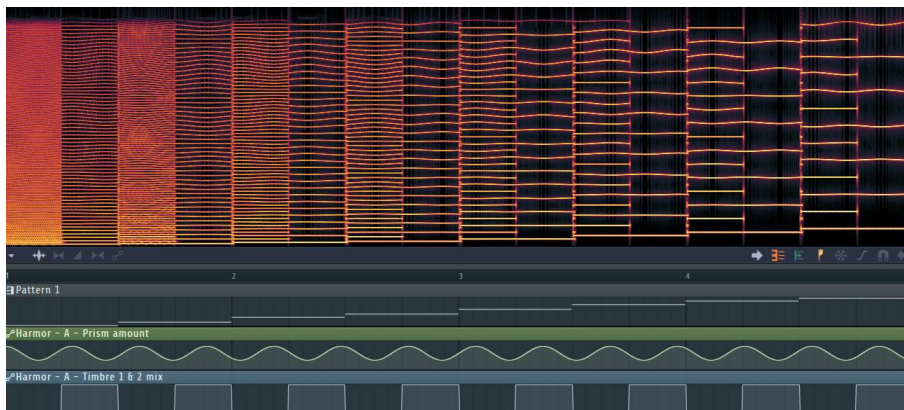
By default, prism is set to add mode and this is also the mode we tried so far. In add mode the shifting amount is the biggest near the fundamental and then the variation gets lesser as

the frequency increases. On the other hand, even though this is how the graph of Image 39 looks like the logarithmic nature works such that for higher frequencies much smaller variations has bigger impact. In Edison we can turn off natural scale and when doing so we can see the spectrum view “compensated” for the logarithmic scale.



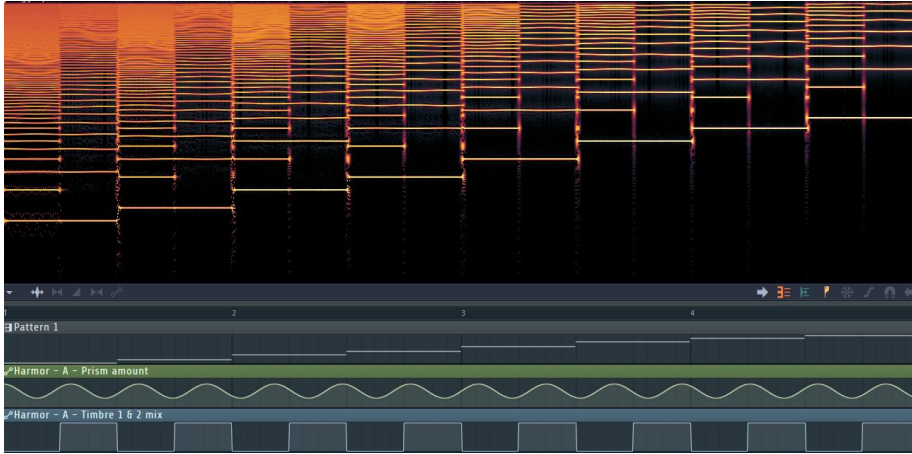
In this view mode of Edison we see that the add mode makes the effect of equal amplitude throughout all partials. Now it is time to switch to multiplication mode and here we notice that the partials near to the fundamental frequency are not shifted. In this experiment <https://www.malmgren.nl/Harmor/Harmor-31.flp> and for this experiment we considerably lower the amplitude of the sinus wave automation clip of the prism amount. This is so that we can see the impact of prism without partial values bouncing out of the graph.

As the frequencies gets higher we see more and more shifting effect.



With natural weighing, we can barely notice what is going on in the higher frequencies.

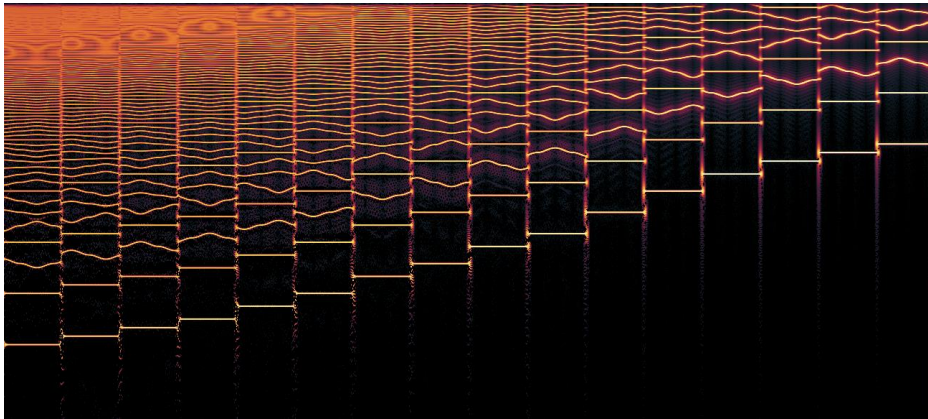




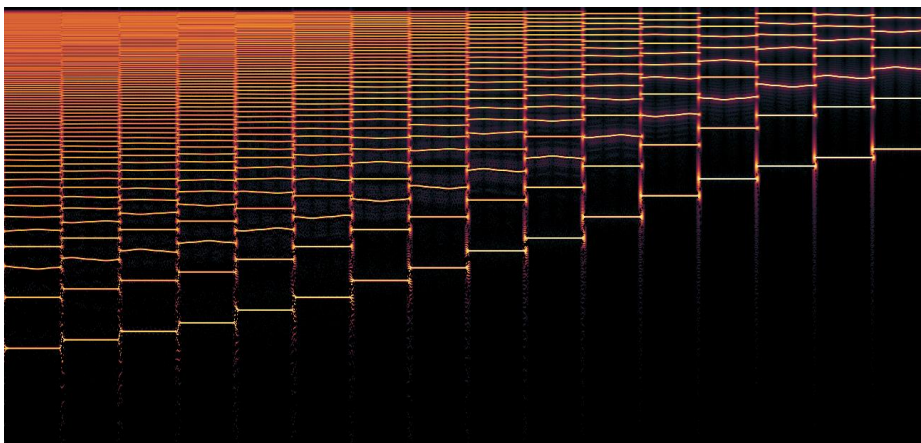
### Harmonic level to prism amount

Here is a graph of harmonic level to prism amount off. In this experiment, I play only saw wave forms. Also in this graph, I play a much lower notes and much higher notes.

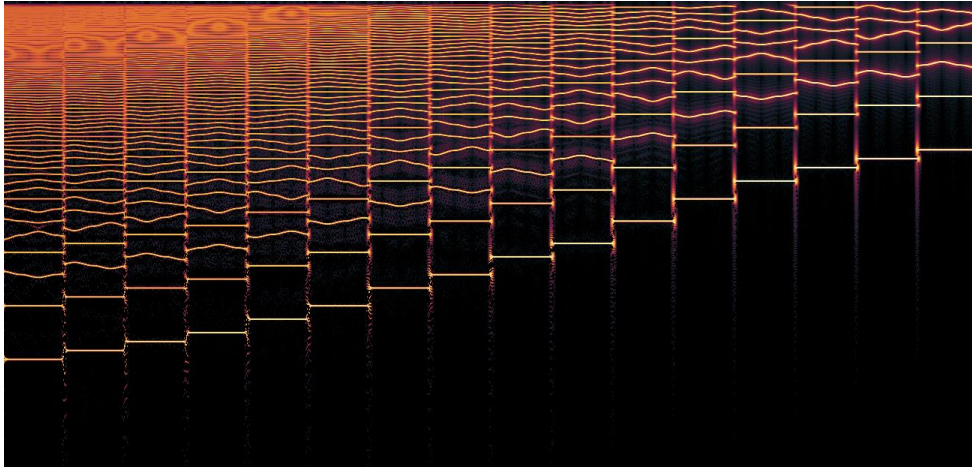
<https://www.malmgren.nl/Harmor/Harmor-32.flp>.



Here I play the Harmonic level to prism amount on.



Each partials distance to the fundamental frequency determines how much prism shift it will receive. We see the same amount on all third partial of all different notes.



Here we play harmonic level to prism amount inverted. In this setting, the longer the distance is to the fundamental frequency the higher prism amount that partial will get.

## Chapter 9. Harmonizer

Harmonizer is an effect that amplify some partials and quiet others. Harmonics around the fundamental frequency are duplicated and the duplicated harmonics are transposed to new frequencies according to algorithms modified by the shift and gap controls.



### Harmonizer mix

The harmonizer mix, *AMT*, specifies how much harmonizing effect will modify the partials.

### Harmonizer width

Harmonizer width, *WIDTH*, specifies how high in the frequency spectrum counted from the fundamental frequency that the effect will be applied. Harmonizer strength, *str*, specifies how strong the effect is. Strength and amount are kind of competing entities however if you reduce amount to 0% there is no patterns produced. If you have amount and reduce strength, there will be a pattern but it will be slightly less pronounced.

Finally, we got the harmonizer pattern control. This is two by two, one digit dials with a sign in the middle. The sign can be '+' for plus and 'x' multiplication. When rolling one of the yellow dials, *offset*, you can set it to a value from 0 to 9. When rolling the white dial, *shift*, you can set it to a value of 1 to 8 with '+' or 1 to 8 with 'x'.

To find out what this thing is doing the best is to set *AMT*, *WIDTH* and *str* to the maximum values. Then set the dials to 0x1 on both shift and gap. This way the effect is on full power but there is no pattern. Then we can decode the algorithms of the dials separately from each other. We can play a note, a saw wave form, and view the visual feedback. Perhaps make the visual feedback **extra wide** with the trick we learned at the end of *Chapter 2*.

*Additive/Subtractive synthesis*. Another useful trick is to turn on Hz mode on the visual feedback. This is handy when we want to count partials.

There must be enough partials going into Harmonizer before it can clone the partials and produce patterns. A sine wave normally has too few partials for Harmonizer to produce any patterns.

The screenshots in this section are all made from project <https://www.malmgren.nl/Harmor/Harmor-33.flp>.

### Harmonizer Shift Step

When gap is set to 0x1, see Image 40, we will start to experiment with shift. It turns out that shift in 'x' mode starts the effect of amplifying a partial at  $offset + 1 + (step - 1) \times 2$ :

Step	Effect base partial, starts at partial number
1	No effect.
2	3
3	5
4	7
5	9
And so on.	And so on.

The next partial in the frequency to be amplified is the previous partial number, minus 1, times the step. This goes on up into the frequency heaven.

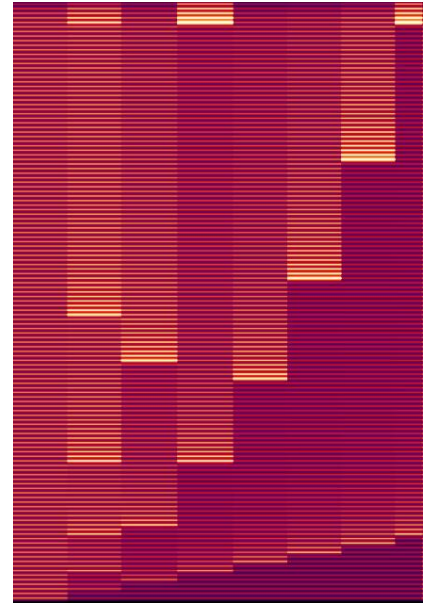


Image 40. Step 1, 2, 3...8 of Harmonizer in 'x' mode

The plus mode, see Image 41, is slightly more easy to understand. Unlike 0x1 where the effect is turned off at step 1, here at 0+1 the effect is noticeable already at step 1. The base partial for the effect is the step. At step 5 the fifth partial is the first that is amplified. The next partial a multiple of 5 for step 5 and so on. This is repeated about 7 times.

### Harmonizer Gap Step

The gap step harmonizing pattern algorithm is fascinating. This pattern not only accentuate partials it also lowers the amplitude of some partials.

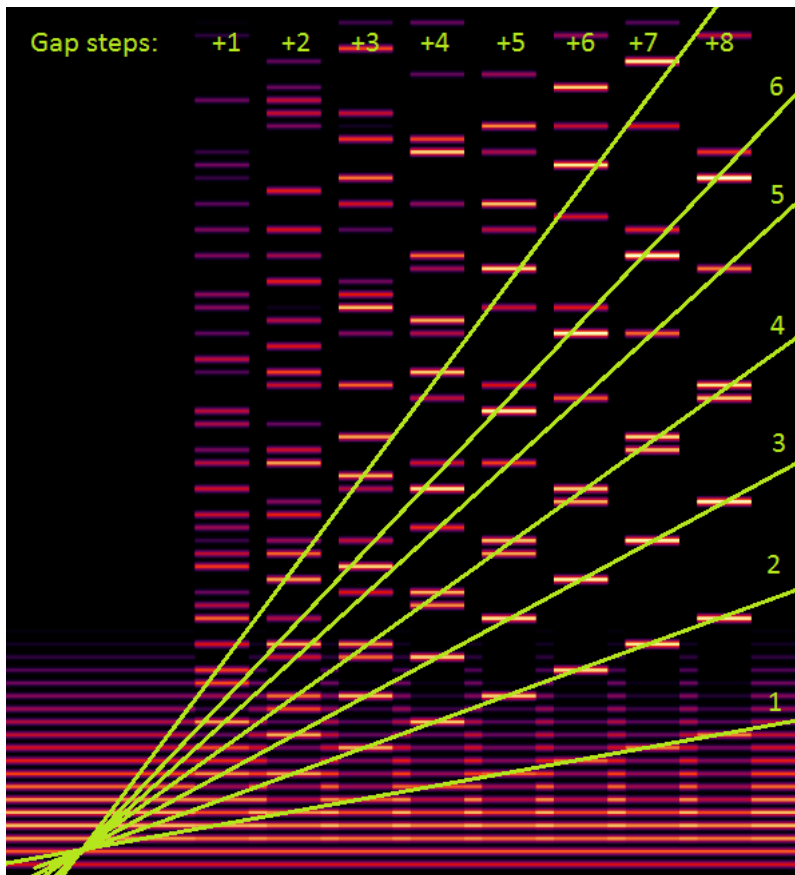


Image 42. The patterns of the gap-step '+' algorithm. This image is from the built in visual feedback in Hertz mode and extra wide.

Were shift step had a regular repeating pattern gap step has something more intriguing. The bottom most three green lines has a single accentuated partial as you can see in Image 42. The fourth line is made with double partials. The fifth line is a single partial again and from the sixth line and on it looks like most lines are double partials and they are wider apart for higher frequencies.

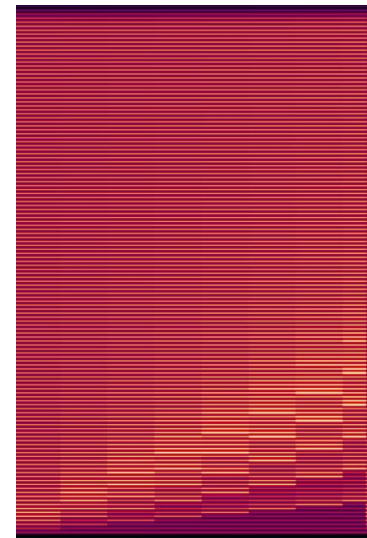


Image 41. Harmonizer shift '+' mode.

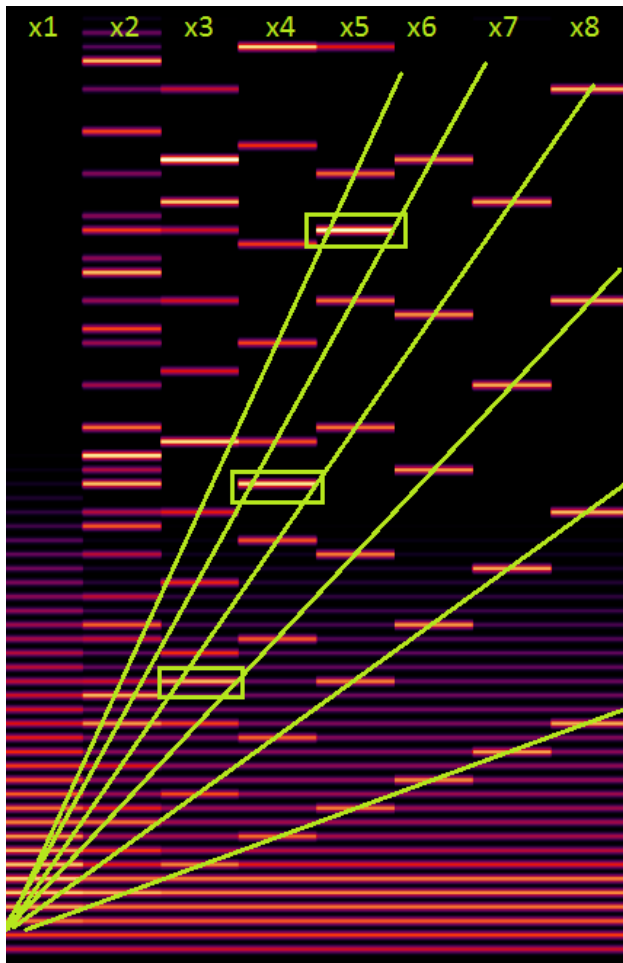


Image 43. Image of the gap step 'x' algorithm.

The gap step 'x' algorithm starts the pattern at partial  $offset + 3 + (step - 1) \times 2$  where step is 2 or higher. Then the next partial of the pattern is repeated every  $1 + (step - 1) \times 2$  partial from step 3 and higher. For step two the pattern is repeated every 3 steps. Then please notice rectangles around extra partials inserted in the middle between two repetitions at the number of steps count. At step four, at four accentuated partials up, there is an extra partial inserted. At step five the extra partial is at five accentuated partials up and so on.



## Chapter 10. Tremolo

Tremolo is the Italian word for trembling effect. Many music terms originate from Italian because Italy was important in the early music business.

Harmors Tremolo section simulates the effect like rotating speakers in electric organs. In the late 1937 Donald Leslie invented these speakers. For more information about his effort please read the WikiPedia article about the Leslie Speaker.

The Tremolo effect is achieved by manipulating the amplitude and phase between the left and right channel. Since the Tremolo effect is a kind of stereo effect it is useful to view the signal with different stereo views of WaveCandy. I created a project that you can use to experiment with Tremolo: <https://www.malmgren.nl/Harmor/Harmor-34.flp>.

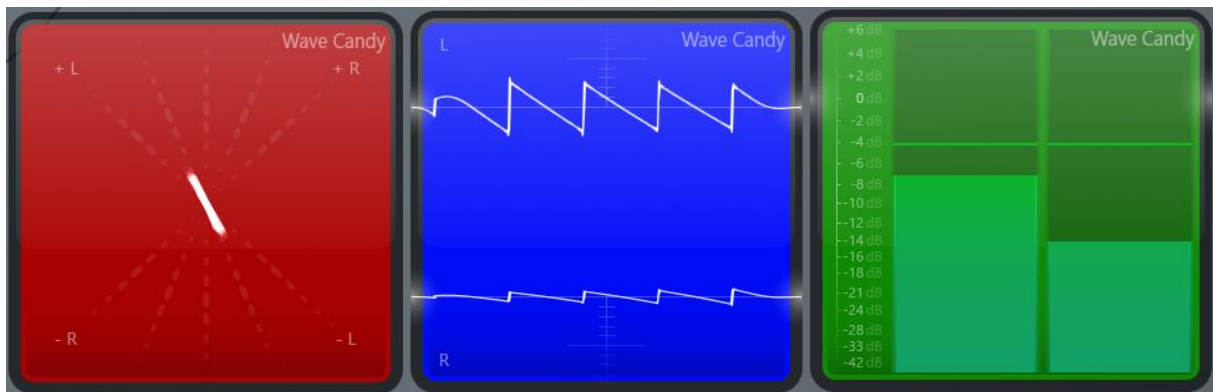


Image 44. Here we play a saw wave and display it with three different wave candy windows. Red = Vectorscop, Blue = Oscilloscope, Green = Meter

In Image 44 we can see a line when we play a saw waveform. If we played a square waveform we would see two dots. The center of the line pivots at the center of the diagram. A completely vertical line is the result of playing for example saw waveform in mono.

### Tremolo speed

The tremolo speed, *speed*, specifies how quick the rotation of the end points are. The minimum value is 16 ms and maximum is 16 seconds. If you change the waveform to 16 ms, then the vector scope will move the dots so fast that you can see a diagram emerge of how the signal is manipulated especially when you play a square waveform because then the signal is “spending most time” at the end-points.

### Tremolo gap

Tremolo gap, *gap*, specifies how the signal moves. In the vector scope the upper end moves counter clockwise when gap is between 0 and 180.

Between 180 and 360 the upper end moves clockwise. When gap is 0 there is no stereo effect at all, the signal is mono.

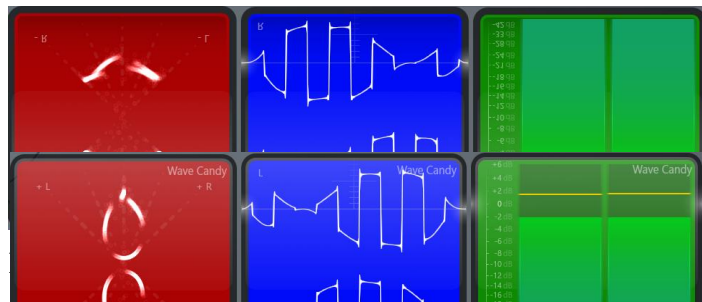


Image 45. Tremolo gap is near 0.

When gap is near 0 the shape is droplet shaped. When gap is near 180 the shape is “like a smiley”.

### Tremolo depth

The tremolo depth knob, *depth*, specifies how much of the original signal that the effect will be applied to. At 0% tremolo depth, the signal will stay at the highest and lowest spot of the diagram. As the depth increases the shape becomes bigger and bigger.

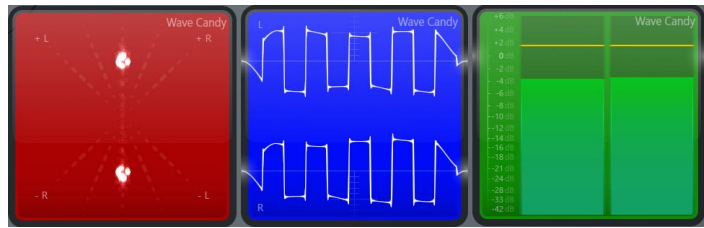


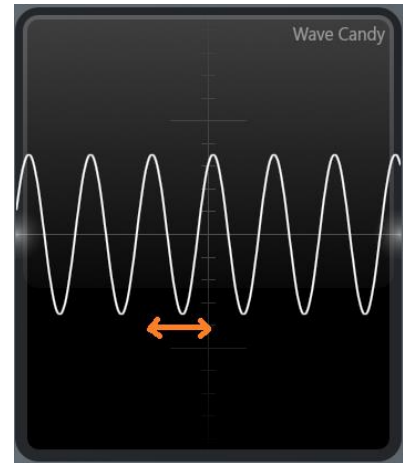
Image 47. Tremolo depth at 17%.



## Chapter 11. Vibrato

Vibrato is the Italian word for vibrate in past participle.

The vibrato effect create vibration by moving the start phase back and forth. This has the advantage that the amplitude stays unaltered. The project <https://www.malmgren.nl/Harmor/Harmor-35.flp> displays a WaveCandy oscilloscope diagram where you can see the waveform moving back and forth.



### Pitch vibrato depth

When the pitch vibrato depth, *depth*, is all the way to the left the effect is turned off. The maximum value all the way to the right is 100 cents. The depth knob specifies how far the is back and forth motion is for each cycle of the vibrato.

### Pitch vibrato speed

The pitch vibrato speed knob, *speed*, specifies how quick the vibration oscillates. The quickest setting is all the way to the left of 125 ms and the slowest setting all the way to the right at 250 ms.

## Chapter 12. Legato and Portamento

Legato is the Italian music term for “tied together” meaning that the musician, or in our case the synth, transition from one note to the other while both notes are playing.

Portamento is also a word that originate from Italian and it means “carrying” and this is the music term for pitch sliding between notes that not necessary needs to be playing at the same time.

These two features of Harmor, Legato and Portamento, are operated from two different parts of Harmor: 12a and 12b. Two parts of harmor, two different effects, there are plenty of room for confusions we need to avoid so I decided to start this chapter by talking about only about Legato. Then when that is done I will come back and talk only about portamento.



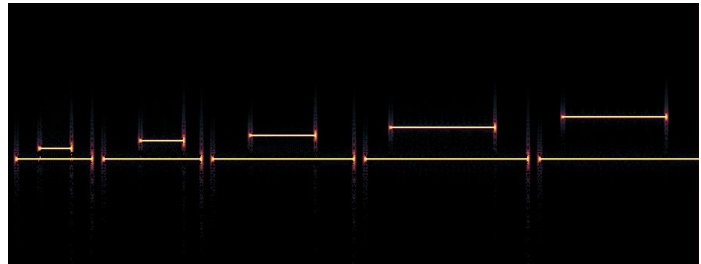
Up until now we primarily talked about the partials from the upper brown area of Harmor. From this chapter and onwards we will talk more about what happens after the partials has been converted into regular sound signals, i.e. the knobs and controls in the black section of Harmor.

## Legato

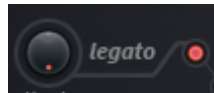
To accompany the explanation of Legato I created a project in Harmor with a little simple tune in C Major of simple sine waves, hence one single partial per note:

<https://www.malmgren.nl/Harmor/Harmor-36.flp>.

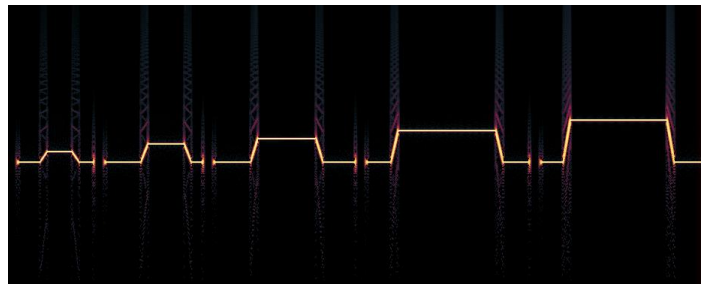
This is how it looks like in Edison:



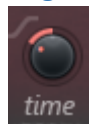
## Legato mode switch



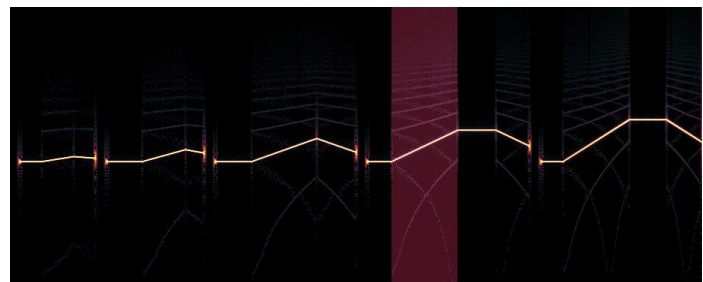
The legato mode switch turns on the mode where notes are tied together when they are played at the same time. With other words this makes notes slide over from one note to another when they are overlapping midi notes. The legato effect transition back to the original note when the overlapping note is done playing.



## Legato time



The portamento / legato time specifies how long time it takes to finish the slide. The left side of this knob specifies the time in milliseconds. The longest slide is when the knob is set all the way to the left 1500 ms and the smallest time is when the knob is at the top-position – instant sliding. Here in the image the slide is set to 643 ms. It is possible to set Edison to display milliseconds. The red area here is 643 ms.



The right half of the time knob sets the time based on the time of the project.

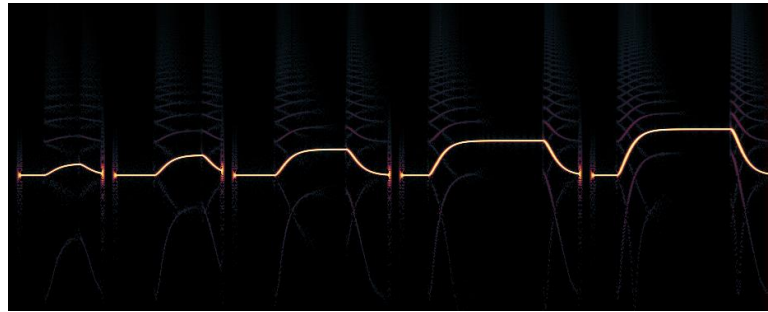
## Linear / Logarithmic legato



The examples so far were linear legato.

When clicking the button under the legato tag we can switch between linear and logarithmic legato transition.

The logarithmic legato can transition to the new note via a curve where the linear transition via a straight line.

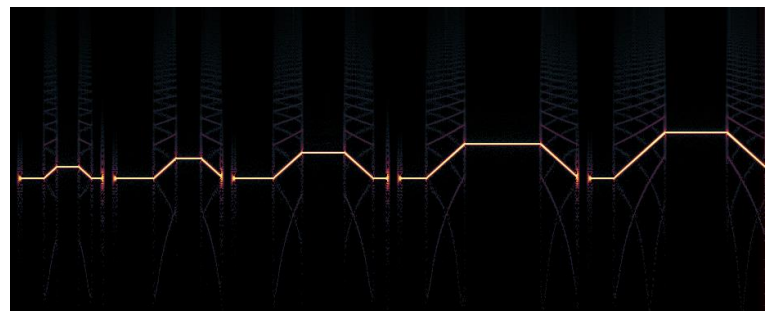


## Fixed / Variable legato time



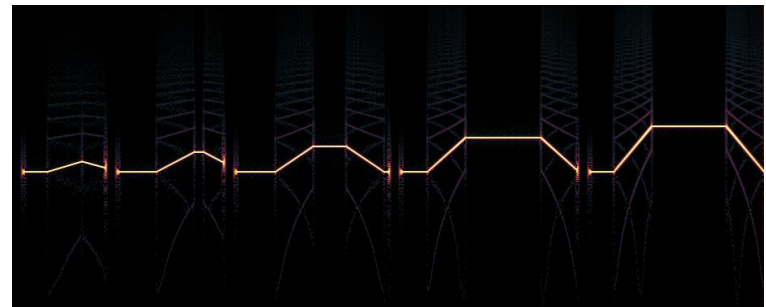
It is possible to make the transition length depend on the distance between the notes, this is variable legato time.

When the rightmost icon is highlighted Harmor applies variable legato time. With a greater distance, it takes longer time to reach the end note.

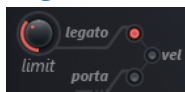


It is much easier to display the effect of this with linear mode because then we can easier see when the transition ends.

When the left icon is highlighted Harmor applies the same time regardless of the distance between the notes.

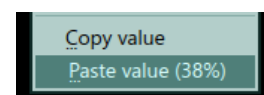
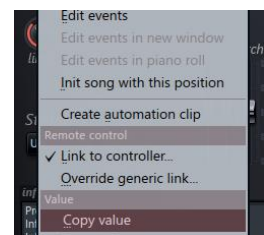


## Legato limit

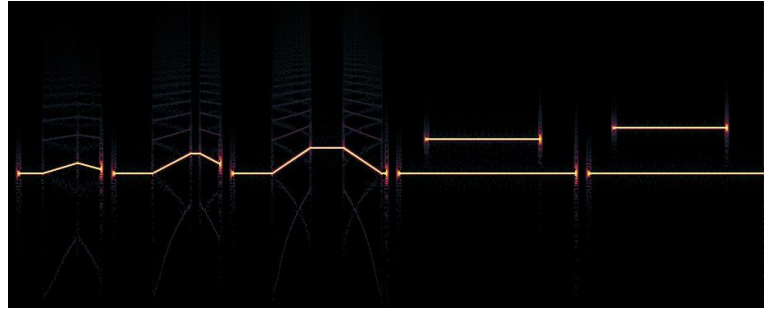


Legato limit, *limit*, specifies the maximum distance allowed for the effect. When the knob is turned all the way to the left the limit is turned off. The maximum value all the way to the right is 2400 cents.

Here is a nice moment for a little extra tip on how to copy and paste values from a knob to an automation point. Suppose you want explicitly 900 cents in the point of the automation clip. Then set the limit to 900 cents and then right-click the knob and select Copy Value. Then navigate to the automation clip connected to the knob and the point you want to set, right-click the point, and select Paste Value. This way you do not need to do the math converting the 900 cents into a percentage value, this is done for you!

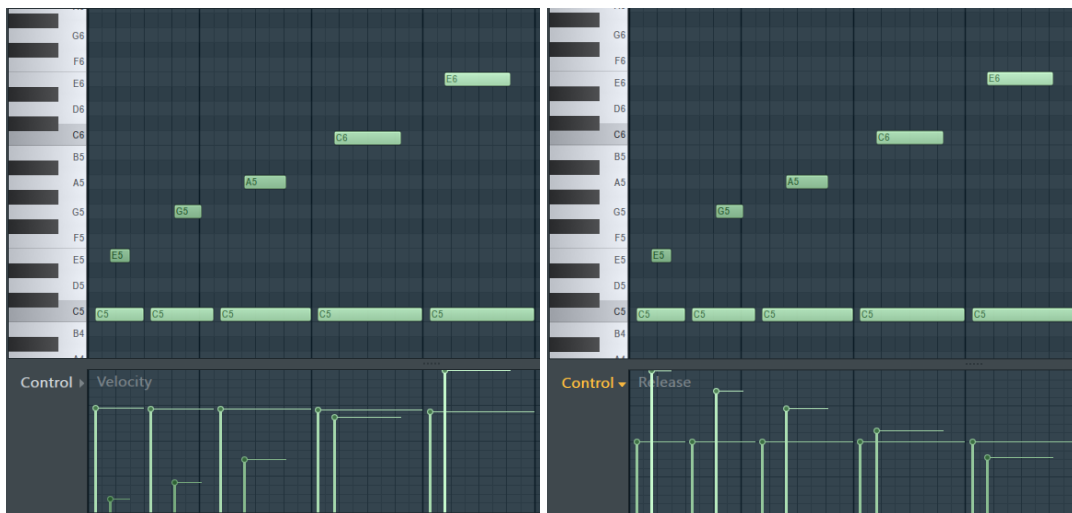


Here you can see the result of legato pitch limit set to 900 cents. The fourth and fifth note “disengage” the legato effect because the distance between the notes was beyond 900 cents.

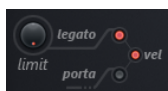


### Velocity / Release to Legato time

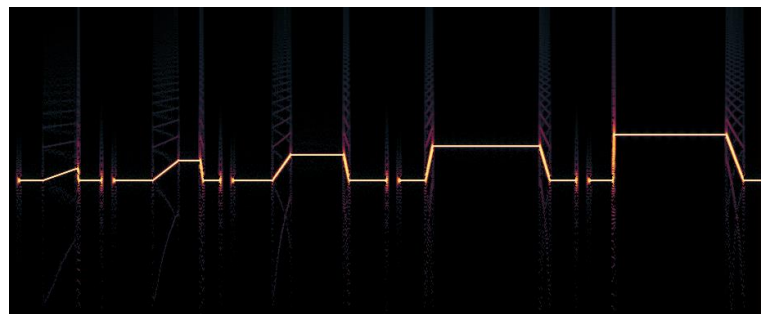
For this pattern, I programmed the velocity in the piano roll in such way that the higher note has ever higher velocity.



But that is not all, I also programmed the release of the higher note to lower and lower values.

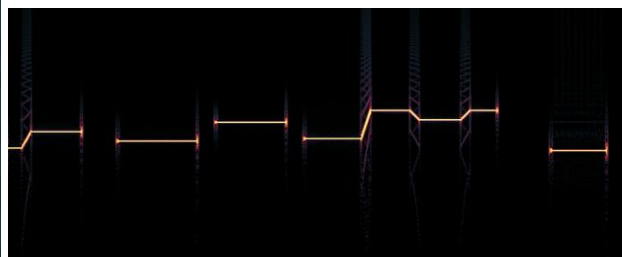


Now when we turn on the velocity switch, *vel*, of the legato effect it is possible to see what outcome this has on the notes when playing the effect.



### Portamento

Portamento comes into action between adjacent notes as well as overlapping notes. To experiment with this I created a project <https://www.malmgren.nl/Harmor/Harmor-37.flp> with this pattern in E minor:

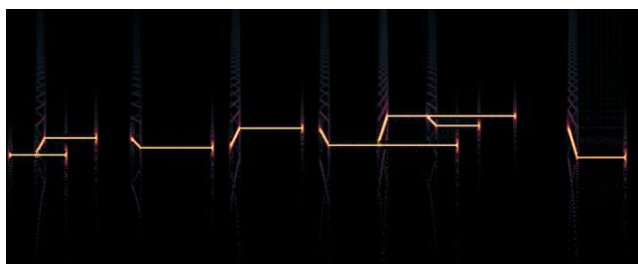


When playing the notes with legato (the mode we discussed in the previous chapter) that gives the result as we can see here above to the right.

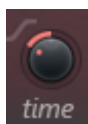
### Portamento mode switch



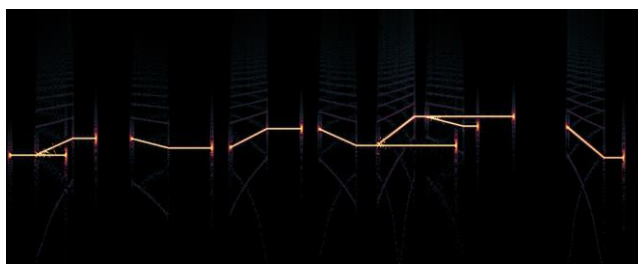
Here are the same notes played with the portamento effect and 100 ms transition time. It is not so that when notes overlap only one note can be heard. The next note will transition from a previous note – even if there was a pause between the notes. Portamento will not transition back to the original note when a note is done playing. The effect only takes place at the front of the notes.



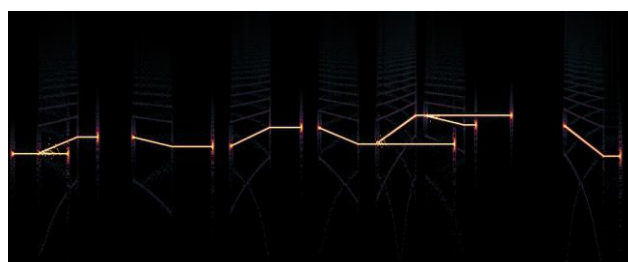
### Portamento time



Here is the same pattern played with portamento of 401 ms. It takes longer time for the transitions to take place.



Here the pattern is played with 4:00 project time.

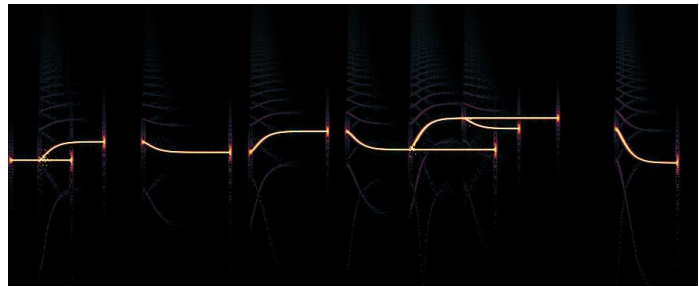




## Linear / Logarithmic portamento



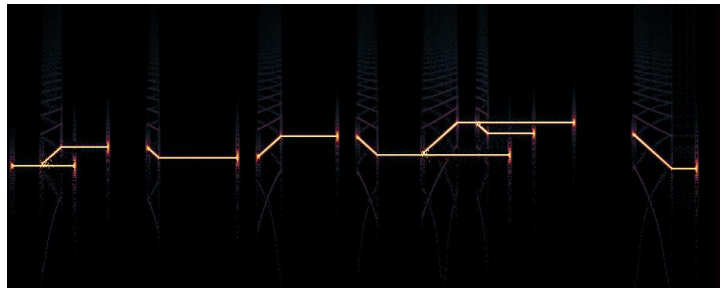
Here is the portamento played with the logarithmic switch turned on.



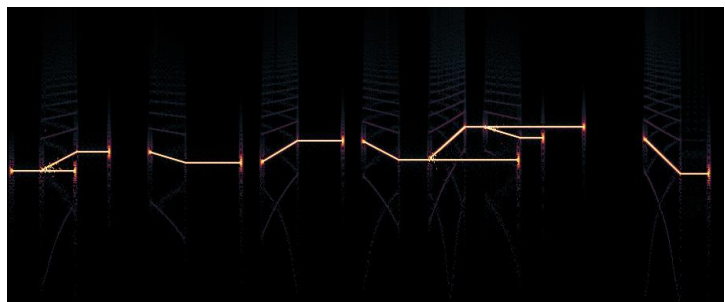
## Fixed / Variable legato time



The transition distance can be used to determine how long time the transition will take. When the rightmost icon is highlighted Harmor applies variable time.

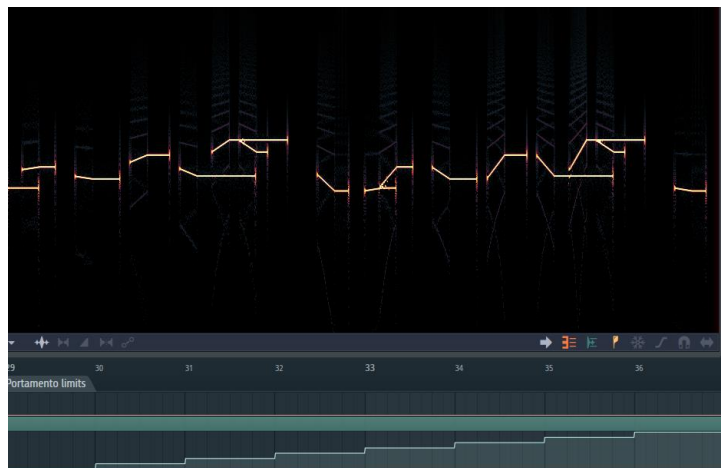


Here is how it looks with a fixed portamento time.



## Portamento pitch limit

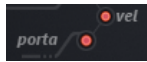
The portamento pitch limit is entirely different compared to legato limit. Legato limit specifies where the effect is disengaged. Portamento limit specifies maximum transition time. The image to the right displays the pattern being played two times where the portamento limit is increased in eight steps.



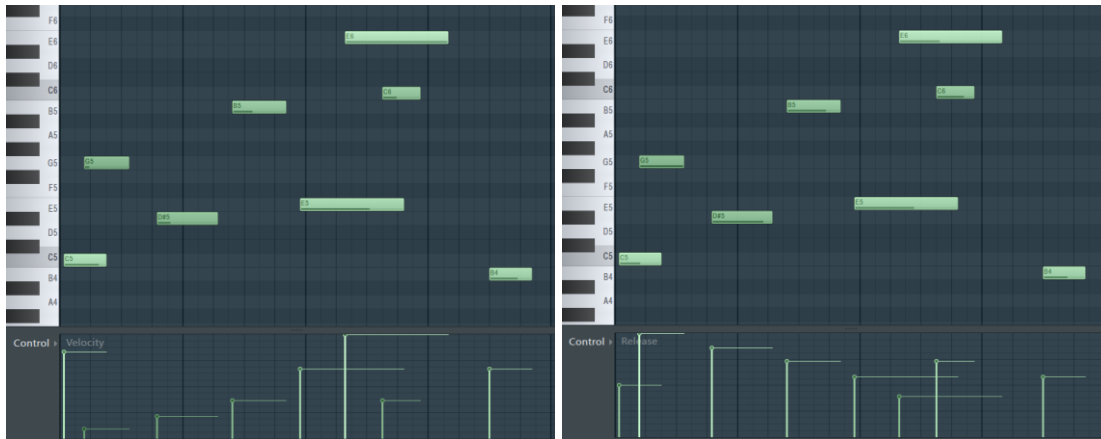
With smaller values the movement is more moderate. With greater values, we see the full pattern more like the unlimited portamento effect. I think that because this effect operates on the partial level the pitch limit button is placed in the upper part of the synth user interface.



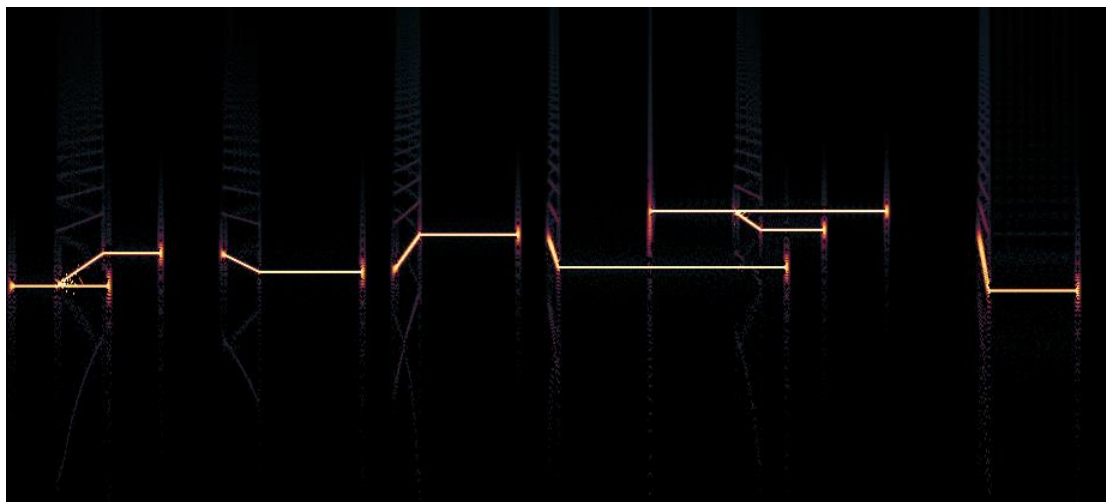
## Velocity / release to portamento time



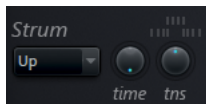
When turning on the *vel* switch the velocity and release time of the midi notes can affect the portamento time.



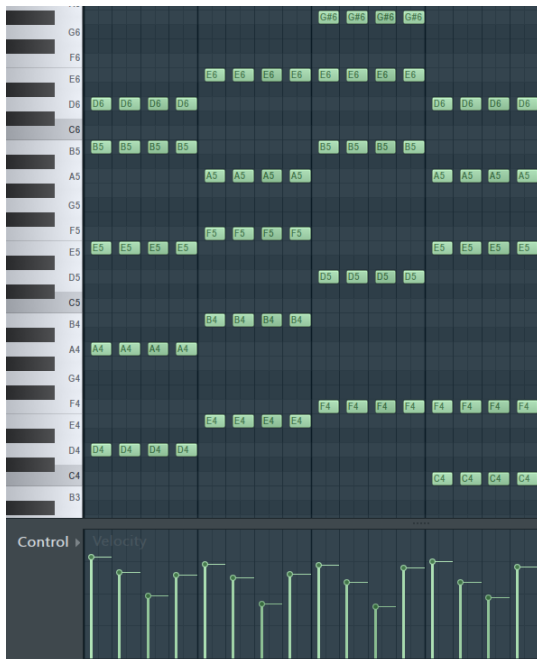
Here below you can see the result of playing the notes with the *vel* switch turned on.



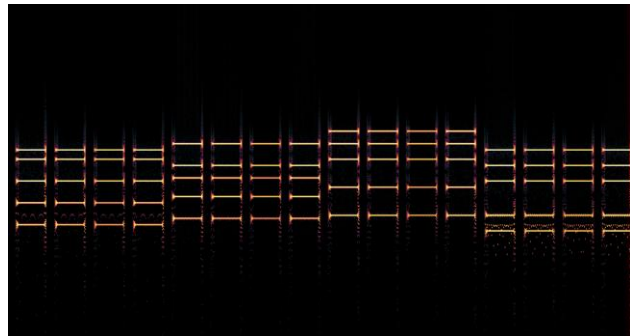
## Chapter 13. Strum



The strum effect makes the notes of a chord to start playing at different times. To experiment with strum I created a test project that you can find here: <https://www.malmgren.nl/Harmor/Harmor-38.flp> The pattern for this is a simple melody in A minor where Harmor is playing a sine wave. This way we can easily see each note of the pattern and how it behaves when we change the settings of the strum effect.



Before we begin with strum please note that I adjusted the velocity of these notes. When the switch velocity to note volume is on then the notes are played louder with higher velocity. When we play the melody, and record it with Edison then the spectrum diagram looks very much like the piano roll pattern.



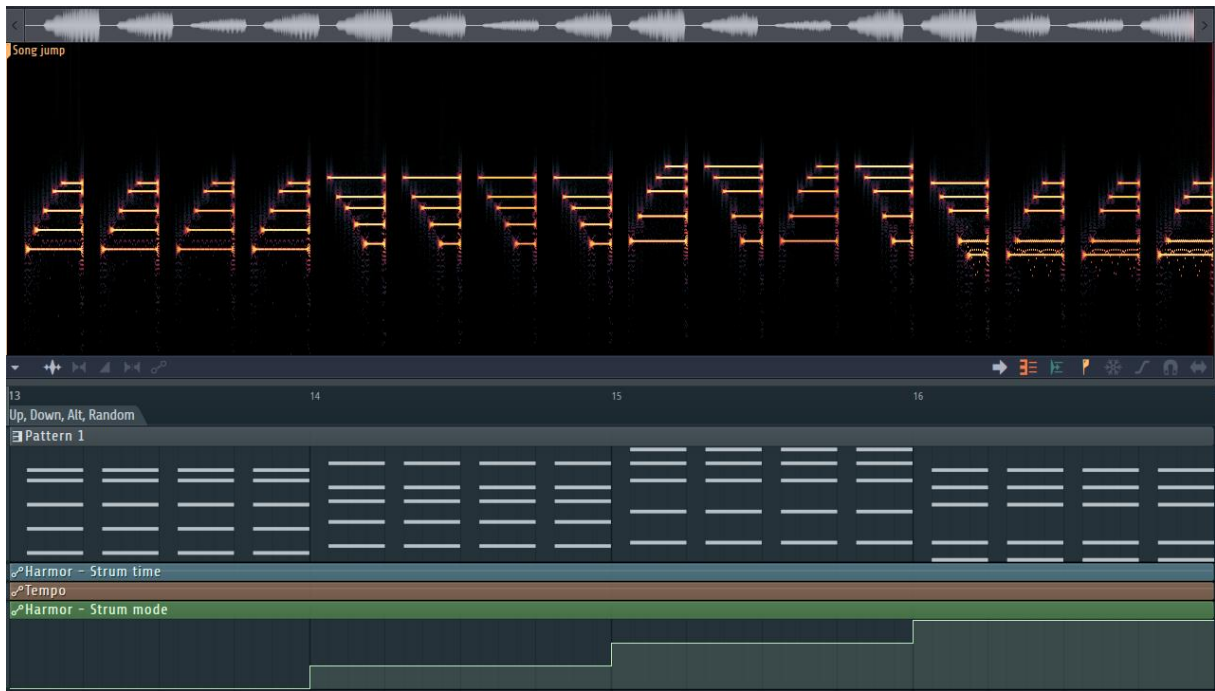
### Strum time

Strum time specifies how long time it will take for the notes to start playing. The minimum value is 0% and max is 100%. Of what? I tried several suspects like the BPM setting of the project or the length of the notes but I cannot establish the exact entity for this. Just suppose we set the time relative to a universal entity. At 100% it takes longer time to walk the strum mode pattern. When the time has a lower value then the strum mode pattern is engaged more quickly.

### Strum mode

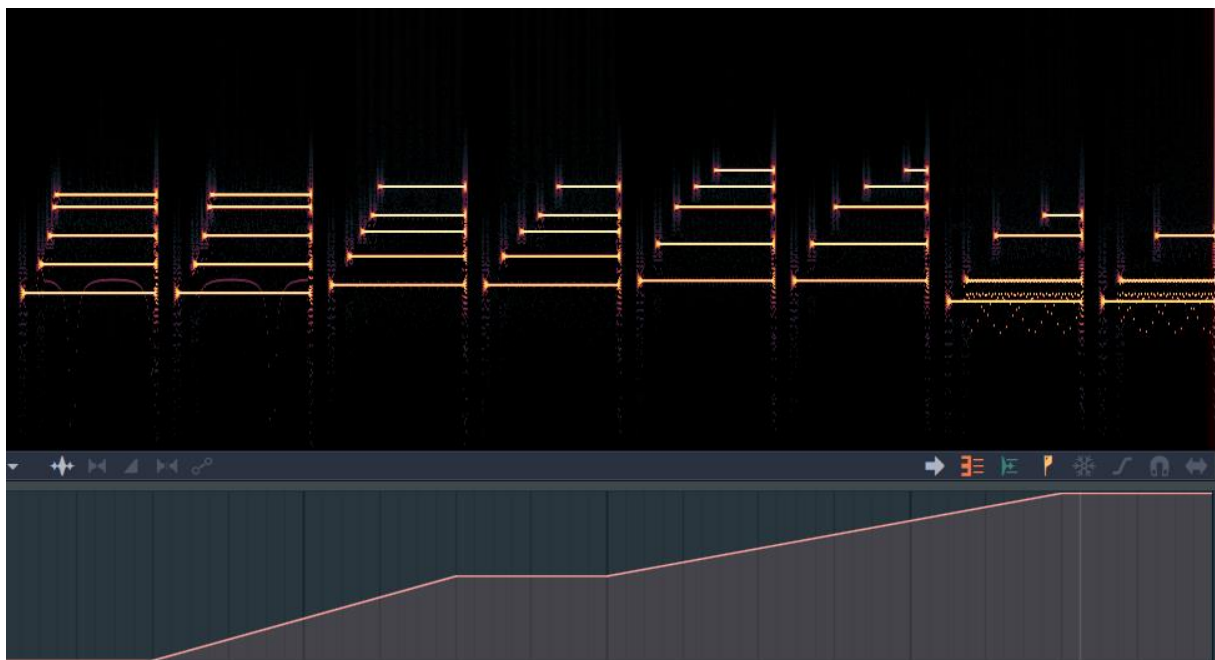
There are four strum mode patterns:

- Up. Engage the lowest midi note first and then gradually engage the next higher note one after another.
- Down. Engage the highest midi note first and then gradually engage the next lower note one after another.
- Alt. Alternates between the two previous modes.
- Random. Randomly select one of the two first modes.



## Strum tension

The strum tension knob specifies how the notes are engaged. Here below you can see the strum mode up where the tension is starting at -100% and then staying at 0% near the middle and then continuing to 100%. With a negative tension value, it takes longer time for the second note to be played but thereafter they play quicker. This effect planes around 0% where time between the notes is almost identical. With positive values the second note is played quicker but the notes thereafter take longer and longer time before they play.



## Chapter 14. Part A/B



Harmor has a certain duality to its design. There are two timbres, two filter sections and there are two synthesis banks: part A and part B.

### Part selector

To switch between the banks, click the A / B button, the part selector. All knobs all settings changes to the values specific to that part.

### Part enable switches

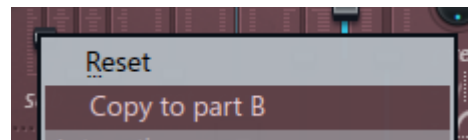
In the image above under A you see a yellow button. This indicates that part A is enabled. Below B you can see a circle, this indicates that part B is disabled. If you disable both parts, then you turned off Harmor!

### Part A/B mix

Stretched between the two enable switches you find a slider making it possible to mix between the two parts.

### Instant replication of the user's changes

Below the mix slider you find the replication switch, *GUI-link switch*. When turned on any change of a control made by the user in one part is replicated to the other part. You can think of it as if the user interface after every change of a knob automatically does a right click and selects “Copy to the other part”.



I think it is a bit unfortunate that the term “link” was used here. It implies that things are connected in a logical sense but as I can see it there is none, just replication of the user's actions when the switch is on. For real linking right click and select “Link to controller”.

## Chapter 15. Distortion



This is the first chapter to discuss the tabs in the lower section of Harmor. Before we get too serious please look at how the blue FX knob shines so beautifully on the effects

panel below it!!!



In the upper part of Harmor everything is about controlling the manipulation of partials. In many lower sections, we will continue to manipulate the partials while other parts in the lower section manipulate the signal after it was converted to sound. The effects panel operate on the sound signal.



In the main area of Harmor you find two sliders making it possible to change the volume of the sound going into the effects section, with the *pre*-slider, and it is possible to change the volume of the sound after the effects has been applied with the *post*-slider.

Now we are going to talk about distortion. I decided to explain distortion in Harmor by “reverse engineering” the effects outside Harmor with the WaveShaper plugin.

### Waveshaping theory

During my experiments, I found that the triangle waveform is the most suitable waveform for investigating how distortion works. I created my triangle waveform with Sytrus and then I exported it as a single cycle wave-form and that file I imported into Harmor.

The theory of wave shaping is that we can use a two-dimensional graph to describe the transformation of a waveform. The horizontal axis represents the input. The vertical axis represents the output. In Image 48 the blue area shows an example of a transformation graph. At the diagonal line from the bottom left to the upper right input is sent untransformed to the output.

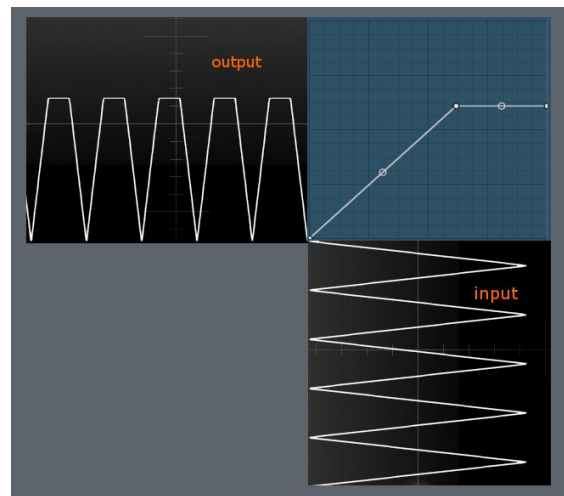


Image 48. Waveshaping.

I set up a test project for experimenting with WaveShaper in this project <https://www.malmgren.nl/Harmor/Harmor-39.flp> where the left WaveCandy oscilloscope is displaying the input and the right oscilloscope displays the output and in the middle the WaveShaper user interface.



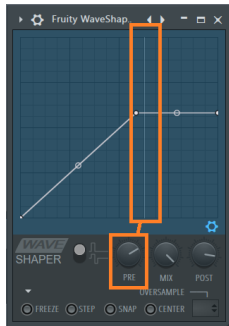
## Unipolar / bipolar mode in WaveShaper



In WaveShaper things are slightly more versatile than the theory. There is a button to switch between unipolar and bipolar mode. In the unipolar mode, the entire transformation graph represents the transformation above the zero-crossing line. That is the center line of the waveform. Then the same transformation takes place below the zero-crossing line.

In bipolar mode you can set different transformation for above and below the zero crossing line.

## Pre amp in WaveShaper



There is a knob pre amp, *PRE*, setting the input level. For WaveShaper to start transforming the waveform the input level needs to be sufficient. A thin vertical white line works as peak meter display. Adjust the input level so that the peak meter line reaches the appropriate part of the transformation graph.

## Mix level in WaveShaper

The mix level, *mix*, makes it possible to blend in a part of the original signal with the distorted signal.

## Post gain in WaveShaper

The knob *post* specifies the output level. With this knob, it is possible to amplify the signal after the distortion.

Now let's talk about the distortion controls inside Harmor. There is no wave shaping graph in Harmor.

## Distortion type



This is a drop-down menu with 11 types of distortion + "Off", by default distortion is turned off. I believe that you can understand the different distortion types as preprogrammed WaveShaper graphs. How these graphs could look like we will come to in a moment.

## Distortion amount

This slider is marked with *amt* and it goes from 0% up to 100%. The default value is 50%. At 0% this has the result that the effect is not activated. At 100% many times the distortion turns the signal into a square waveform or almost a square waveform.

## Distortion asymmetry / extra



The slider, *asym*, when moved up provides a mono modification to the sound. In the oscilloscope it looks like the upper part of the waveform is flattened and the lower part get less distorted, as displayed in the image to the left.





When moving the slider down a stereo distortion takes place. In the vector scope, it looks like a bent shape. In the oscilloscope you can see that the shape is flattened just like the mono version but the right signal is phase shifted 180 degrees and inverted. When we analyse signals like these and the oscilloscope is not displaying thin lines anymore then it can be a good idea to try stereo separating the image to see if that is more informative.

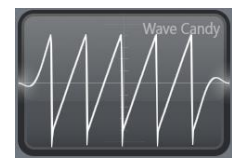
### Distortion wet volume

This slider, *wet*, sets the amplification of the distortion effect. Either that or it makes it possible to mix in the distortion with the undistorted signal. This is similar to the *post* knob in WaveShaper.

### Distortion mix



The *mix* slider set to +100% turn off the distortion. At minus 100% the signal is inverted. In the image to the left we see a saw waveform at 100% and to the right we see the same waveform at minus 100%. At 0% the



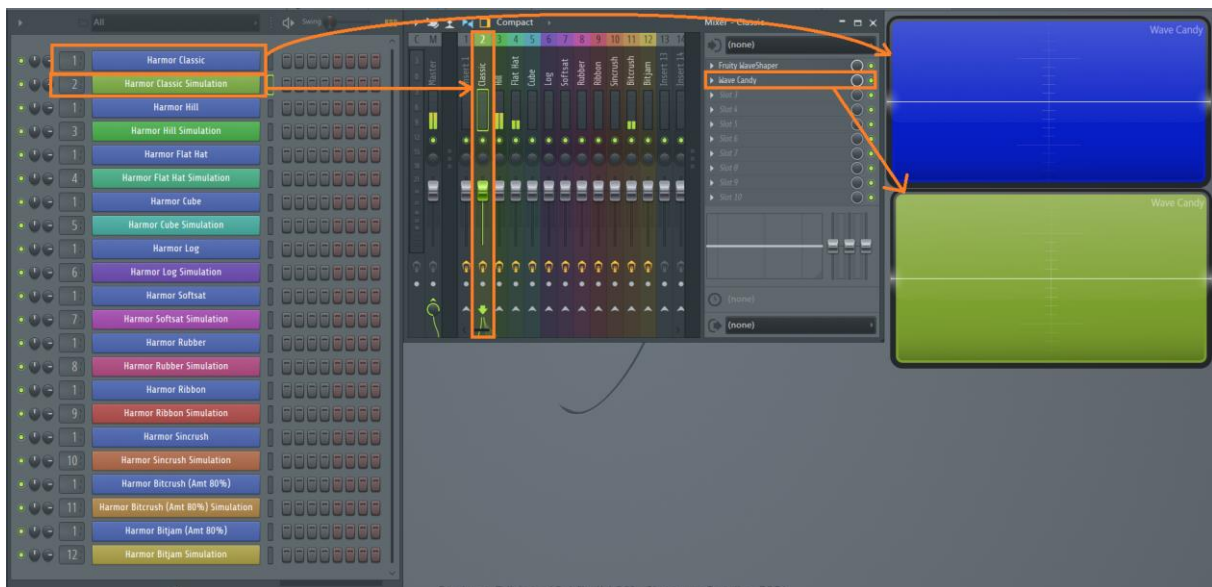
signal is mixed with the original signal. At 50% only the distorted signal routed to output. At minus 50% only the distorted signal is sent but it is inverted.

### Distortion high cut

This is an uncommon name for a low pass *filter*. This is by default turned on a little bit.

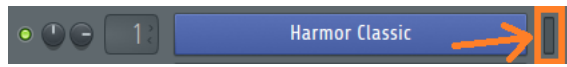
### Distortion illustration project

Please open the project <https://www.malmgren.nl/Harmor/Harmor-40.flp> to try out different Harmor distortion types.



In this project I inserted 22 Harmor patches in the channel rack as you can see in the image here above, two for each distortion type, a native and a simulated.

There are 11 Harmor patches colored blue and they are all connected to mixer track 1 with a WaveCandy oscilloscope also made blue. The idea is that you can select for example Harmor Classic and play a C5 note. If you click on the Harmor Classic patch button itself FLStudio



will select that channel **and** display the Harmor interface. If you click on the small rectangle to


the right of the blue button, then that channel is selected but the Harmor user interface is not displayed. This is nice trick to keep things uncluttered.

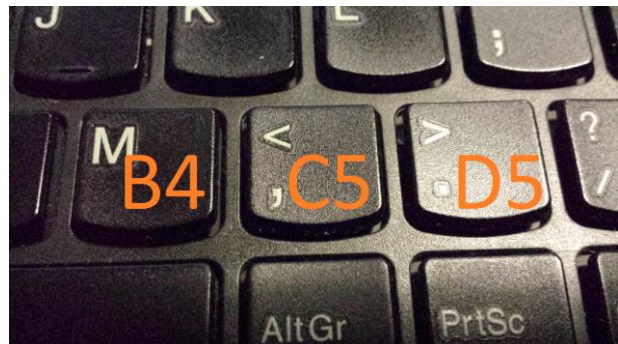
When playing a C5 with Harmor Classic selected the blue oscilloscope will display a triangle waveform distorted with the Harmor Classic distortion type. If we press the freeze button in WaveCandy then it is possible to keep the picture of the



waveform without playing the note. To bring up the interface of WaveCandy you can either click on it in the mixer track or you can right-click the WaveCandy and select *View settings*. When the graph is frozen, we can release the C5 key and hide the WaveCandy settings panel and move on to another Harmor to compare with.

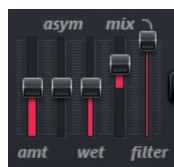
Then select for example Harmor Classic Simulation. It is connected to mixer two where there is a WaveShaper and a green WaveCandy oscilloscope. Again, play a C5 note to see how a triangle waveform is distorted with WaveShaper.

Perhaps you do not have a music keyboard connected to your computer? That is no problem! You can use the computer keyboard as well when you got this symbol  highlighted in the top section of FLStudio. In the tooltip window of the note that you are playing is displayed. You can find the C5 note on the lowest row of letter keys somewhere at the right side of the



keyboard. On my keyboard C5 is under the "<,"-key.

The oscilloscope would not zoom in enough to display all the intrinsic aspects of the waveform. To solve this I set the base note to C7 so that the result is a C3. This very low note displayed in the oscilloscope displays a much more zoomed in picture of the waveform.



For all the experiments below, unless explicitly mentioned, for the native distortion I set Harmor to the setting as displayed in the image to the right.

This is so that I get the distortion without a high cut filter and without mixing in the original signal.

## Harmor Classic Distortion



## Harmor Hill Distortion



## Harmor Flat Hat Distortion



## Harmor Cube Distortion



## Harmor Log Distortion



## Harmor Softsat Distortion

This filter is doing almost nothing.



## Harmor Rubber Distortion



Finally some serious action!

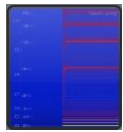
## Harmor Ribbon Distortion



## Harmor Sincrush Distortion



## Harmor Bitcrush Distortion



In this distortion type there is much going on but with the regular settings there is not much to see. There is an extra “tone” that we can see when we look at the signal with the spectroscopic diagram. When we increase the amount to 80% then a more visible oscilloscope pattern is revealed. I decided to try to achieve that distortion.



## Harmor Bitjam Distortion

Also here I set amount to 80% of the native distortion effect.

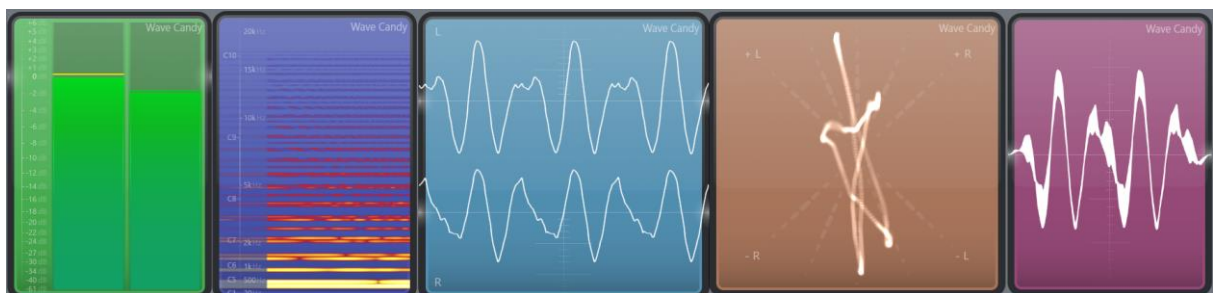
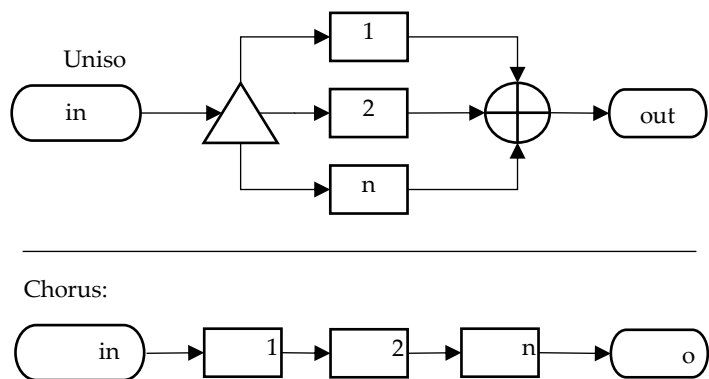




## Chapter 16. Chorus

The chorus effect is a cheaper form of Unison. To the right, you find a diagram indicating the differences between the two effects. Unison in Harmor is an additive effect operating on partials and each sound processing step is taking place concurrently. Chorus is an effect that operates on the sound signal at later stages of Harmor. Chorus manipulates the sound and sends the sound signal to be processed in the next step and so on. The number of steps is called order and each step is called a voice. I created a test project for chorus:

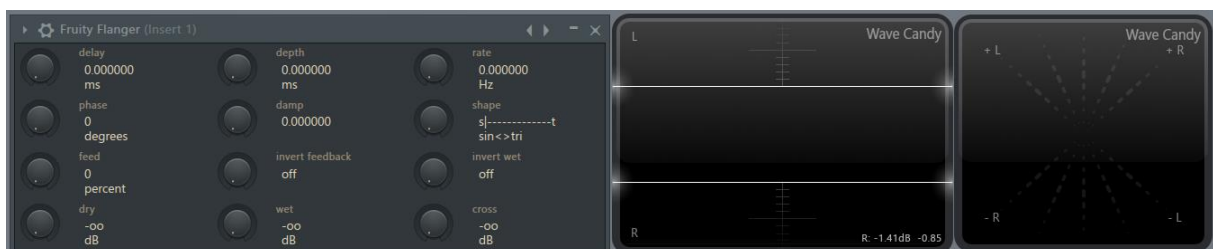
<https://www.malmgren.nl/Harmor/Harmor-41.flp>. Here I play two notes with a chorus effect with as result that the vectorscope displays a five-pointed star swirling around in 3D!



Chorus in Harmor is the same as the stand-alone Flangus effects plugin. Flangus is made by stacking an “order” number of Flanger effects.

### Fruity Flanger

To understand chorus in Harmor we can look at how Flanger and Flangus works. In the test project <https://www.malmgren.nl/Harmor/Harmor-42.flp> I set up a project with Flanger and an oscilloscope with stereo separation and a vectorscope. To try this, play a C5. Initially you will not hear any sound.



Flanger got 12 knobs! The first three to try out are dry, wet and delay. The dry knob sends the original signal out and the wet knob sends the modified signal out. Try setting these two to 100%. The delay signal introduces a delay between the dry and the wet signal. If we only got wet knob set, then there is not going to be much of an effect.



Suppose we set delay to about one millisecond. Now we set depth to around for example two milliseconds.



By doing so we introduced a delay of in total three milliseconds. Flanger can by itself sweep between the delay value and the delay value plus depth. In our case, between 1 and 3 milliseconds. Now we set the sweeping speed with the rate button. Set it to for example 0.3Hz.



When you try this in the test project you will see that the right and the left signal will sweep between the dry and the wet. Now when we turn on phase to 90 degrees the left and the right signal will move coordinated but with a 90 degrees' difference.



The knob damp provides a high cut filter to the wet signal but not to the dry signal. Here below you can see that one tip of the waveform is round and one is sharp.



The shape knob changes the wave type used for the sweep motion. Sine makes the sweep smoother. Now suppose you do not want any of the dry signal into the effect what do you do then? It is possible to feed the wet signal “back into” the effect instead of using the dry signal.



Because the wet signal already contains copies of earlier feeds when increasing the feed percentage, the result quickly holds many copies of the original sound. Next you can see how it looks like when we invert the wet signal give it reverse polarity.



Here we invert the feedback signal.

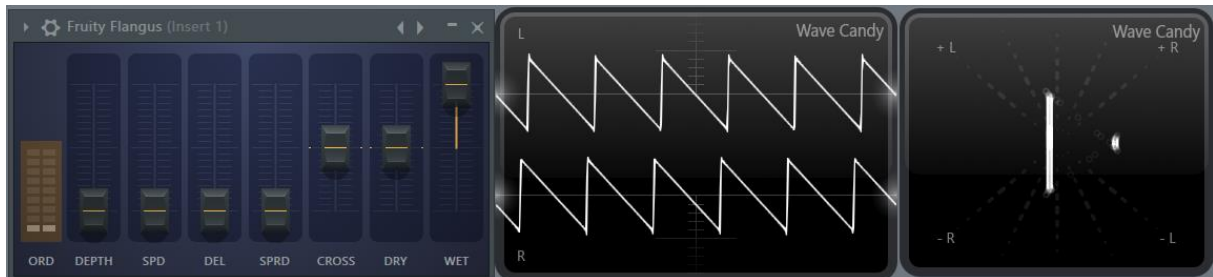


The cross knob copies a percentage of the left wet stereo signal over to the right stereo signal and vice versa.

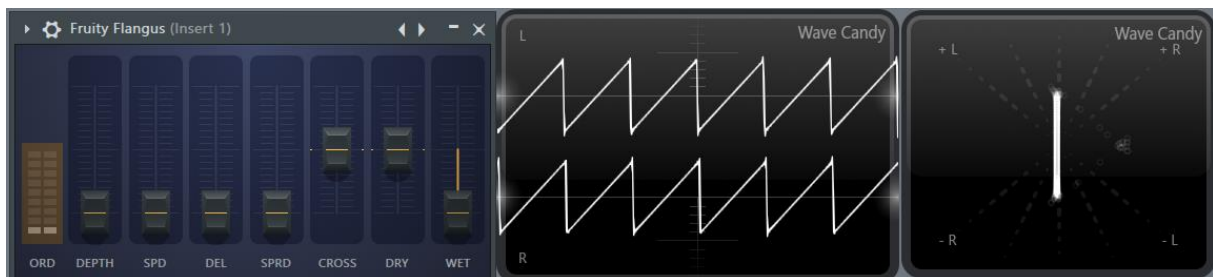


## Fruity Flangus

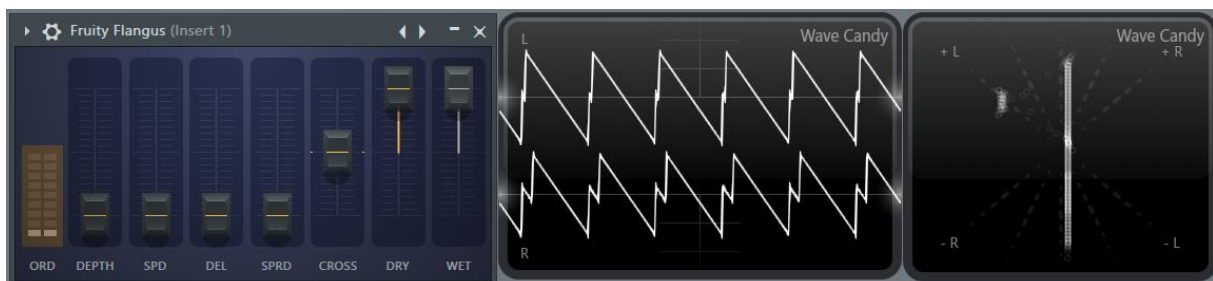
Now we can analyze Fruity Flangus in the same way. For this I set up another project <https://www.malmgren.nl/Harmor/Harmor-43.flp> with a Fruity Flangus and an oscilloscope and a vectorscope. Here is Flangus sending the signal directly out when wet is set to 100%.



Please notice that it is possible to send out -100% wet as well. There is already a tiny difference between left and right.



I restore the wet back to +100%. When we increase the dry knob we introduce a second copy of the signal is added to the output. This stays stable.



When increasing depth, we introduce a difference between the two copies.

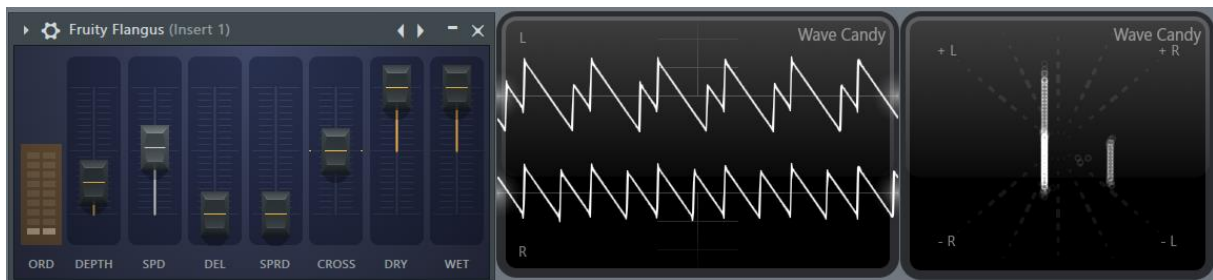


By adding a bar to the order control, we get three copies of the signal mixed into the output. You can experiment with adding more bars and see how many peaks are added to the graph.





When we increase the button speed, *spd*, then the sweeping starts to take place. In Flanger this was called *rate*.



Delay, *del*, specifies how long time it takes for the different copies to start to get added to the effect. I must say that it looks like there is only a delay between the first copy and then all the rest comes at almost the same time and this holds especially when the spread knob is set to 0%. When the spread knob is set to 100% the delay between the voices are evenly spread. It is possible to see this when recording the sound with Edison and then zooming in on the sound to see how it behaves when starting to play a note. This I could observe. The manual says that each voice is assigned “a different blah blah blah *etcetera*”. That could be true but what is really going on?

Please recall that in Flanger we had the knob phase. Suppose that here when the spread knob is at 100% all the phase knobs for each voice in Flangus are set as evenly as possible across 360 degrees.



However, what I can see is that the *sprd* knob makes the use of phase more “efficient”, especially for many voices. In the image here above, spread is at 100% and the signal wobbles and transforms but overall there is a great control of the stereo picture.





Here spread is 0%. There are interesting geometrical figures created but a more wild and untampered stereo picture is created.

The cross knob got two modes, positive and negative. The positive cross setting is doing the same as *cross* in Fruity flanger, it copies a percentage of the left wet stereo signal over to the right stereo signal and vice versa.

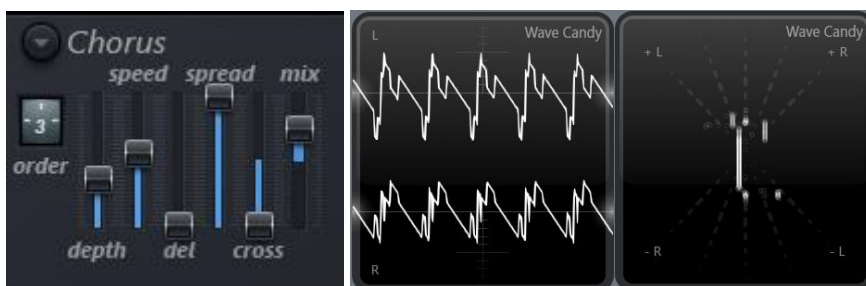


When setting cross to a negative value the copied signal is first inverted.



## Harmor Chorus

And now we can conclude that Harmor got exactly this form of Chorus engine!



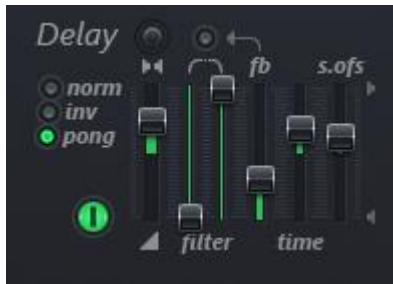
When mix is set to 0% in Harmor then we only get the dry signal. At +100% we only get the wet signal. At 50% in Harmor is like setting dry and wet both at 100% in Flangus and Flanger. At negative values you get the same but inverted. See <https://www.malmgren.nl/Harmor/Harmor-44.flp>.

## Chorus presets

The chorus presets options drop-down menu has nine chorus presets. Here below in the table all the values of the presets are listed but also the default setting, when set to order 4 becomes classic.

<b>Preset</b>	<b>Order</b>	<b>Depth</b>	<b>Speed</b>	<b>Delay</b>	<b>Spread</b>	<b>Cross</b>	<b>Mix</b>
<i>Classic</i>	4	50%	75%	6%	100%	-25%	50%
<i>Thick</i>	5	100%	85	0%	100%	53%	76%
<i>Deep chorus</i>	7	50%	50%	100%	100%	42%	50%
<i>Flanger</i>	6	55%	83%	0%	0%	30%	100%
<i>Alt flanger</i>	1	12%	30%	0%	0%	100%	-100%
<i>Stereoized</i>	2	8%	62%	0%	0%	0%	75%
<i>Modulation</i>	1	4%	93%	0%	0%	-17%	100%
<i>Surround flanger</i>	9	7%	16%	0%	0%	-51%	90%
<i>Vibration</i>	7	3%	96%	0%	0%	0%	69%
<i>Default</i>	Off	50%	75%	6%	100%	-25%	50%

## Chapter 17. Delay



Before you became a sound engineer you called it echo. From here and onwards you will call it delay. But it is echo. In actual life, you come across the echo effect in for example the Rocky Mountains or a fjord in Norway where the opposite slope is not too far away. Now suppose you are on one side of the fjord and you shout something then the sound of your voice will bounce on the walls of the opposite

side and come back a moment later and often a lot less loud. If you are lucky the sound bounces back and forth a couple of times. Some aspects of all this is created with this effect, but here it is called delay.

The basic idea is simple. The signal enters this effect module. Inside it there is a memory recording the signal and after a while it is played.

The online manual from Image-Line at the section of delay has a screenshot like I have but the text talks about other sliders??!?

### Enable delay



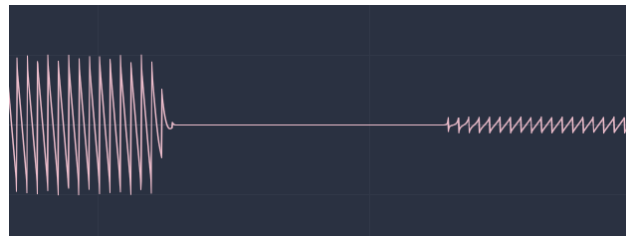
It is not a green cats eye. You just turned on the delay.

### Delay input volume and delay input panning



We do not necessarily need to input all the signal. In that case, you set this slider to a positive value from 0 to 100%.

On the other hand, you might want to enter the signal with reverse polarity and in that case, you set the slider to a value from 0 to -100%. In the image to the right you can see first the input signal is playing a regular saw waveform and the echo “replies” with a signal of reverse polarity because the input was set to a negative value.



If you need to restrict one side of the input, then you can change the panning knob above the slider.

### Delay feedback mode: Normal, Inverted and Ping Pong



There are three feedback modes. Normal sends the input to the output. Inverted sends left to right and vice versa. Pong switches between left and right.

If you would like to ping pong the left and right channel, then you set the input panning to plus or minus 100 percent and set mode to pong. When doing so the output will jump from left to right.

## Delay feedback level



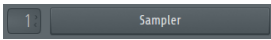


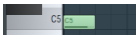
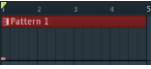
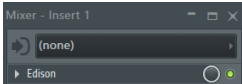
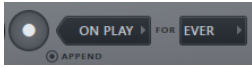
The knob delay feedback level, *fb*, specifies quick the sound in the memory will die out. At 100% it never goes away. At 0% one echo, will come out of this effects unit.

## Delay time

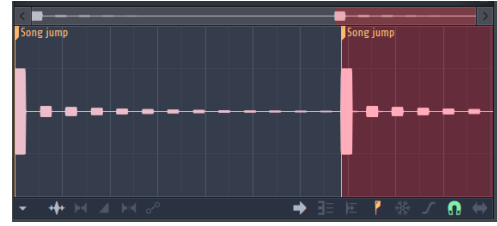


This slider specifies how long time it takes for the sound to travel to the other side of the valley and back. Above the middle the slider specifies the time in terms of project time. Below the middle the slider specifies the time in milliseconds.

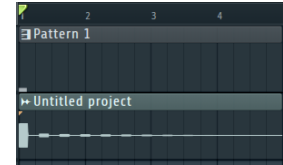
Often when we bring up time it becomes a bit confusing. Clearly it is not so difficult to understand the general idea but how about knowing exactly what time we talk about here? Here is a little experiment that you can carry out to find out what the project time is. You can record the pattern you want to investigate. Then we can put the recording into the piano roll and make conclusions about the time in there. For this experiment, I have not created a project so if you want to try this then follow the steps here below.

1. Create a new empty project.
2. Replace the default sampler  with a Harmor.
3. In Harmor activate the FX tab and turn on delay. 
4. Increase delay time to 5:00 and feedback to around 80%, leave the rest at standard settings. 
5. In the channel rack right click Harmor and select piano roll. Set one note of C5 in the pattern. You already hear the echo but how long time is there between the echos? 
6. Now in open the playlist and place the pattern. Drag out the pattern so that it is really long. 
7. For mixer slot 1 of mixer track one add Edison. 
8. Set the *when* drop-down to “On Play” and maximum recording time to “Forever”. 
9. Press the round record button inside the already open Edison (do not push record button in the Toolbar).
10. Now in the toolbar press play.

11. When starting playing Edison will start recording. When a song jump emerges in Edison you can stop playing. Switch to Edison and double click the last Song jump marker. Now you can press Del/Delete on your computer keyboard to delete the extra sound.

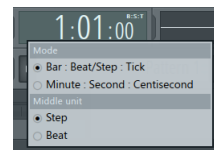


12. Drag from the “drag button”  to the play list. Release the sound there.

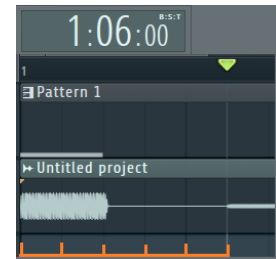


13. Now we got the recorded sound in the playlist. When it is there we can start investigating how it looks like.

14. In the toolbar right click the time window. By doing so we can see that it is set to display B:S:T and it represents Bar: Beat / Step : Tick and the middle unit is Step.

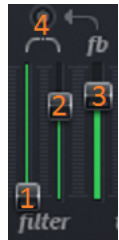


Now we can start counting! In the image to the right you can see that the green song location is at Bar 1, Beginning of step 6, tick 00. It has taken 5 steps to get here!



With this method, you can switch between different time types without getting confused!

### Low-cut, High-cut, feedback and feedback damping



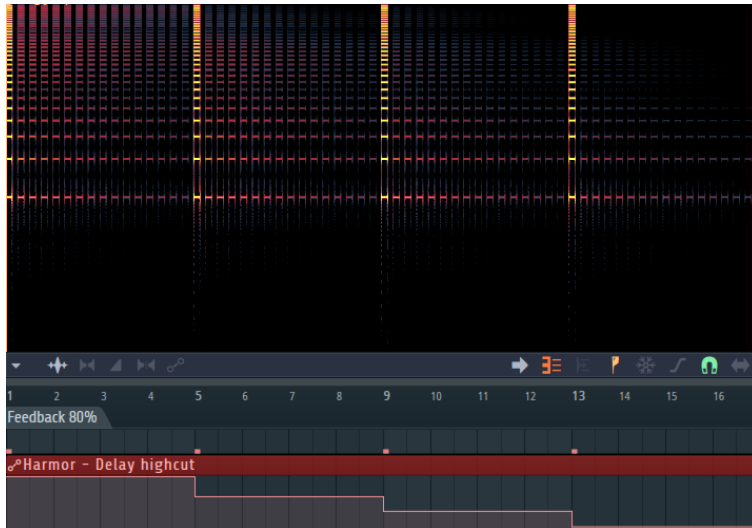
Now we will talk about the controls in the image to the left.

1. Delay low-cut. This is a high-pass filter.
2. Delay high-cut. This is a low-pass filter.
3. Delay feedback.
4. Delay feedback damping.

For the explanation of this section I created an example project:

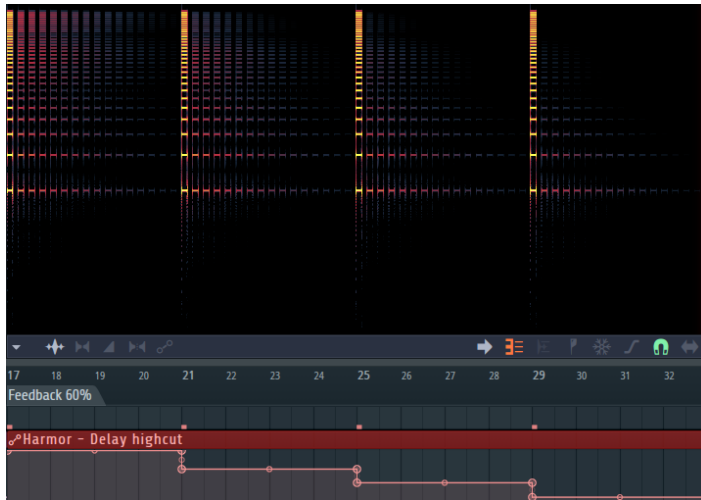
<https://www.malmgren.nl/Harmor/Harmor-45.flp> that I will use to explain what this part of the delay effect is all about.



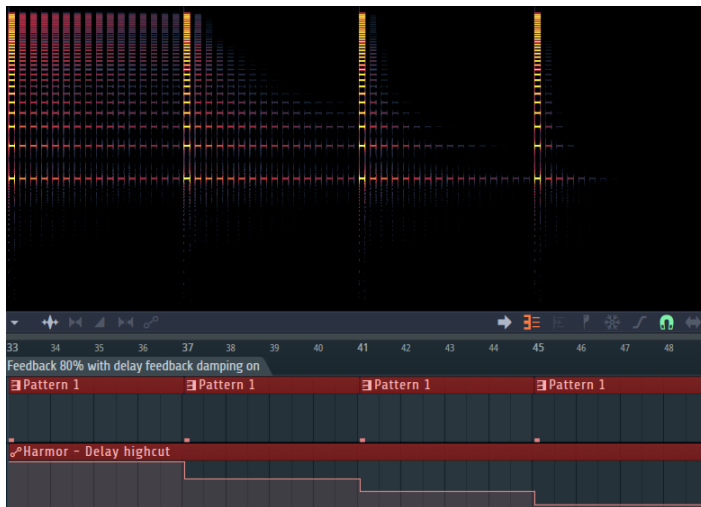


you can see that initially the note is played. Then the echo effect takes over and repeats the note and slowly making the sound quieter.

In the first section of the project at the marker "Feedback 80%" I am playing a note C6. It is a saw waveform and delay effect is turned on. Feedback is indeed set to 80%. Delay feedback damping is turned off. Now we can play a note and let it die out and observe how this looks like. In the project, I automated the high-cut slider to the following values: Off, D8, F6, G4 in that order. In the image to the left

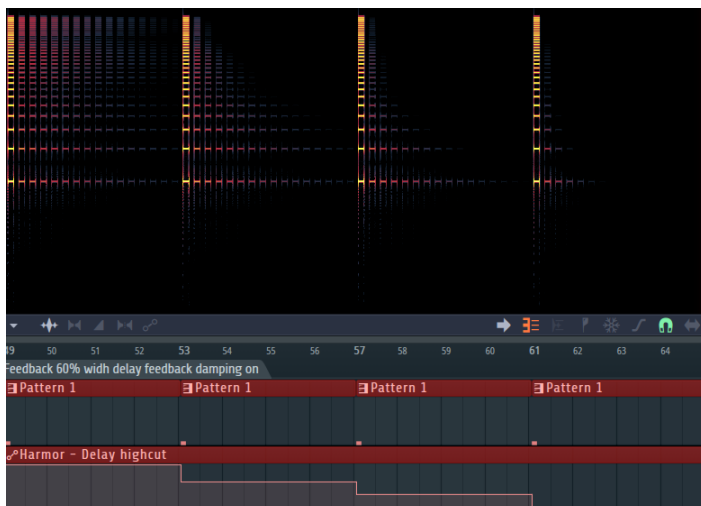


The rate at which the echo gets quieter is determined by the feedback slider. To see this, we set feedback to 60% and play this. Now the effect is dying out quicker. There is a tendency that the lower frequencies are surviving longer.



Now we are going to do the same experiment again but this time we will turn on the switch “delay feedback damping”. In the image to the left, feedback is set to 80%.

This time the signal dies out much quicker as the high cut is lowered. With feedback damping on, the output fed back into the filter each time and when doing this on an already filtered output the result quickly gets very filtered.

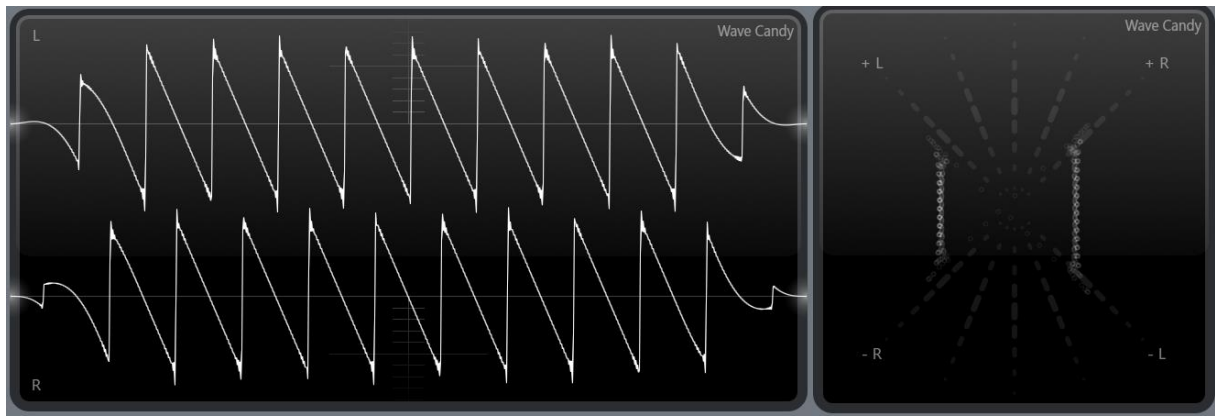


In the image to the left feedback is set to 60% and feedback damping is turned on. The signal dies out even quicker.

We got the same result for the delay high-cut filter but it is less visible so I will not bore you with images of that.

### Delay time stereo offset

This slider is marked *s ofs* and it specifies the offset between the right and left channel.



This effect is changing phase between the left and the right signal. In the project <https://www.malmgren.nl/Harmor/Harmor-46.flp> I set up an experiment where the stereo offset is automated so that it is possible to experiment with the various other sliders to see what the result it gives.

## Chapter 18. Reverb - Reverberation

If you are new to sound design and you turn on reverb for the first time ever - magic occurs. With reverb the sounds you play are placed in a context, a room. The reverb effect has something in common with echo. Perhaps you already tried echo with very short delay time and did you notice that the result was a kind of room effect? This is what reverb is about. In the previous chapter, we talked about a valley or Norwegian fjord bouncing sounds between the slopes with a substantial distance between the slopes. The reverb effect is the sound effect of bouncing sounds a much shorter distance, between walls in a room. Several factors play a role here such as the size of the room and the material on the walls etc.



I can be wrong but I got the impression that the reverb effect is created by stacking a number of echo effects. Oh, sorry I mean “delay effects” of course.

The reverb effect module of Harmor perhaps originates from Sytrus. If you are familiar with Sytrus then you will be pleased to hear that Harmor has exactly the same reverb user interface as Sytrus.

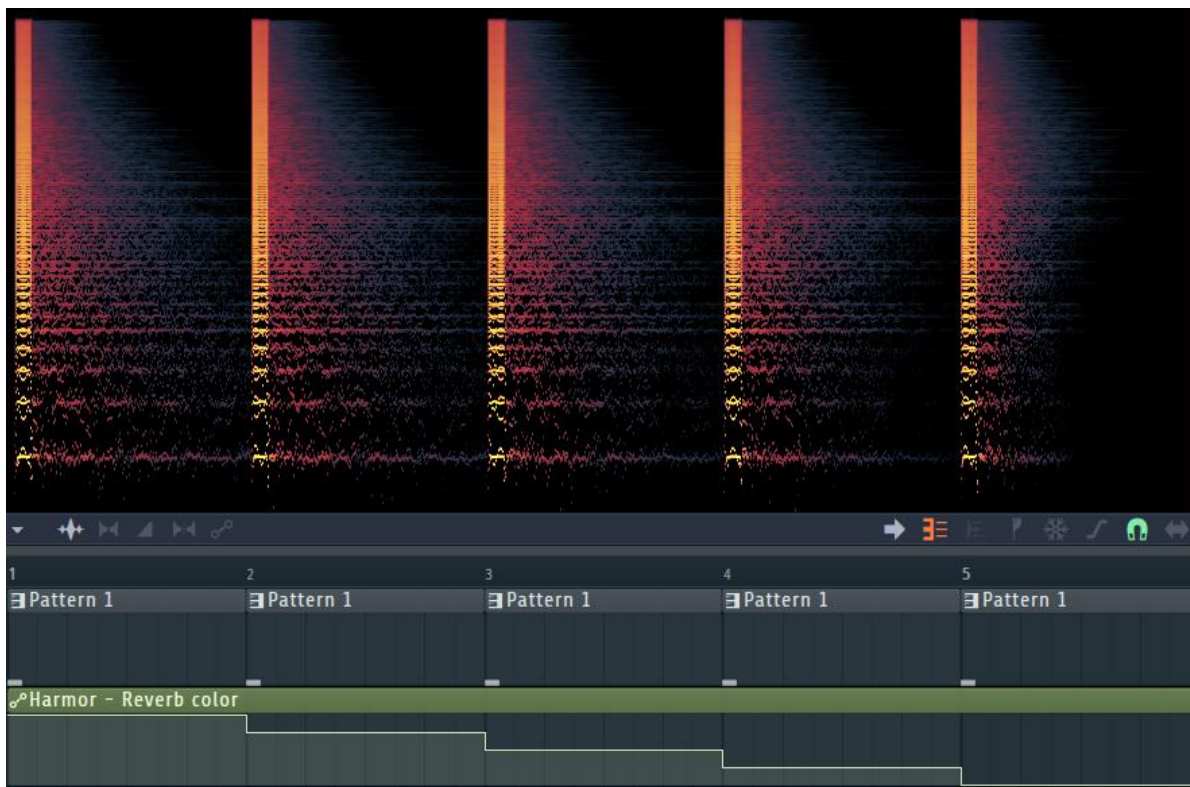
### Reverb color



The reverb color selector specifies how the bass frequencies are decaying during the reverb effect. When much bass is kept during the effect then the room is called to be “warm” and when there is a small amount of bass kept during the effect then the room is called “bright”. There are five reverb color settings:

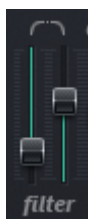
- Brighter, less bass (B+),
- Bright (B),
- Flat (F),
- Warm (W),
- Warmer, more bass (W+)

We need to experiment with this! In project <https://www.malmgren.nl/Harmor/Harmor-47.flp> I inserted a default Harmor, enabled the reverb and automated the reverb color. This is how it looks:



The first bar plays the warmest version of the reverb. The bass at the bottom of the frequency spectrum is resilient. For each step towards a brighter room the bass is more short-lived. At the brightest setting, it looks like the bass is decaying the fastest.

### Reverb input filter: Hight cut and Low cut



It is possible to filter the signal before it enters the reverb effect. For this there are two sliders above the tag *filter*. The left slider is the reverb low-cut filter and the right slider is the reverb high cut filter. Why things are called how they are called is perhaps not so important if we know what it is. A low-cut filter is a normally called high pass filter and high cut is normally called low pass. For a beginner confusions like these can be enough to give up on sound design but let's hang on in here because we are getting to the most exciting parts of Harmor!

If it is possible to filter the bass frequencies before they enter the effect, then why do we need the reverb coloring? This is because it is two different things. The filters work on the input signal but they are not "active" during the decaying of the sound whereas the coloring is. Now suppose you decrease the bass frequencies considerably then the result will sound brighter so could they not be "active" anyway during the decay? No because the reverb is working on the input it is given after filtering and if there is no bass at all to work with then obviously the result will be bright.

### Reverb pre-delay

It is possible to delay the start of the reverb with reverb delay, *del*.



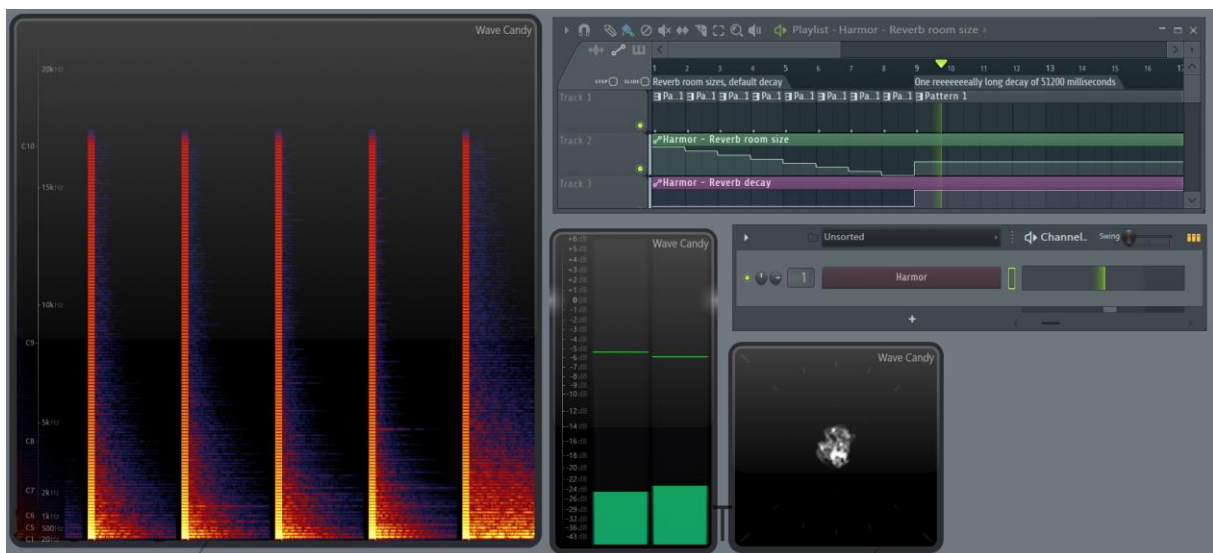
## Reverb room size

Reverb room size, *size*, sets the size of the room. Room size in this respect is obviously an abstract term because there is no room in here, this is an algorithm. That said of course it sounds like bigger and smaller rooms. But how? When analyzing the signal, it looks like primarily the room size indicates how long time it takes for the higher frequencies decaying. In bigger rooms the higher frequencies survives longer. The lower frequencies are dying out at equal pace regardless of room size.

## Reverb decay

Reverb decay, *dec*, also influence the decay time of the reverb effect however this slider sets both the high and the low frequencies.

We need to experiment with both room size and reverb decay. For this I created project <https://www.malmgren.nl/Harmor/Harmor-48.flp> where the first part has the default decay time while the room size gets smaller and smaller. The second part experiments with the reverb decay while the room size is constant. The first note in the second part has a reverb decay of the maximum value of 51200 milliseconds! The subsequent notes have lower decay values.



## Reverb high damping

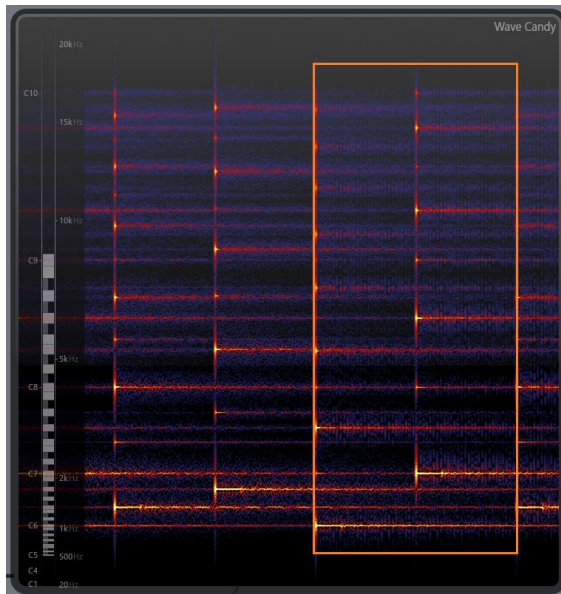
In our previous chapter about the echo effect, oh sorry “Delay”, we talked about the damping functionality. There is a switch to turn on damping and when doing so the signal being feed back into each echo cycle are also first being filtered with the filter settings. Reverb got a similar system but it only has high frequency cut off.

The everyday life explanation to this slider is that rooms with many hard surfaces bounce higher frequencies much easier while rooms with more soft surfaces absorbs more of the high frequencies instead of bouncing them.

## Reverb diffusion

The reverb diffusion, *diff*, controls how distinct the reflection of the sound is when it bounced off a wall. A low value makes the bounce distinct and high values introduce more noise into

the bounce. Low values can be thought of as if the room is empty while high values could be of rooms with more irregular surfaces.



Here is an example showing how this works <https://www.malmgren.nl/Harmor/Harmor-49.flp> where a simple melody is played. For each bar the reverb diffusion is lowered one quarter. In the fourth bar the sounds are noticeably bouncing of the wall due to the low value of the reverb diffusion. In the image, to the left, I marked the fourth bar. If you look closely you see that in the fourth bar the sounds are “wobbling” slightly.

### Reverb wet volume

The reverb wet volume slider, *wet*, goes from -100% to +100%. At 0% the output is identical to the input. The default value is 50% when 50% of the original signal is mixed with the output of the effect. At +100% we can only hear the effect.

50% is the normal setting and it sounds as if the listener is standing next to the synthesizer. Perhaps the listener is playing the synth? At +100% you could say that the listener is standing in the room a bit further away from the synthesizer?

What is the difference between the positive and negative values?



The left image here above is a saw waveform treated by reverb at +50% wet signal and at the right the same saw waveform at -50%. It could be that the shapes are perhaps reversed but it is not so decisive.

When we look at the same in the vector scope view we have a similar situation. It is as if what was left moves to the right side but it is also getting the a slightly different shape.

I suggest that you play around with various settings of reverb and try out how they sound like and how the sound looks like in the vector scope, great fun!

To try this, you can load this project: <https://www.malmgren.nl/Harmor/Harmor-50.flp>.

### Reverb options

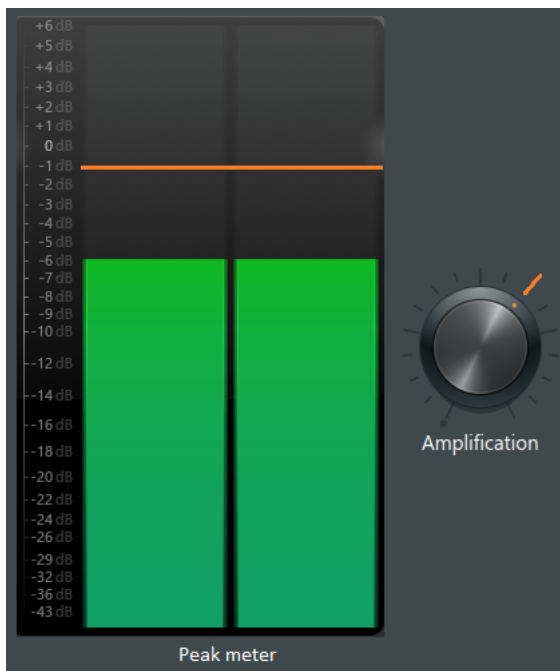
The reverb options drop-down menu has six presets for the reverb. It is really informative to see how the different options are created. Here are the values of the presets:

<i>Preset</i>	<b>Color</b>	<b>low cut</b>	<b>high cut</b>	<b>Pre-delay</b>	<b>Room size</b>	<b>Diffusion</b>	<b>Decay</b>	<b>Damp</b>	<b>Wet volume</b>
<i>Classic</i>	F	105 Hz	6300Hz	0:00	18%	100%	503ms	3000Hz	55%
<i>Large hall</i>	F	Off	11516Hz	48ms	40%	100%	2232ms	5082Hz	60%
<i>Cathedral</i>	W	35Hz	2181Hz	74ms	90%	100%	4589ms	3107Hz	60%
<i>The venue</i>	B	391Hz	6086Hz	83ms	60%	65%	2359ms	3746Hz	53%
<i>Drum room</i>	F	100Hz	3644Hz	0:00	35%	100%	602ms	3569Hz	55%
<i>Small studio</i>	F	100Hz	6086Hz	0:00	17%	100%	492ms	3043Hz	55%
<i>Default</i>	F	75Hz	4043Hz	0:00	40%	100%	1514ms	4071Hz	50%

In the last row, I added the default values.

## Chapter 19. Compression

The basic idea of compression is simple. Suppose we are presented with this hypothetical device:



This instrument has one peak meter display and one amplification knob. Our job description as compressor would be “When the green bars of the peak meter comes above the orange line then quickly you need to lower the amplification. When the bars are back below the orange line then please bring the amplification back to the original position.”. Simple as that. It would be a tedious job though!



The next step of compression functionality is to split the frequencies of the signal in for example three bands: low, mid, and high. Then each band have its own compressor so to speak. At this point it starts to get really interesting. With a three-band compressor there are more to be done to the signal than just making sure it is not peaking to high above a certain level of decibel. It is for example possible to accentuate lower frequencies, at various degrees, in favor of higher frequencies. This is also what the compressor of Harmor is specialized into doing.

### Compression type

The compression type menu has five options: Off, Limiting, Warming, Heating and Burning. It is my assumption that these names are analog to the room temperature in the reverb effect such as that when there is more bass the room is “warmer”. Here in this case the compressor is giving the lower frequency bands more amplification the warmer the compression type is getting.

### Compression amount

When the compression type is set to limiting then the compression amount is disabled. For the other types this slider sets how strong the effect is supposed to be.

### Compression band: Low, mid, and high

Compression amount has an overall influence on how much compression effect will be applied to the signal but if that is not sufficient then it is also possible to adjust the “balance” between the different bands with help of the compression band sliders.



## Chapter 20. Line editor

The manual of Harmor call this part “envelope or mapping shapes”. That is perhaps the correct term for the function that these lines have but here we are going to talk about the lines themselves and the editing of them. In subsequent chapters, we will talk about the various function that the lines can represent.





You already used the line editor several times! The line editor is everywhere in FLStudio: Parametric EQ, Timbre windows, Sytrus, Automation clips, WaveShaper etc. It is a shame that I waited with explaining this editor until after you figured it out yourself.



Image 49. In this chapter, we only look at the editing tools within the orange line in this

### Point modes

In the lower right corner, you find the point mode icons:






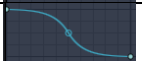



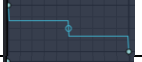




- Step editing . Left drag from the background to create points of the last selected type. Right click and drag across the graph to delete multiple points. This is the most convenient way of deleting many points. I suggest that you remember this tip!
- Freeze editing . Use freeze to lock the graph from accidental editing.
- Snap to grid . Points are snapping to the grid.
- Slide succeeding points . Use this to preserve the graph on the left side of the dragged point.

### Basic line operations

Here are the basic line operations with step editing turned off :

- To create a line segment right click the background. To insert a new control point of an existing line right click the line.
- To move a control point left click and drag the point. Hold shift while dragging to lock the vertical position and hold alt to lock the horizontal position.
- Right click a control point and select delete to delete it. You can also hold alt and left click a control point to delete it.
- Right click a control point and select copy the vertical position of the control point. Right click another control point and select paste so that the second point gets the same position as the first point.
- Right click a control point and select a segment type.
- Somewhere in the middle of the line you find the tension handle. Left click and drag to change the tension. What this is doing exactly depends on the segment type.
- Right click the tension handle to reset it.
- Control left click and drag the tension handle to fine tune it.

## Line segment types

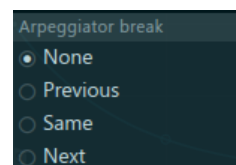
Hold	
Smooth	
Single curve	
Single curve2	
Single curve3	
Double curve	
Double curve2	
Double curve3	
Half sine	
Stairs	
Smooth stairs	
Pulse	
Wave (triangular setting)	
Wave (sine setting)	

The wave line segment type can have two different shapes depending on the setting of the tension point.

## Arpeggiator points

When a graph is time-based, it represents actions of some sort over time, then it is often possible to assign a point arpeggiator actions.

Arpeggiation comes into action when playing several midi notes at the same time. With normal behavior Harmor will play all the midi notes from the piano roll at the same time if you for example programmed a chord. With arpeggiator breaks we change that behavior so that Harmor starts jumping between the notes of the chord. With the arpeggiator break flags we can decide how Harmor will jump between the chords.

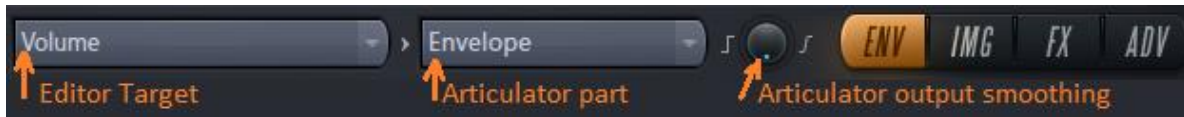


- None, no jumping
- Previous, jump to the note below.
- Same, plays the same note.
- Next, jumps to the note above.

## Chapter 21. Target basics

Until now we talked about the default behavior of various parts of Harmor. We changed the settings of knobs manually or with help of automation clips. Now we will learn more methods to modify the working of Harmor.

In the ENV tab you find two drop-down menus.

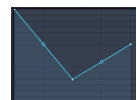


If you click the Editor Target drop-down menu it will display three groups of targets from left to right:

- Articulation. These targets have the most features: articulator part with 11 options and articulator output smoothing. The articulator part window has LFO and Envelope window with ADSR controls etc.
- Mapping. For these targets the articulator part drop-down menu has 9 options. No LFO or Envelope. The articulator has output smoothing.
- Shaping. These targets have no articulator part drop-down menu at all and no output smoothing.

The first time I opened the Editor Target drop-down menu I first thought that the tooltip was broken because it only displayed three different values! When you are hovering over the menu of the editor target, over articulation and mapping items, you will notice three different texts in the tooltip.

1. Combining mode: Bipolar sum. This is a graph with 0% in the middle. Above the mid axis positive values. Below the mid axis negative values. The shading is centered along the middle axis. On the next page, I indicate this combining mode with “BΣ”.
2. Combining mode: Unipolar product. This is a graph with 0% at the bottom and positive values above. The shading starts at the bottom and up to the line. This mode is indicated with “U×” on the next page.
3. Combining mode: Bipolar product. The shading centered to the middle looks the same as for the bipolar sum. This mode is indicated with “B×” on the next page.



## Editor Target drop-down menu

Here are the items of the Editor Target drop-down menu. All items in this table are links to the section explaining the function.

Articulation		Mapping		Shaping
Panning	BΣ	Filter 1 envelope amount	U×	Local EQ
Volume	U×	Filter 2 envelope amount	U×	Harmonic phase
Effects dry / wet mix	B×	Global envelope attack time	BΣ	Timbre 1 harmonic level
Timbre 1 & 2 mix	U×	Global envelope attack scale	U×	Timbre 2 harmonic level
Filter 1 frequency	BΣ	Global envelope decay scale	U×	Harmonic randomness
Filter 2 frequency	BΣ	Global envelope sustain offset	BΣ	Harmonic protection
Filter 1 width	U×	Global envelope release scale	U×	Harmonic clipping
Filter 2 width	U×	Global LFO amount	B×	Harmonic prism
Filter 1 resonance amount	U×	Global LFO speed	B×	Harmonic unison pitch
Filter 2 resonance amount	U×	Global LFO phase	BΣ	Image formant mutation
Filter 1 resonance width	U×	Unison phase	B×	Filter shape 1
Filter 2 resonance width	U×	Predelay	B×	Filter shape 2
Filter 1 resonance offset	BΣ			Resonance shape 1
Filter 2 resonance offset	BΣ			Resonance shape 2
Filter 1 & 2 mix	B×			Pluck shape
Pluck amount	BΣ			Phaser shape
Phaser mix	U×			Harmonizer distribution
Phaser width	BΣ			Global EQ
Phaser offset	BΣ			Filter 1 mask
Harmonizer mix	U×			Filter 2 mask
Harmonizer width	U×			Filter 1 resonance mask
Harmonic clipping threshold	U×			Filter 2 resonance mask
Harmonic blur amount	U×			Phaser mask
Prism amount	B×			Harmonizer local mask
Pitch	BΣ			Blur local mask
Pitch vibrato depth	U×			Image freq local mask
Unison pitch thickness	U×			Image selected column
Image time offset	BΣ			
Image fine speed	B×			
Image formant shift	BΣ			
Image frequency pixel scale	B×			

Here are the items of the articulator part drop-down menu. Each of these parts has their own section later in this chapter.

- Envelope (available for articulation and mapping items only). This part describes the attack, delay, sustain and release settings for the sound, ADSR.
- LFO (available for articulation and mapping items only). This part is used to manipulate the sound with a low frequency oscillator, LFO.
- Keyboard mapping. This part makes it possible to connect values to the note being played.
- Velocity mapping. This part is used to connect functionality to the velocity mapping.
- Modulation X, Y and Z mapping. The X, Y and Z mappings connects the MOD control to functions of Harmor.
- Random mapping. Set knobs to a random value.
- Unison index mapping.



- Held index mapping
- Mod X speed mapping

### Combining mode: Unipolar product

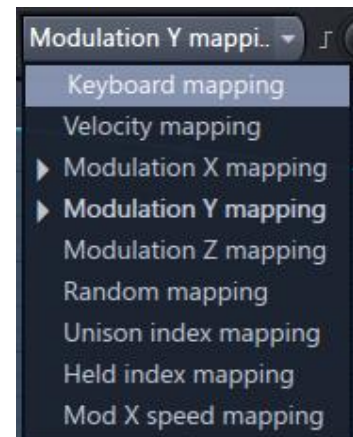
The position of a knob provides the maximum value of the articulator parts graphs with the unipolar product type of knob. The articulator parts graphs are two dimensional graphs such as that the x axis is given by the specific function of the graph while the sound designer draws the y axis values. The final value is the minimum of all articulator y axis positions. Now this became really abstract so let us look at this mechanism with an example.



To explain the combination mode in a general way I will use a fictive unit: “smoot”. The smoot unit has nothing to do with sound design so that works very well for explaining the math of the unipolar product. Suppose Harmor had a knob that we could set from 0 smoot up to 364.4 Smoot. That would be a unipolar product knob.

The smoot knob could be turned manually in the user interface, either that or it could be programmed with an automation clip and that would set the maximum value of the articulator parts graphs. If Harmor had a knob like this the tooltip would display smoot units ranging from 0 to 364.4 but if we programmed it with an automation clip, then the points in the clip would be mapped from 0 to 100% of the smoot range 0 to 364.4 smoot.

Now suppose our smoot knob was of the kind “mapping”. Then the articulation menu would contain the parts as in the menu here to the right. When an item in this menu has a little triangle next to it then that item is “engaged”, it is active. Here below I mocked up how this would look like for Modulation X and Modulation Y mapping:



When we work with the articulator graphs we could easily forget the setting of the smoot knob but that knob is always giving the maximum result value. Suppose the Oliver Smoot Position knob is set to 50% then it is corresponding maximum value is 182.2 smoot.



When we put the graphs on top of each other half transparent it is easy to see that at all points in this example the minimum values of all x positions is a straight line!



### Combining mode: Bipolar product

Suppose we got another Smoot button in Harmor. It is bipolar so that it has 0 in the middle. Then +182.2 smoot on the positive side and -182.2 smoot on the negative side. Everything we said already about the unipolar product still holds also here for the positive side of our bipolar smoot. The button position set the maximum value of the graph. The middle point of the graph is also the middle point of the knob.

What happens on the negative side? The minimum value is given by the knob and also here the middle of the knob is the middle of the graph.

### Combining mode: Bipolar sum

This mode takes the absolute value of each articulator part and the result is the sum of the resulting y values.

### Articulator part: Envelope

When you start Harmor for the first time then this is the first envelope you will see. It is the volume envelope. This is a standard control of FLStudio so most probably you have seen this in other synthesizers as well.



ADSR section	
✓ Decay	
✓ Loop start	
✓ Sustain / loop end	

The Envelope articulator part can program the sounds attack, decay, sustain and release or ADSR for short. By turning on the switch in the bottom left corner you activate envelope. The default graph represents the default tension of the various ADSR sections. You can

adjust the tension as you like or use the default. The ADSR knobs below the graph modifies the default graph. You can add extra points as well and as long the original points are kept the ADSR knobs keeps working. If you for example remove the point with the S flag then the default knobs don't know what to do anymore until you insert a new point and right click and selects sustain / loop end.

The switch *tempo* sets the time to project time.

When the switch *global* is turned off all notes have their own timing. The individual note is playing their own position in the envelope pattern. When the *global* switch is turned on then every time a new note is played then all playing notes gets that same envelope as the last playing note.

It is possible to drag an audio clip and drop it into the volume envelope window. This is equivalent to open the options menu and select *Analyze audio file...*



Image 50. The sound VOC\_138InProgress dragged into the Envelope window.

Here I dragged in the sound "VOC\_138InProgress" from the Packs -> Legacy -> Vocals. When analyzing audio, it can be useful to remove some extra points if needed. This is possible with tools in the options menu. Here below these options are discussed in detail.

#### [Envelope window option menu: Make all loop](#)

This function sets the first point to a *Loop start* and the last point to a *Sustain / loop end*.

#### [Envelope window option menu: Insert points at markers](#)

This menu item is always greyed out.

#### [Envelope window option menu: Flip vertically](#)

This function turns the graph upside down.

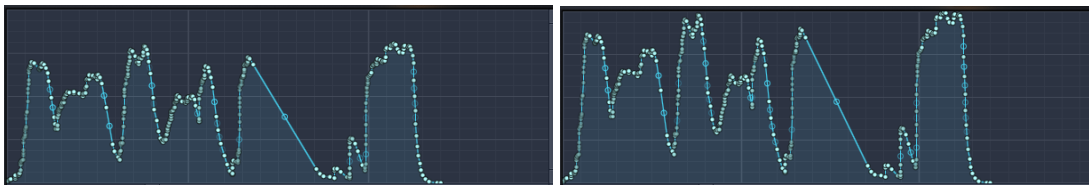
### Envelope window option menu: Scale levels...

This function opens a scale window. The slider *Offset* can move the graph up or down. When the points cannot go any further they are stuck at the top or bottom. The slider *Multiply* can stretch out the graph or compress it. Below the center of the slider it is possible to flip the graph vertically – stretched or compressed. The knob *Tension* and the slider *Center* works together and I will explain them with an example. Suppose you compress the graph first with the multiply slider. You can move the graphs upwards to the middle with Offset. Then Center moves the graph up or down relative to the Offset position when Tension is set to 0%. If you set Tension to another value and then moves the Center slider you will see how the graph is stretch and compressed at the same time.



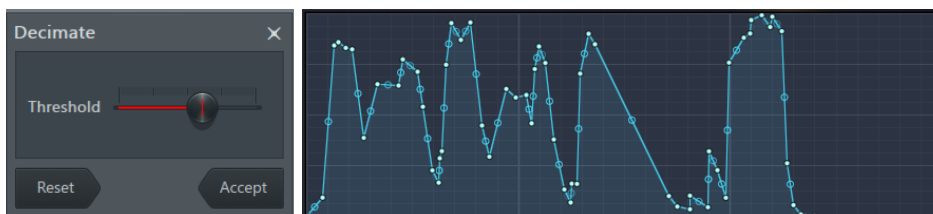
### Envelope window option menu: Normalize levels

Suppose your graph is not using the entire window vertically, then this function will stretch the graph to fill the window.



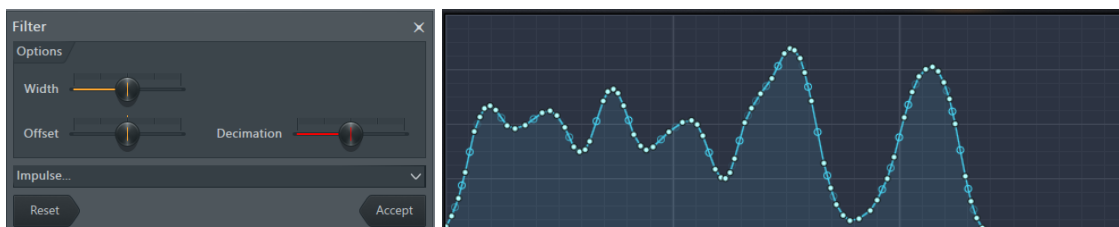
### Envelope window option menu: Decimate points

If the graph has too many points you can use this function to remove points.

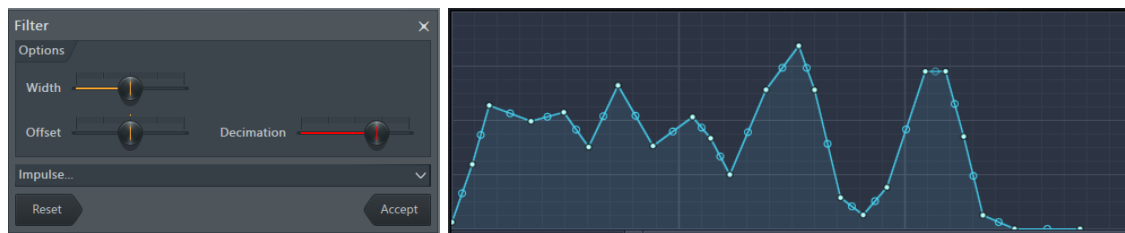


### Envelope window option menu: Filter...

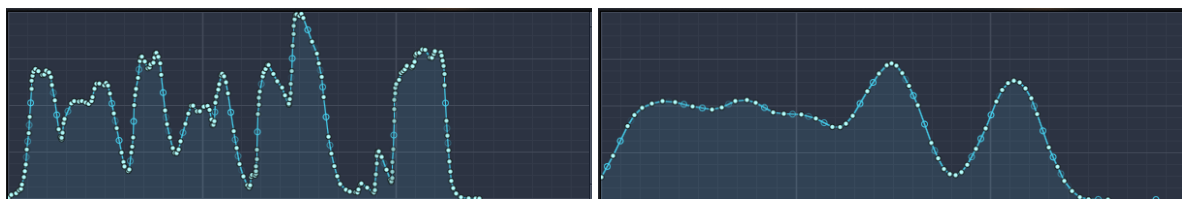
Suppose we start with the sound from Image 50 and then activates the Filter function. The default filter window is displayed.



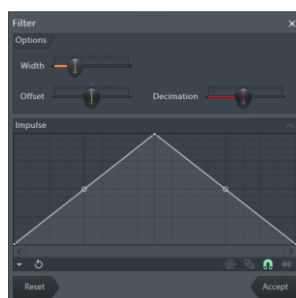
The result of the filtering is displayed directly. When dragging the slider *Decimation* to the left we keep the shape of the graph but we are introducing many more points. When dragging Decimation to the right points are removed – to that point that the graph looks different.



The slider Time Offset, *Offset*, moves the entire graph left or right. The slider Envelope Width, *Width*, sets the filtering envelope. Many envelopes right now! When dragging width to the left the original form will emerge and when dragging to the right the graph is more filtered.



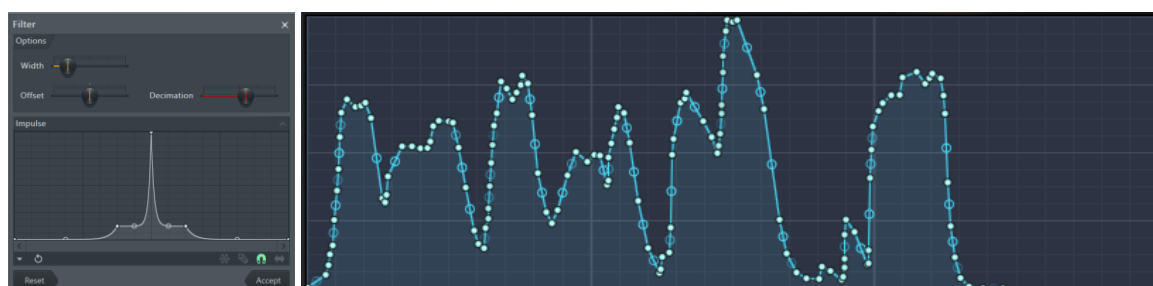
When you click, the drop-down arrow after *Impulse...* a new feature unfolds!



This is the impulse curve per which the graph is being filtered against. As I see it, this impulse shape is moved across the original graph and the points below the impulse graph are used to calculate average values. It is not so that all points in the window have equal significance in the average calculation. Instead the significance of each point is given by the distance from the bottom of the impulse graph to the line. The point in the middle has the most relevance

while points further away are less relevant. This graph is moved step by step over the original graph and during each step the average value is calculated. The *width* slider specifies the width of the impulse graph. A narrow graph will generate fewer average values while a wide impulse graph will make many more values taken into the average value calculation for each step of filtering – hence making rounder forms.

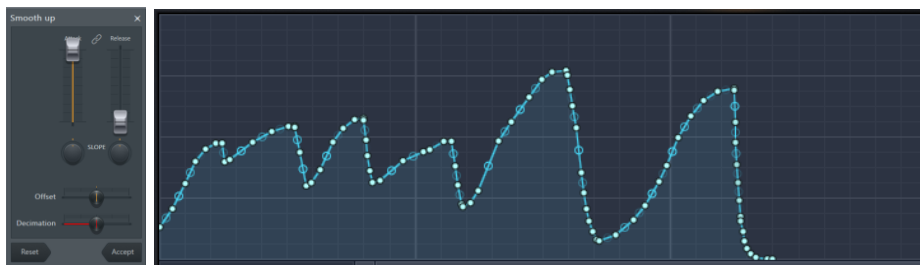
You can click on Impulse to open premade impulse graph files. You can also edit the graph yourself. Here below you see my experiment.



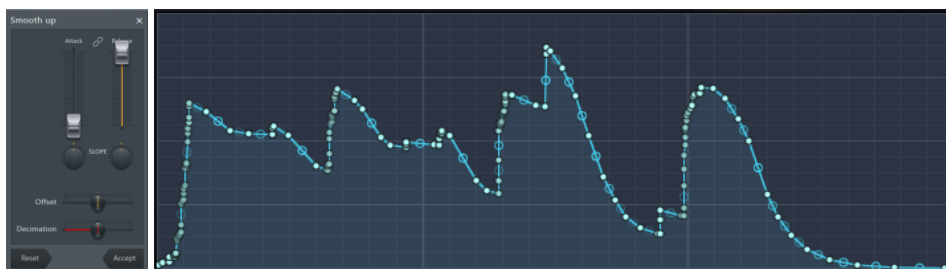
[Envelope window option menu: Smooth up...](#)

This is also a form of filter. Suppose we start with the sound from Image 50 and then activates the *Smooth up* function. This function is smoothing the graph with special attention

to shaping the attack and the release. If you set attack to maximum and release to minimum you can see how the graph now builds up slowly with quick release.



When we do the opposite, we can see that the quick attack and slower release.



Below both the *Attack* and the *Release* sliders you see *SLOPE* knobs. They amplify their effected part for positive values when the knob is to the right. The tooltip doesn't show the values but that is my guess. When the knob is to the left, their affected part is made more "quiet".

The slider *Offset* moves the graph right or left.

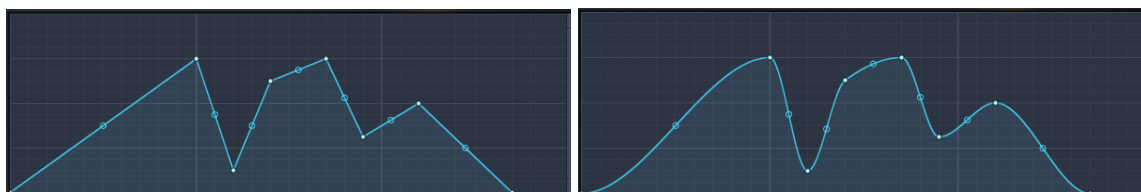
The slider *Decimation* introduces points when dragged to the left and removes points when dragged to the right.

[Envelope window option menu: Smooth up abrupt changes](#)

This function smooths up abrupt changes!

[Envelope window option menu: Turn all points smooth](#)

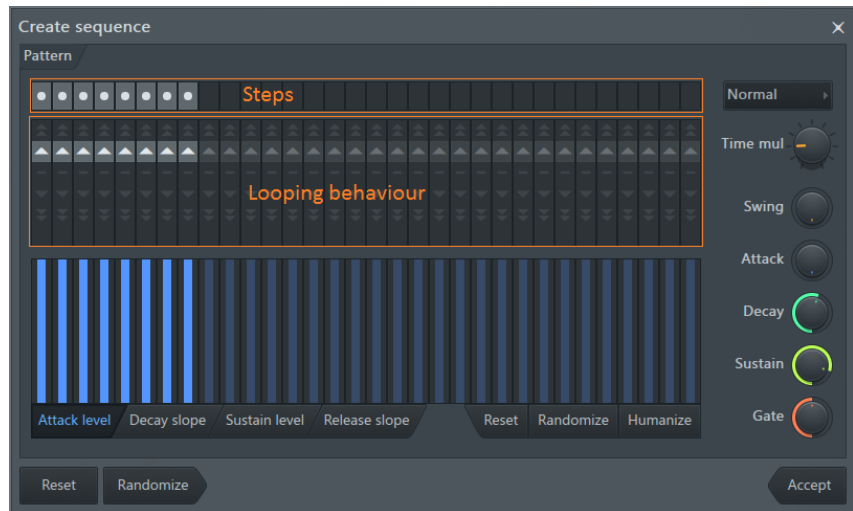
This function change the line type of all line segments to smooth types.





## Envelope window options menu: Create sequence...

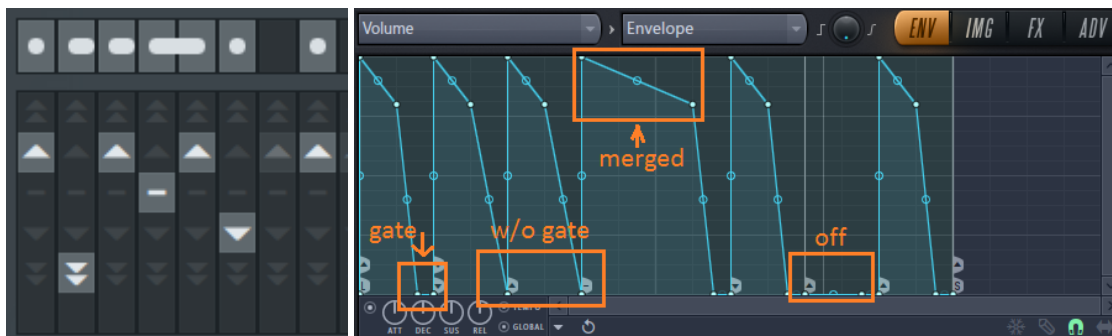
This is a tool that can automate the programming of arpeggiator breaks. To try this I suggest you set out a long-playing chord in the piano roll of a Harmor patch. Then start playing the chord and then open Harmor and then launch the create sequence function. The sequences you create with this tool is based on the grid spacing as it is defined in the window from where you opened the tool.



Here above, you can see how the sequence tool window looks like. The top row contains steps. Each step can have different states:


- On, step with gate
- Step without gate
- Step merged with next
- Off

Click on a step repeatedly to switch between the states. Right click to switch backwards. Here is an example of a step pattern and what graph it will produce.



The next area of the tool controls the looping behavior. You can set any of these:

- Play the second next higher note of the chord jump over one.
- Play the next higher note of the chord
- Play the same note as the previous
- Play the next lower note from the chord

- Play the second next lower note from the chord  jump over one.
- Off, the symbol is grayed out. All notes of the chord are played.

You can set the attack levels for each step.



You can set the decay slope of each step.



You can set the sustain level of each step.



You can set the release slope of each step.

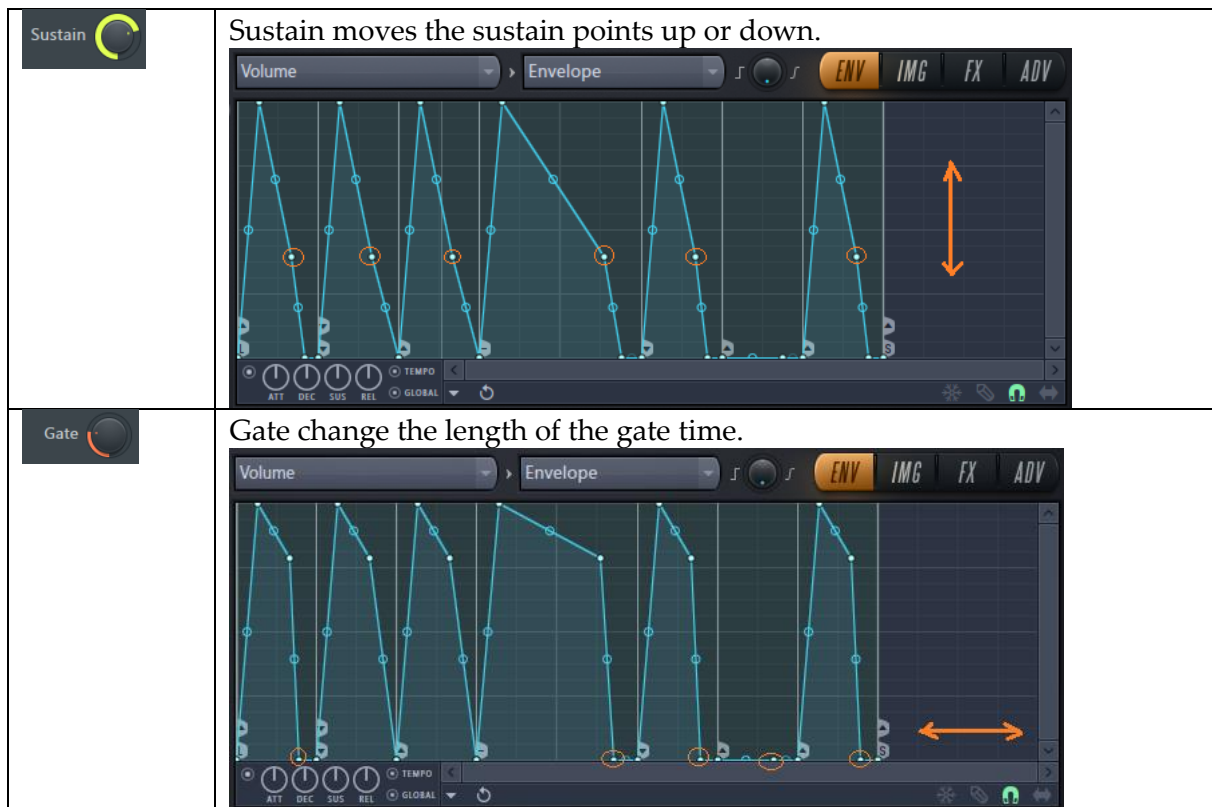


There is a reset button in case you want to reset the curves of release, sustain, decay and attack. The button randomize sets random slopes.



It is also possible to humanize a curve. This is done by introducing a small amount of random changes to the values.

<p>Normal</p> <p>Normal</p> <p>Ping-pong</p>	<p>In the upper right corner, there is a mode menu. Default value is normal. You can also select ping pong that will replicate the pattern one time and reverse the second copy of the pattern.</p>
<p>Swing</p>	<p>Swing makes every uneven step longer and even steps shorter.</p> 
<p>Time mul</p>	<p>This knob, time multiplier, changes the time base. The default time is multiplied by one and with this knob this ratio can be changed.</p>
<p>Attack</p>	<p>Attack makes the attack part of each step longer or shorter but this knob also moves the rest of each steps point towards the other step.</p> 
<p>Decay</p>	<p>Decay modifies the decay points of each step.</p> 



At the bottom of the window you have the following buttons:

- Reset. Resets the pattern to the default pattern.
- Random. Sets a random pattern.
- Accept. Accepts the current setting and closes the window.

If you don't want the pattern then you can close the window with X in the upper right corner.

### Articulator part: Low Frequency Oscillator, LFO

In these two chapters *Chapter 10. Tremolo* and *Chapter 11. Vibrato* we talked about two methods for creating Low Frequency Oscillations. LFO is the third method. By default, the LFO articulator part is turned off. If you adjust any of the knobs, then the on switch at the bottom left corner is turned on – or you turn it on by clicking on it.



The switch *tempo* and *global* has the same functionality as described in the previous section.

On the left side in the graph, you find a dot which indicates the amplitude of the LFO graph. Move the dot up and the amplitude increase and move it down and the amplitude decrease. Move the dot below the line and the starting point starts 180 degrees' different phase. If you

draw line segments from the dot, then these lines give the maximum amplitude for the graph over the time given by tempo.

There are four knobs below the graph:

- LFO speed, *SPD*; sets the frequency of the LFO curve.
- LFO tension, *TNS*; with skew at 0% then tension at 0% gives a triangular waveform. Tension at about 32% gives a sinus and at 100% a square waveform.
- LFO skew, *SKEW*; makes the waveform “lean over” so that at -100% you can create a regular “N”-shaped saw waveform and at +100% you can create a “I”-shaped waveform. I found this nifty reverse polarity *И* letter in the Cyrillic alphabet!
- LFO pulse width, *PW*; at 0% the top and bottom of the waveform have equal length. When pulse width is negative the bottom is bigger than the top and with positive values the top is bigger than the bottom.



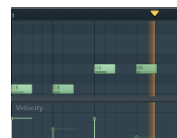
In the main part of Harmor you find the main lfo amount knob. Use this knob to change the LFO amount over all programmed LFOs at the same time.

#### Articulator part: Keyboard mapping

The keyboard mapping articulator part makes it possible to set values based on what note is being played. Why it is not called note mapping? I don't know. It is really convenient to look at the tooltip window when hovering over the diagram of the keyboard mapping because there you can read at what note the cursor is currently on.

#### Articulator part: Velocity mapping

In the piano roll every note can be given a velocity value. It is possible to program Harmor to these values as well. In the image to the right you can see notes in the piano roll with different velocity values.



#### Articulator part: Modulation X, Y and Z mapping

The X, Y and Z modulation control provides a standardized method for changing values in the synth. At first this control looks two dimensional but when you change the Z back and forth you notice that the checker pattern is the bottom of a cube approaching the viewer when the Z value is increased. In preinstalled patches that we talk more about in Chapter 26, Presets, the information window gives a brief explanation of what X, Y and Z is doing. When this information is available it is easier to browse through various Harmor patches and listen into what the patch can do and change the X, Y and Z.



The easiest way to connect a knob to X, Y and Z is to right click the knob and select “Quick map to” and then in a sub menu what modulation you need. Not all knobs have this menu option.

#### Articulator part: Random mapping

With this articulator part, similar rules apply as we discussed in the introduction of this chapter. However, the x axis of the graph seems irrelevant. It is the y range of the graph that



is relevant. The resulting value is a random value from the y range according to the combination mode rule.

#### Articulator part: Unison index mapping

We talked about the Unison effect in *Chapter 3. Unison*. Before editing this part, you will need to specify the Unison Order. The number of order voices in Unison will give the number of lines on the x axis in the graph. Each line on the x axis represents the setting of that Unison voice. For the y axis, the regular combination mode rules apply.

#### Articulator part: Held index mapping

This part gives held notes and index and makes it possible to map that index. A held note is a note that is overlapping another note but not starting exactly at the same moment. As soon as there is a gap between the notes the index starts over.

If you make chords in midi, then the held index mapping will not engage because the notes starts at the same time. If you then apply strum on the midi notes of chords, then they will start at different moments and then the index counter will be active.

Each line on the x axis represents one index step.

#### Articulator part: Mod X speed mapping

With this articulator part, we can map the speed of the mod x controls movement! Why are there no equal Y and Z versions as well? I don't know.

#### Articulator output smoothing

Next to the articulator part drop-down menu, you find the articulator output smoothing knob.



This is a filter like what we talked about with the filter options in the volume envelope filter: *Envelope window option menu: Filter....* This knob sets the width of an impulse graph, probably triangle shaped, used to calculate average values to filter the graph. A narrower impulse graph gives more of the original shape and a wider impulse graph makes the graph rounder and more filtered.

## Chapter 22. Image synthesis / resynthesis

There exist various legacy algorithms to preserve an audio signals pitch when playing it faster or slower. The sound is cut in analysis frames where each snippet is superimposed and then stitched together as good as possible. In Harmor all such legacy imperfections are removed because the additive engine has full control over the sound at any speed. As added benefit, we can do a lot of new cool things as well!

The *IMG* tab provides two ways to process sounds:

- Resynthesis. In this mode, you see the image options menu and “plane(s) to work on”-drop-down menu.




In this mode the sharp button is active. 

In this mode, the options menu has the third and fourth section grayed out.

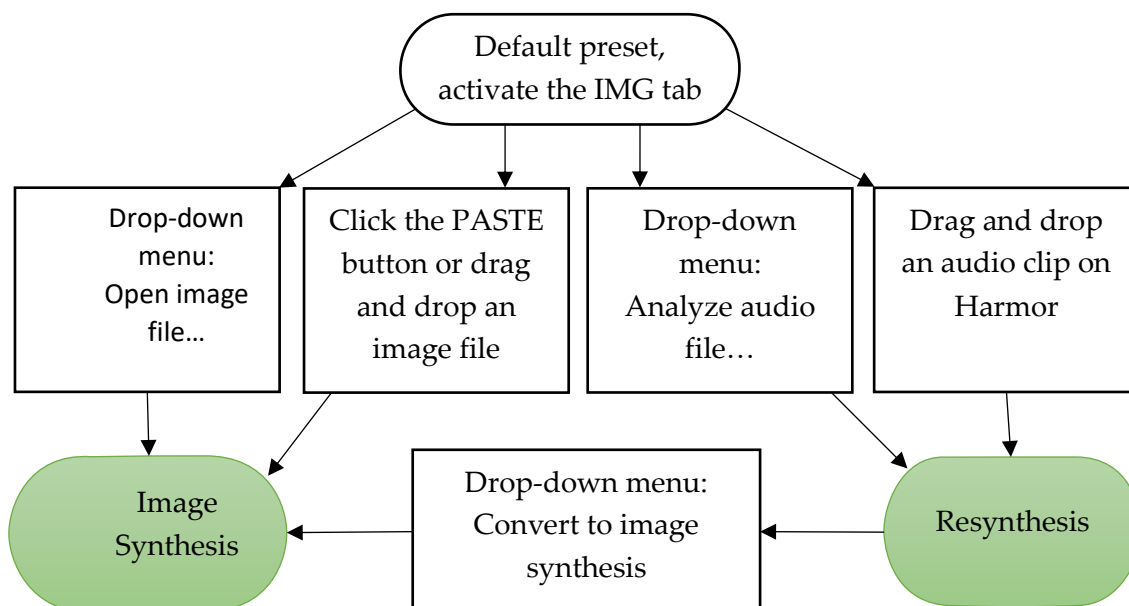
- Image synthesis: This mode is similar to resynthesis but in this mode, you see two additional buttons *copy* and *paste*.



In this mode the sharp button is inactive. 

In this mode, the options menu has sections image and selected plane enabled.

There is no switch back and forth between the two modes so that you can turn on image synthesis and then revert to resynthesis. The reason for this is because resynthesis keeps the phase data together with the “image data”. As a way of speaking you could say that resynthesis keeps an internal “higher resolution copy” of the sound. The image synthesis is a simpler form of synthesis. It is possible to convert from higher resolution to lower resolution but not the other way around. Here is how you arrive at the two modes:



When converting from resynthesis to image synthesis the harmonic phase window is injected with the harmonic phase data from the audio.

We will talk about the drop-down menu in a moment but first we will talk about other parts of the image synthesis / resynthesis.

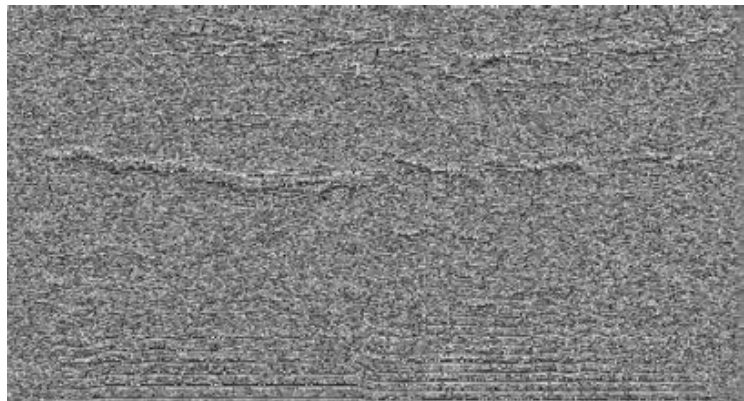
[View mode: Plane\(s\) to work on](#)

Next to the tag *PLANE*, we find a drop-down menu that sets the plane viewing mode. There are three viewing modes:

- Gain, displays the gain image. If you analyzed an audio file, then often this looks like a regular spectrum diagram. That is also how the image is configured. The lowest frequencies at the bottom and higher frequencies at the top. The x axis is the time.
- Freq, displays the frequency map. Often when you analyzed an audio file the frequency plane is filled with grey pixels. Sometimes a little darker or lighter but mostly grey pixels.
- All, combines the *gain* and the *freq* planes.

The gain plane gives what partial will be engaged, what default amplitude it should have and when. Horizontally, on the x axis, we have time. Vertically, on the y axis, a pixel defines what partial to be affected. Black pixels are silent and white pixels are playing.

The frequency plane can change the frequency of a partials from the gain pane. 50% grey is not changing any frequency. Depending on the frequency setting a darker pixel will increase or decrease the frequency of a partial. When you analyze audio often you will find that the frequency pane becomes grey



with occasional bands of lighter and darker stripes. Here above is the frequency plane of Laurie Webb Falling from Packs -> Vocals.

On the right side of the panel you find the resynthesis / image synthesis controls. They work the same in both modes – except for the *sharp* knob that only works in the resynthesis mode. Let us have a look at the different knobs here below.

[Image fine speed](#)



This knob, *SPEED*, sets how fast the image is played. The right side is 0% to 200% and it plays the sound forward. The left side is 0% to -200% and it plays the sound backwards.

### Image course speed



This knob has no tag. It is placed above the *SPEED* knob and it has a little C on the left side of it. The function of this knob is to increase or decrease the image fine speed knob. The default setting is 1.000. On the right side this knob multiplies a value to the image fine speed knob from x1.000 up to x64.000. On the left side, this knob multiplies with values starting at x1.000 down to x0.0016.

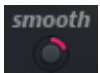
Changing speed of the playback is surprisingly robust!

### Image time offset

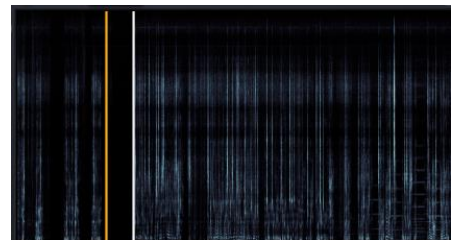


This knob, *TIME*, sets the starting position to play from in the image. You can also click in the image to set the offset. Now if we place a midi note in the piano roll of this Harmor patch then when you play the pattern you will see inside Harmor that there is a red vertical line at the starting point and a white line moves in the image indicating where Harmor is playing the image at that moment. If you set speed to a positive value, then the white line moves forward to the right. If you set the speed to a negative value it moves backwards to the left.

### Image time offset smoothing

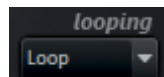


This knob, *smooth*, is one of the smallest in Harmor! It is so small it looks like a switch. This knob specifies how smoothly the changes to the image time offset should be. If we set the speed to 0 then it is possible to experiment with the offset smoothing. Put the cursor in the image and drag. You will notice that the white cursor follows the yellow closely. Now turn the time offset smoothing knob to the left and try dragging in the image. The yellow line follows the cursor and the white line is lagging.



On the right side the smoothing knob works similar to the left side except when the distance between the white and the yellow becomes “too far” then it jumps to the white.

### Looping mode



The looping mode specifies how looping takes place. With default settings, the whole window defines the beginning and end of the loop area. Later we will see how the loop area can be redefined. The drop-down menu for looping mode, *looping*, gives three options:

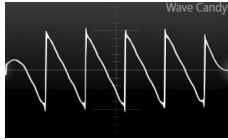
- One-shot: Plays the sound from the start position and stops playing at the end of the sound.
- Loop: Starts playing at the start position and continue play the sound from the beginning.
- Ping pong. Starts at the start position and when the end is reached reverse the direction and play the sound backwards to the beginning and then start from the start position again.

## Image sharpening



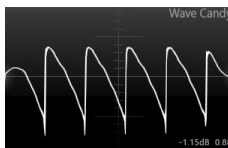
This knob is called image sharpening with the tag *sharp* but it is only active in the resynthesis mode so why is it not called resynthesis sharpening? I don't know.

On the left side, the knob specifies 0% to 100% transients only sharpening. On the right side the knob specifies 0% to 100% transients and phase sharpening.



If we record the sound of a perfect saw waveform and play it with Harmor resynthesis then the waveform is fairly well reproduced with the image sharpening knob all the way to the left – 100% transients only provided the other knobs are set to their default values.

0% is also doing a great job of reproducing the sound.



At 100% all the way to the right, sharpening transients plus phase, then the upper or lower part of the oscilloscope graph display a slightly filtered signal – it has rounded edges.

I found a peculiarity, probably a bug. The only sharpening setting not affected is all the way to the right. For all other settings when the playback reached the end and we either start over or reverse the play direction, based on the setting of the looping mode, it looks like the sharpening setting is internally pushed all the way to the right. If playback passes the starting point, then the sharpening setting is reset to where we set the knob to.

## Level section



This part of the resynthesis / image synthesis relates to the gain plane.

## Image gain pixel scale

This knob has the tag *scale* setting the amplification of the pixels in the gain plane. 50% is the default value. Above 50% makes the sounds quieter. Below 50% makes the sounds louder.

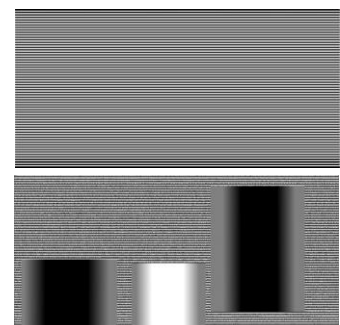
## Image gain mix

The image gain mix knob, *MIX*, sets how the frequency plane is mixed with the gain plane. Normally it is set to 100%, for what I can see this is also the best setting. When the knob value is lowered then the frequency plane is amplified. If you place a useful image there then perhaps this feature is useful.

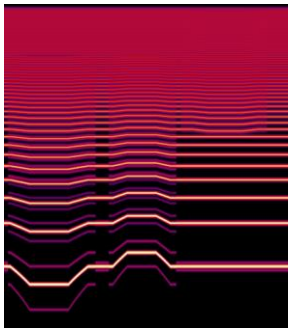
## Frequency section



The frequency plane modifies the frequency of the partials. At 50% grey no modification takes place.







The upper image to the right displays the gain map of a saw waveform. The lower image is an example of a frequency map that will help us understand what is happening here. When the frequency map gradually change the value from grey to black or grey to white we can see what is going on. If we just insert a black block or a white block, then it is more difficult to see what is happening.

#### Image frequency pixel scale

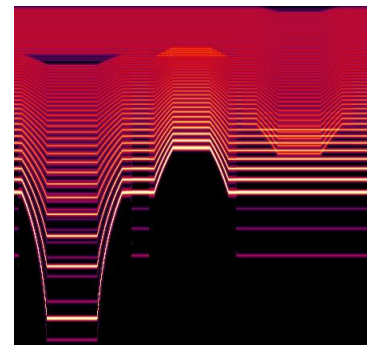
The default value of the image frequency pixel scale knob, *SCALE*, is +100%. When this knob is in the middle it is at 0% and on the right side we have positive values up to +200% and on the left side negative values up to -200%.

In the default position, at 100%, pixels darker than 50% grey, lowers the frequency of that partial. Pixels lighter than 50% grey makes the frequency higher. When the scale knob is set to a negative value then we get the opposite situation: Lighter pixels decrease the frequency and darker increase the frequency.

#### Image frequency mode

The drop-down menu for image frequency mode makes it possible to change how the frequency should be changed. The default value is Oct mode. These are the available options:

- Oct, this is the most normal mode. Lower frequencies are given more significant movements in this mode while higher frequencies are moving less.
- Hz. In this mode, all frequencies are moved the same.
- Wide Hz. The image to the right is Wide Hz mode.
- Unison. In this mode, the image controls the Unison detuning. To try this, turn on Unison. I must say it is not so easy to notice this effect however, there *is* something going on.



#### Image formant shift

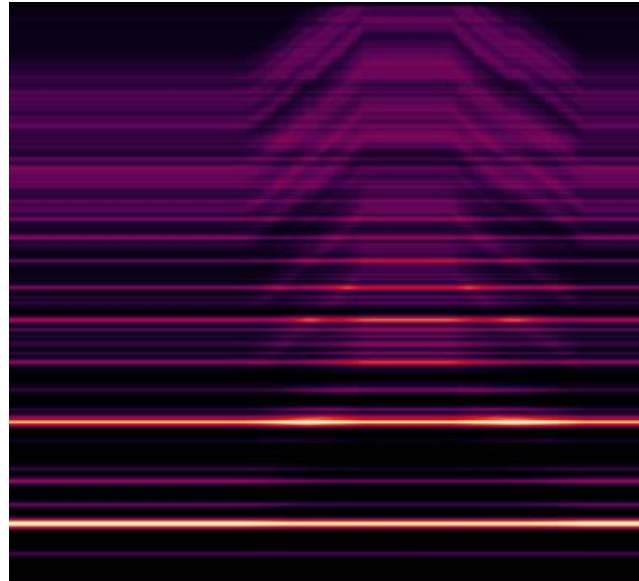
When changing the pitch of synthetic sounds the frequency moves up or down and all is fine. It is different when changing pitch of a singer's voice for example. This is because the human vocal tract has a specific shape.

That cavity resonate sounds and creates specific sound profiles regardless of what pitch a singer sings. A singer can sing at a higher pitch but the vocal tract size and configuration stays as it is. The resonating sounds created by the human tract are called formants. They are specifically noticed at vowel sounds. Formant shift preservation in Harmor is designed to preserve the formants during pitch shifts.



The knob, *FORM*, specifies the maximum range in cents that the formants preservation will be active.

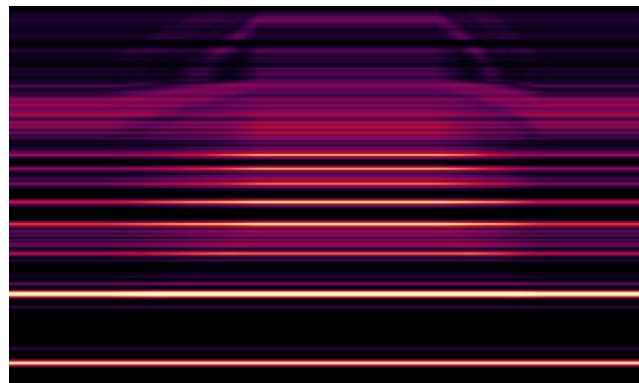
To experiment with this feature, I suggest dragging in an audio file with Formant qualities such as FLS\_FreeStyle\_05 from Packs -> Legacy. Set speed to 0%, Image frequency pixel scale to 100%. Set formant to -600 cents, and mix to +100% (we will talk about mix shortly). Now when we play the sound it is paused to one position in the image. Here below I then increased the formant knob up to +600 cents and then back again to -600 cents. You can see how the pitch of specific partials stays the same while the “cloudy” formant partials are scaled up and down. One application of this is to avoid chipmunk effect when increasing the pitch of a singer’s voice, you compensate by decreasing the formant position.



With the default setting we are moving all the formant according to the settings of the formant knob. It is possible to set the image formant mutations even more precisely by opening the editor target drop-down menu and selecting *Image formant mutation*.



The envelope makes it possible to adjust the formant changes with great precision. In the image to the right you can see the result of the envelope.



### Image formant shift mix

Next to the *FORM* knob you find the *mix* knob. This knob makes it possible to mix the formants preserved signal with the original signal and send that to the output. You can set this knob from 0% up to 100%.

### Image copy, Image paste

When we paste an image that is 516 pixels high then each pixel corresponds to a partial in Harmor. The width of the image is the length of the sound. The brightness of a pixels is interpreted as the amplitude of the partial. A black pixel is a partial that is turned off. The pixels of the image are assigned from the bottom corresponding to the lowest partial. If the image would be smaller 516 pixels then the remaining partials will not be used.

Images are calibrated to a brown noise profile. We talked about the Brownian filter in the section about timbres: *Harmor Timbres*.



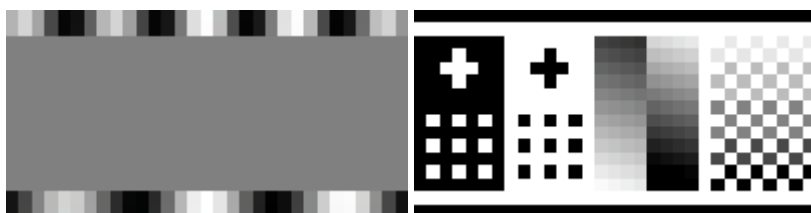
The two effective planes are gain and frequency. As I can see it the *all* mode is not a plane but a view mode. However, the user interface allows you to copy and paste to and from the *all* view mode image as well. IMHO that only gives confusions. Suppose you set view mode to all and copy the image and paste it into an image editor. You make changes to the image, for example adding green and blue areas, and now when you paste it back the part that belongs to the frequency should go to the frequency and gain goes to gain but that is not what happens... You get a mix?!

### Image frequency interpolation curve and Image gain interpolation curve

The pixels in an image can fluctuate abruptly and that can give undesired result for the sound engine. To compensate for this Harmor provide various image interpolation algorithms calculating various average values. In these drop-down menus, you can select between these items:

- Block
- Cap
- Bell
- Linear
- Cubic
- Rise. Allows only rising values.
- Fall. Allows only falling values.

To try these modes, I created the test images here below. Both are 31 by 16 pixels. You can download the image from here: <https://www.malmgren.nl/Harmor/ImageGain.png> and here <https://www.malmgren.nl/Harmor/ImageFreq.png>. Here below I scaled up the images so that you can see them. ImageFreq.png to the left and ImageGain.png to the right.



In the table below you find the results of the tests.

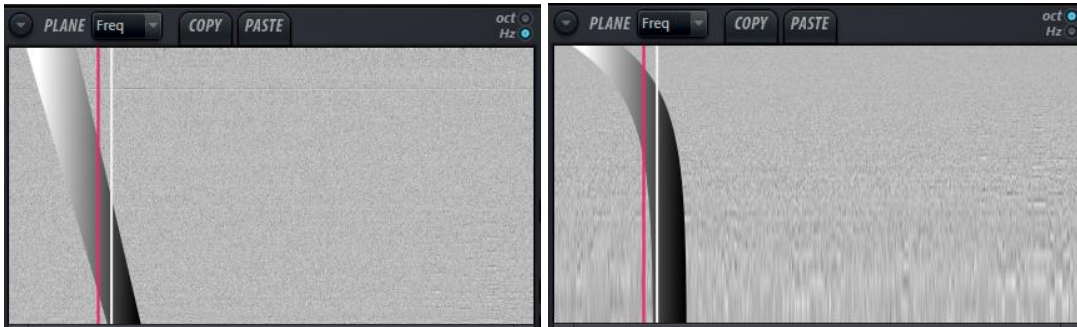
Here below you can see the different settings and the result:



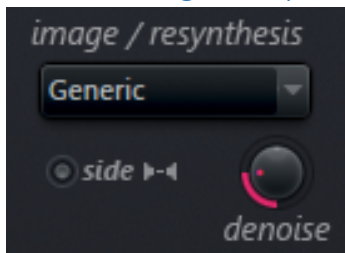


## Display scale

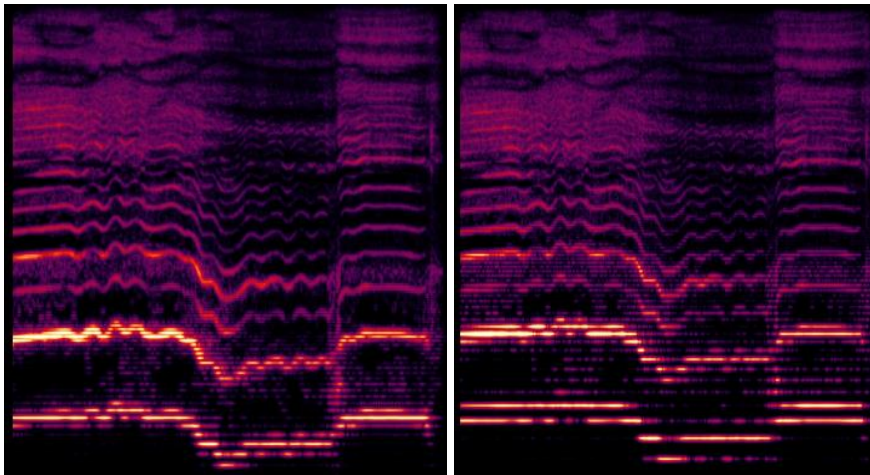
Above the image area you find the display scale switch *oct/Hz*.



## ADV tab image / resynthesis settings: quality mode



The default setting produce a reasonable approximation of the original sound. We can also change to *high precision*. With higher precision, Harmor is using less partials to produce the sound and each partial is used more efficiently. Higher precision can be more CPU intensive. You can consider using higher precision when scaling frequencies.



Here above I play FLS\_FreeStyle\_05 with resynthesis. The right image is playing in high precision mode and the right with generic mode. The high precision mode keeps the active partials more together and uses fewer partials

## ADV tab image / resynthesis settings: denoise

The knob *denoise* can be used to filter faint artifacts from the image.

## ADV tab image / resynthesis settings: Side processing / invert right channel - stereo

By default, Harmor provides mono resynthesis / image synthesis. To activate stereo, you will use part A and B for the right and left channel.

1. Activate part A. Load the audio data.
2. Activate part B, turn on part B. Load the audio data again.
3. In the ADV panel activate the switch: *Side processing / invert right channel*.



Now the resynthesis is stereo based. If you want to replicate settings for between part A and B simultaneously then activate the Part link button.

Although you can drag in audio directly from for example Edison you will save space by saving the audio to disk first and then dragging in the file from disk because that creates one copy whereas dragging it in directly from Edison doubles the data.

### Image options menu: Basics

In the resynthesis mode the image options menu displays the tools menu and the first two items of the audio file menu. In resynthesis mode the image and selected pane options are greyed out.

Tools	Audio: R09_0001	Image	Selected plane (gain)
Clear all	Analyze audio file...	Generate random cloud	Open image file...
Prepare time envelope	Convert to image synthesis mode	Degrade horizontally	Save as image file...
Create bouncing loop time envelope	Map audio markers to keys	Degrade vertically	Edit image
Slow down within loop region	Map audio regions to keys	Crossfade 10%	Copy to image clipboard
Tune pitch for time offset	Add points at audio markers	Crossfade 25%	Paste from image clipboard
Set viewed zone as loop region	Prepare time envelope for drumloop	Crossfade 50%	Clear
Set last played zone as loop region		Crop top	Invert
Map time offset to last hit key			Flip vertically

In image synthesis mode, the last two sections are available as well.

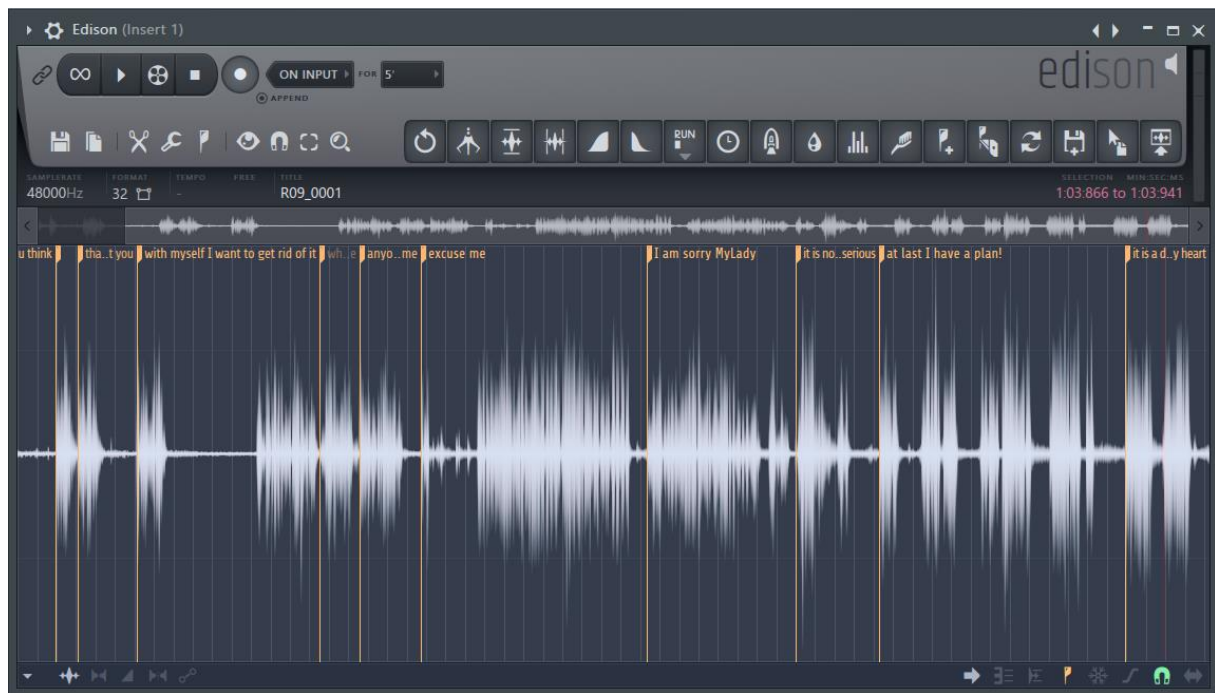
Tools	Audio	Image	Selected plane (gain)
Clear all	Analyze audio file...	Generate random cloud	Open image file...
Prepare time envelope	Convert to image synthesis mode	Degrade horizontally	Save as image file...
Create bouncing loop time envelope	Map audio markers to keys	Degrade vertically	Edit image
Slow down within loop region	Map audio regions to keys	Crossfade 10%	Copy to image clipboard
Tune pitch for time offset	Add points at audio markers	Crossfade 25%	Paste from image clipboard
Set viewed zone as loop region	Prepare time envelope for drumloop	Crossfade 50%	Clear
Set last played zone as loop region		Crop top	Invert
Map time offset to last hit key			Flip vertically

We already discussed *Analyze audio file* and *Convert to image synthesis mode*. In the following sections, we will discuss all items that aren't self-explanatory.

### Image options menu: Map audio markers / regions to keys

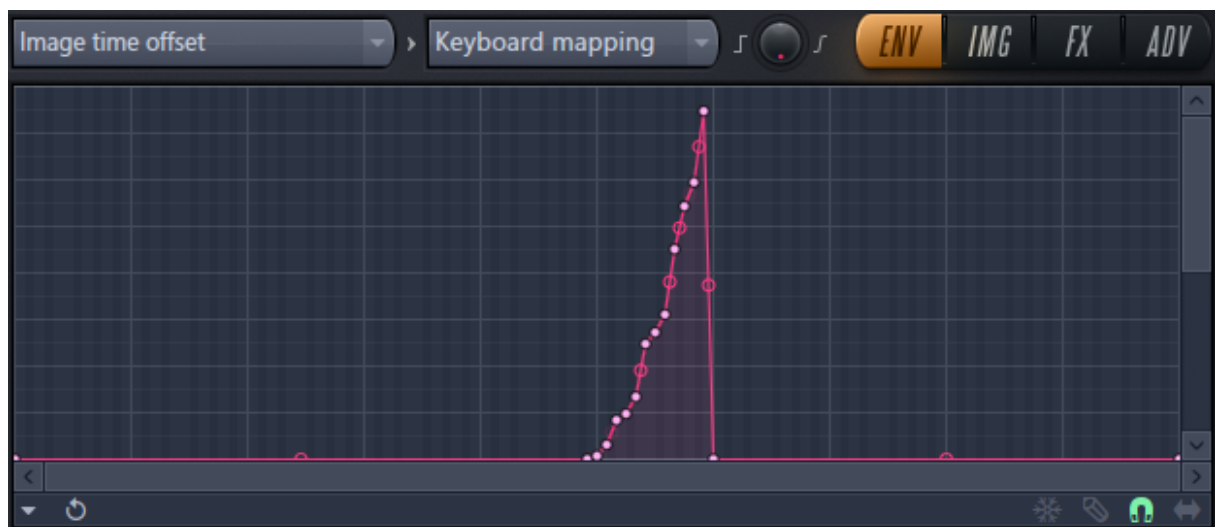
An audio file can be tagged with markers and it is possible to trigger a position in the audio file by pressing a key. This sounds like it could be used for a trick from a movie where a person's voice has been recorded and then the bad guys (or good guys) of the movie are calling somewhere and triggering words or sequences from the prerecorded voice as if the person is saying it himself. Okay let's do this!

We start by loading sounds in Edison. At each spot where we need a marker you select the sound and then press M. A little window appears and you can name the marker. With Alt + M you can set a region.



When you are done, you save the file. In Harmor you then load the audio file. With an audio file with markers the audio marker menu item is lit up.

Map audio markers to keys  
Map audio regions to keys



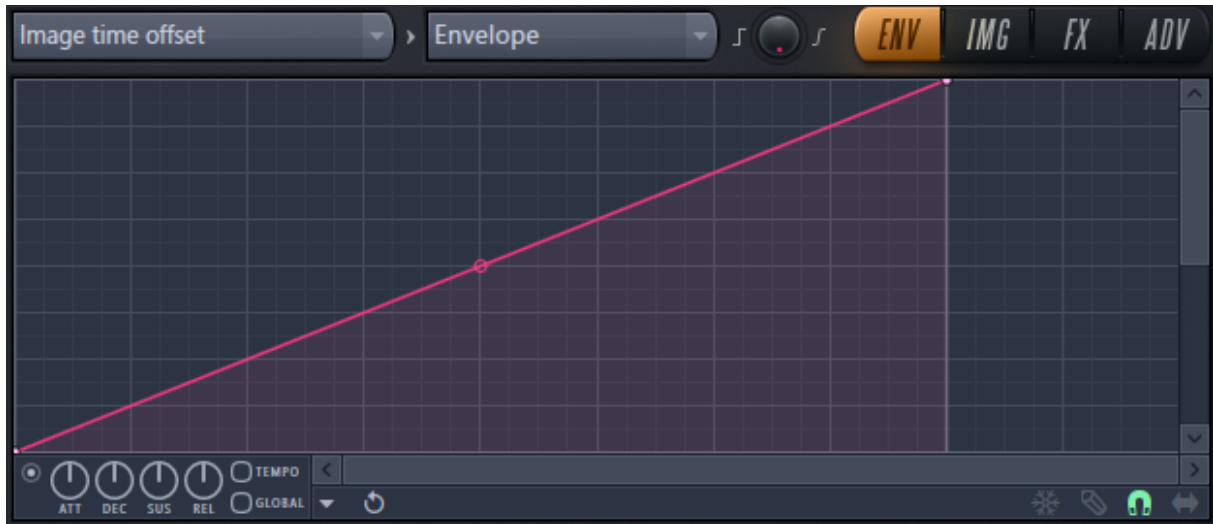
Now keys are programmed to trigger points in the audio. What's left? Remember the key for each phrase to play during the phone call. Good luck.

### Image options menu: Prepare time envelope for drum loop

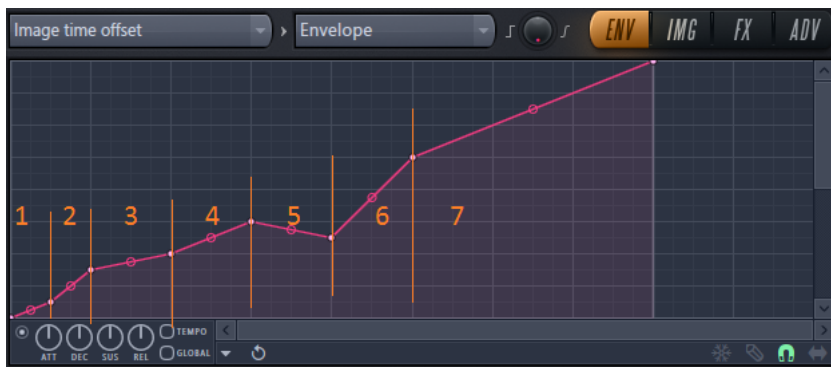
What we did in the previous section can be applied to drum loops as well, each part of the audio clip can be triggered by a key. For this purpose, there is an analysis function. To try this open Packs -> Loops and drag in a drum loop into the image window. Select *Prepare time envelope for drum loop*. Now you get an *image time offset* target with a *keyboard mapping* articulator just like in our previous section.

### Image options menu: Prepare image time envelope

This function is not preparing so much. It sets up this envelope – and that's it. When this envelope is played the x axis is converted to positions in the image.



This will just play the image as we are used to. This can be modified.



Here are the special features of each section of the modified graph:

1. This has the original slope, hence the regular speed.
2. Playback goes faster.
3. Playback goes slower.
4. Regular speed.
5. Playing backwards slowly.
6. Playing forward, quick.
7. Regular speed.

### Image options menu: Create bouncing loop time envelope

I cannot see any difference with the previous function.

### Image options menu: Slow down within loop region

This function sets a dot in the image fine speed envelope. If we change this to a sloping envelope, then this will decrease the fine speed relative to the fine speed knob.



### Image options menu: Tune pitch for time offset

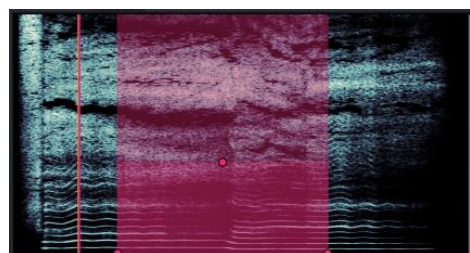
This function tune the pitch exactly at the position where the time offset is placed when you select this option.

The pitch envelope window opens with a dot placed at the left edge at the vertical pitch position. The pitch envelope refers to the main pitch slider. Let us consider a couple of questions regarding this feature. When did you change the pitch a few cents up or down? Perhaps you lowered the pitch with -600 cents or -1200 cents to create a bass sounds. Suppose you would like to tune your sound then do you really want to do it with the main pitch?



### Image options menu: Set viewed zone as loop region

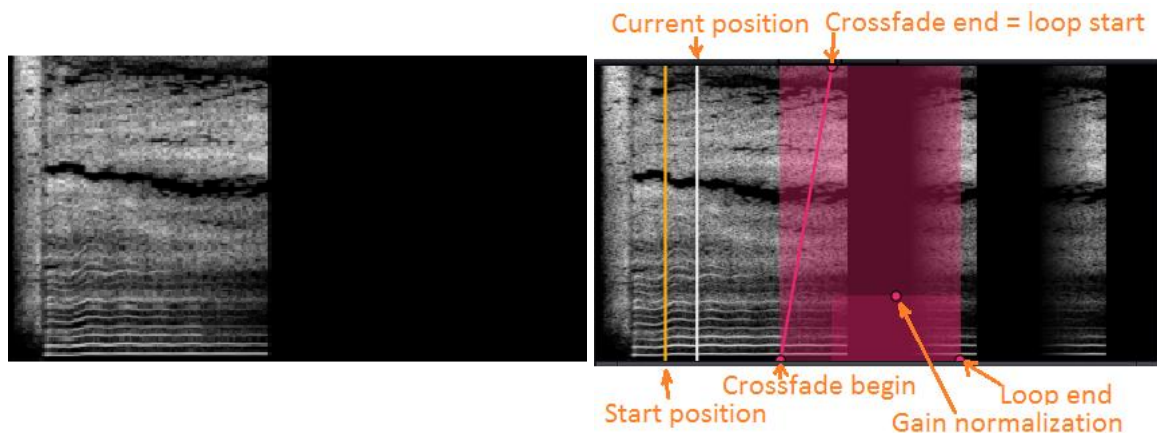
Before using this function, you could zoom in on the region you are interested in. Then when you selected this option a part of the window will be marked red. Then you can zoom out again.



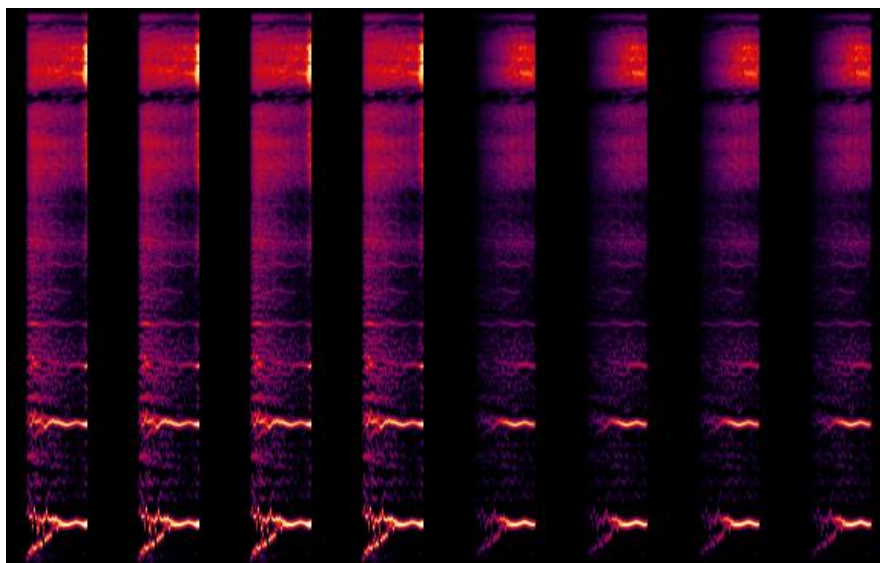
If *looping* is set to *one-shot* mode, then this region is ignored. In the image to the right the playback would start at the red line and passes through the red region. There are three red dots. The left and the right indicates the beginning and end of the region. The middle dot is the handle of the gain normalization control. Gain normalization is inactive in the one-shot mode. You could wonder why we get that normalization mode when it is not doing anything. I don't know.

In the ping pong mode, the playback can start before the region but once inside in the region the playback starts ping ponging inside. Setting the start cursor after the region while the current position is inside the region will not start from the start cursor – it will start from the current position.

When *looping* is set to *Loop* mode that is the only mode where the gain normalization has any effect. To try this I analyzed a sound and then I erased one part of the sound like in the image here below to the left.



In looping mode, we get crossfade controls as well. The area behind the loop end is overlapping the with the amount of the crossfade (between crossfade begin and end) to a maximum of half the zone. You can drag the crossfade end marker to the right to decrease the loop start and decrease the crossfade overlap.



Here above the first four loops are played with maximum gain and the four last are with minimum gain.

Right click the begin or end of the loop region to remove it.

[Image options menu: Set last played zone as loop region](#)

Play a part of the image and then select this function to make the played zone as a loop region. For the rest this is the same as described in the previous section.

[Image options menu: Map time offset to last hit key](#)

To use this function, press a key. Then set time offset. Then select this function from the option menu. Repeat this as many times as you like.



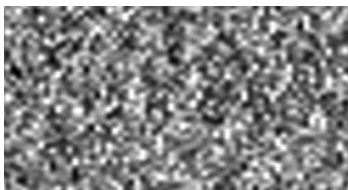
The result is points in the Image time offset editor target, in the Keyboard mapping articulator part. If you would like to start over, then you could erase the points in the graph and start over.

Another option is to edit the points directly in this graph.



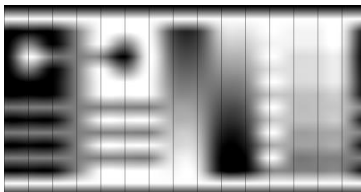
[Image options menu: Generate random cloud](#)

Gives you random pixels.



[Image options menu: Degradate horizontally](#)

This option smears the image horizontally. Here below I used the test image we used earlier.



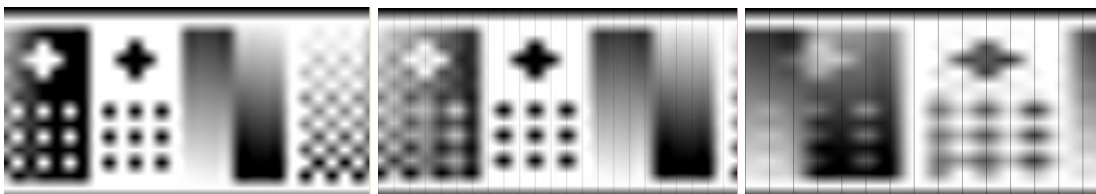
[Image options menu: Degradate horizontally](#)

This option smears the image vertically.



[Image options menu: Crossfade 10%, 25% and 50%](#)

From left to right here below 10%, 25% and 50%.



[Image options menu: Crop top](#)

Removes the top of the image.



#### Image options menu: Invert

Inverts the image color information.



#### Image options menu: Flip vertically

Flips the image vertically.



#### Clipboard monitoring

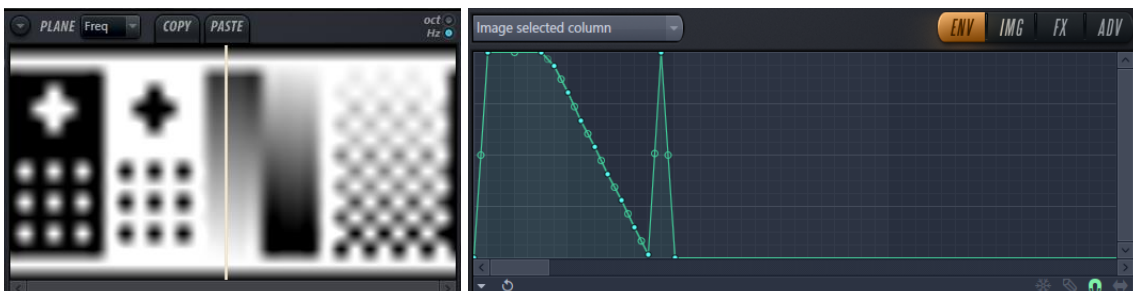
Harmor can monitor changes to the clipboard memory. You turn on this feature by right clicking the paste button. When doing so the button Copy will light up in red. Now you can for example start playing the clip and when you copy an image in an image editor Harmor will detect the new image in the clipboard and automatically paste it into the image window of Harmor and play it in the currently active plane.

#### Edit image

This will open an external editor for editing the image. The default program for the image editing will be opened. For most people this is Paint.

#### Image selected column

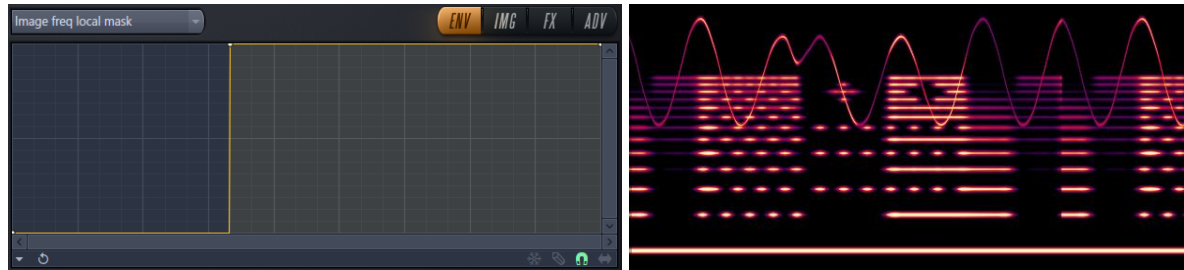
This feature makes it possible to edit one row of the current image inside Harmor. For this to work set image speed to 0. Now when setting the cursor in the image and clicking in the image you play one specific column of the image. This last played column will now be displayed in the graph of editor target Image selected column.



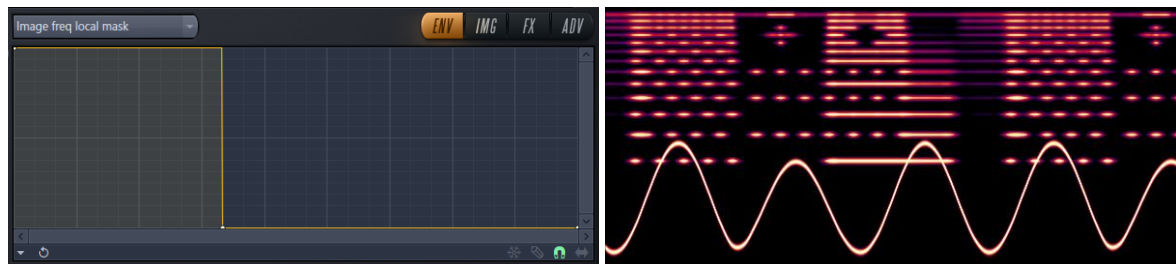
If you change the graph in the editor target window, then the image will reflect your changes.

### Image freq local mask

This editor target graph makes it possible to cancel frequencies from the frequency plane. With a couple of examples from our test image, it will be clear what this means. In our first example the low frequencies are filtered.



Here is the reverse situation. Here are the higher frequencies filtered. The frequency plane is the same in both cases.



## Chapter 23, Advanced targets

The first part of this book talked about the knobs and sliders that you can see while we carefully avoided the envelopes, articulators and drop-down menus. Then in chapter 21 we talked about the basic concepts of targets. Since the image synthesis / resynthesis has a whole slew of targets that chapter came right after the targets basics. Now in this chapter we will collect all the remaining targets, see how they work and how to use them.

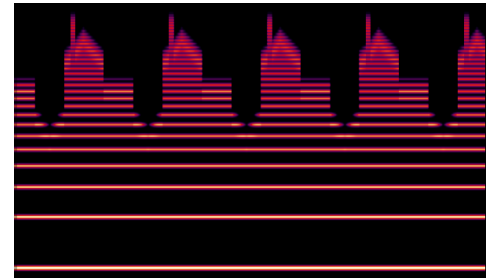
### Filter envelope amount



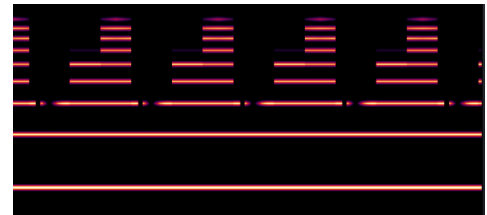
In the filter section, we have two *ENV* knobs that are useless in the default configuration. To make any use of these knobs, we need to configure the envelope for the filter frequency as well.



Right click *FREQ* and select Edit articulator. This way the Filter 1 frequency target and Envelope part will be displayed. I decided to draw a house as the filter frequency envelope. Turn on the envelope in the lower left corner. Here I set the begin to loop start and the end to sustain / loop end. Then I added a bit of resonance as well.

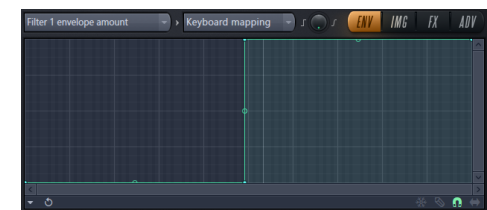


Then we turn the envelope knob to -30%. Now the house is turned upside down.



Now here comes a little bit of confusion because when you turn the ENV knobs they display 0% in the middle and on the left side they display negative values to -100% and on the right side +100%. Normally throughout the rest of Harmor this would make the envelope Bipolar product or Bipolar sum but in this case, it is Unipolar product. This means that you can program the one half of the ENV knob with the Filter envelope amount. Either you produce 0% or a negative or a positive value but you cannot in the same graph produce both. I think it is a bug.

To try this reasoning in practice set the *ENV* knob to +100% and right click the *ENV* knob and select edit articulator. Enter a graph like in the image to the right. Now when playing notes C5 and higher the building will be displayed in the visual feedback. When playing notes below C5 there will be no building "just flat ground". This is because the minimum value is 0% and not -100% as we would like it to be.



Now turn the ENV knob to -100%. Now when playing notes over C5 we get the reverse image. When playing notes under C5 then we get just the "flat ground".

While we are at this part of the filter let's talk about the Adaptive filter 1 envelope mode. According to the manual the bandwidth of the filter changes width as a function of the notes frequency, so that the duration of envelopes applied to low vs high notes sound similar in duration.

### Global envelope ADSR settings

For this to work we first need to turn on the volume envelope by turning on the switch in the left lower corner. Now suppose we want the attack to be quicker based on various factors. For example, based on what note we are playing.



To do that we open the articulator Global envelope attack time. Now higher notes get quicker attack time and lower notes get slower attack time.

There is also an envelope for the attack scale, release scale and decay scale.

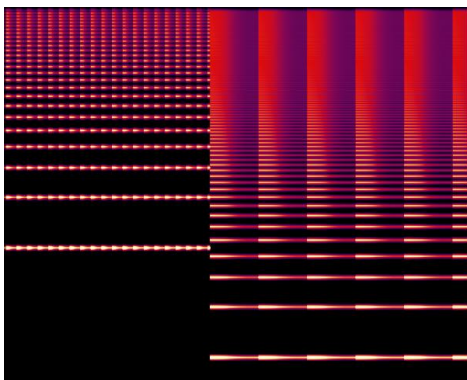
Sustain offset can be moved up and down with an envelope as well.

### Global envelope LFO settings

For this to work we first need to turn on some LFO volume envelope. Next, we open the global envelope. To turn on LFO go to the volume editor target, switch to the LFO articulator part. When this works as you like you switch to the global LFO speed for example.



Then when we play a low note the LFO speed is slow and a high note gives a quick LFO.





## Global EQ

Equalizing is that frequencies are amplified or made quiet. In Harmor there are two EQ systems:

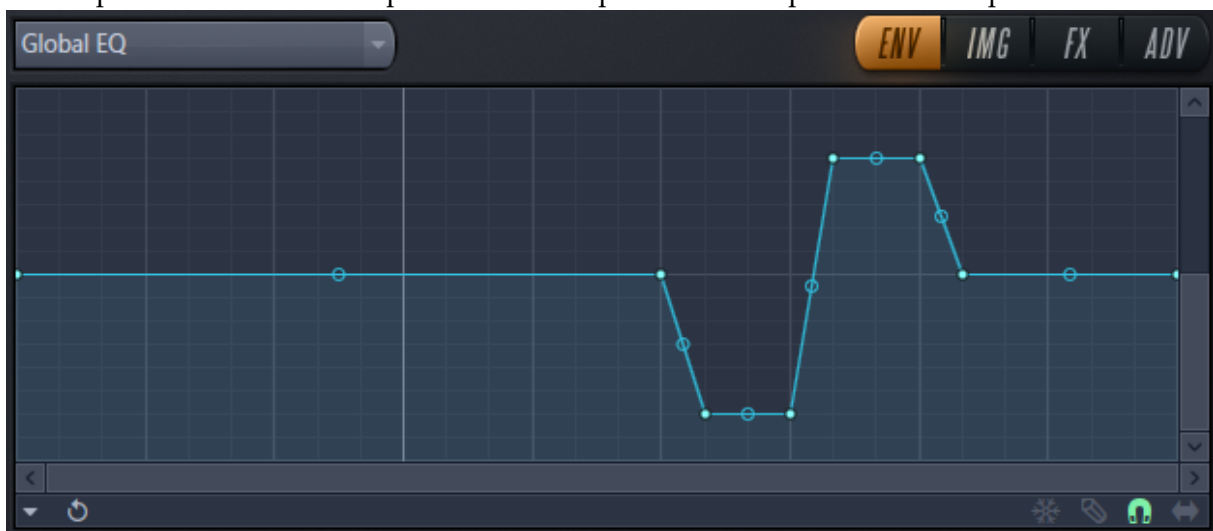
- Global EQ: Is equalizing regardless of what notes are being played.
- Local EQ: Is equalizing and following notes being played, see the next section.



Both these systems operate on the partials before the data is converted to a regular sound signal.

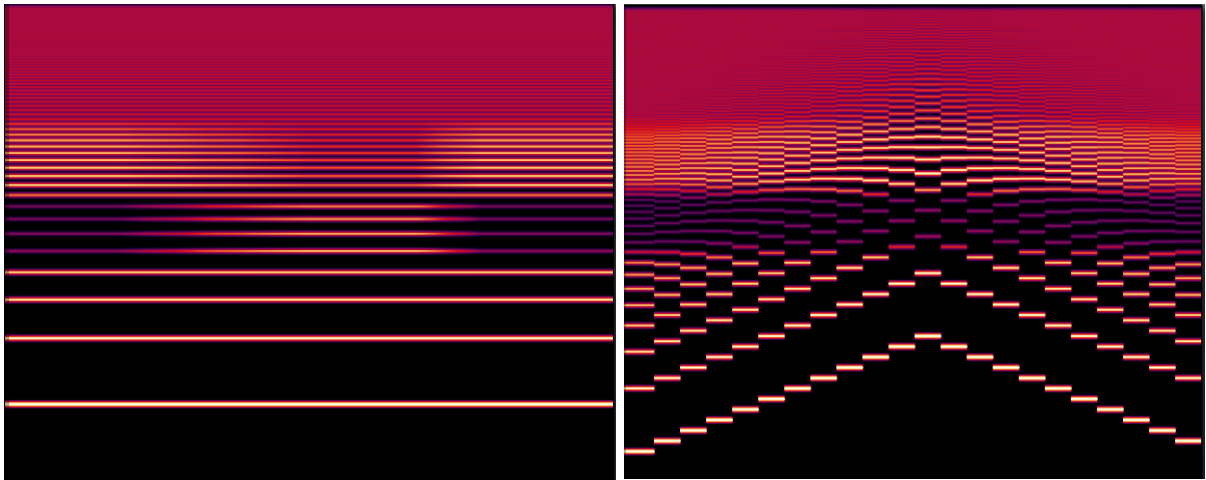
The Global EQ has a knob in the UI. When you right-click on this knob the global EQ envelope editor target window will open. This is the same as selecting editor target global EQ directly. Actually, the knob is not doing anything until the graph has been changed. So let's do that.

If we make this graph that you can see here below then it specifies that from C6 up to about D7 frequencies will be more quiet. From D7 up to E8 the frequencies are amplified.



The Global EQ has no influence on the timbre, it stays as it is. When playing a note with this graph and lowering the EQ knob to 0% and then increasing it back to 100% the result is the following feedback, see the left image here below.

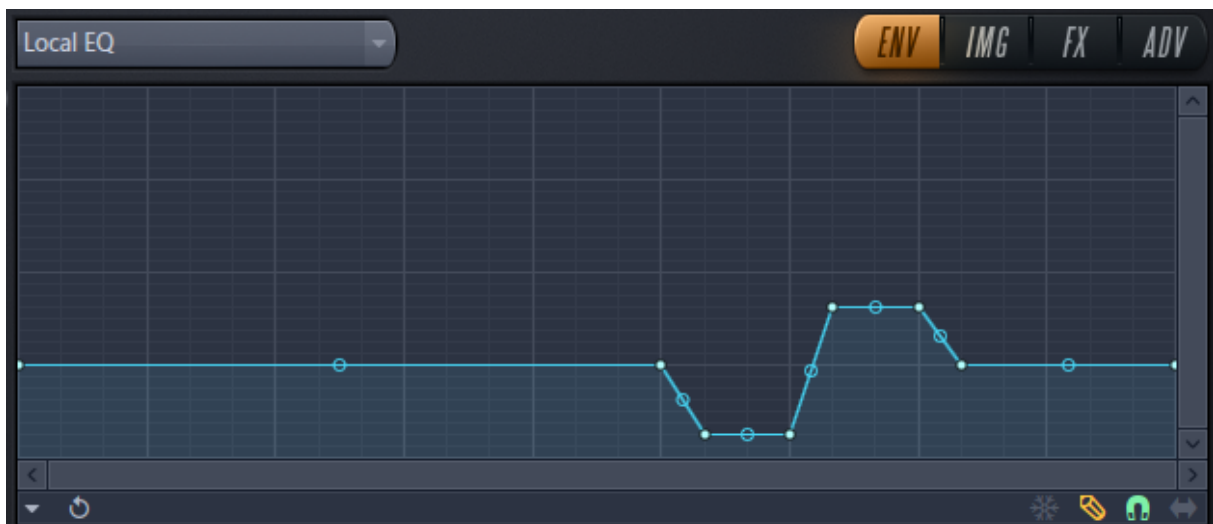




When playing notes, the Global EQ profile stays where it is like you can see in the image to the right.

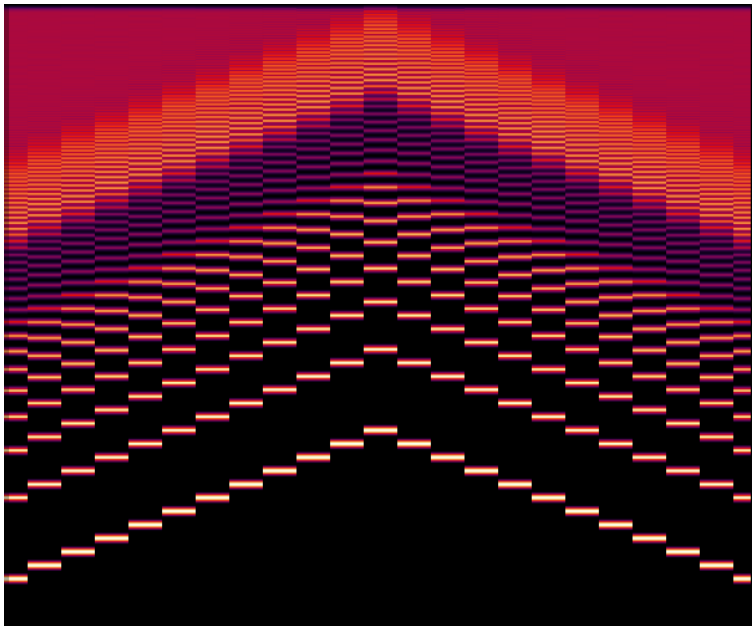
### Local EQ

The local EQ in Harmor follows the notes being played. To modify this envelope, select Local EQ in the editor target drop-down menu. Suppose we program this graph:



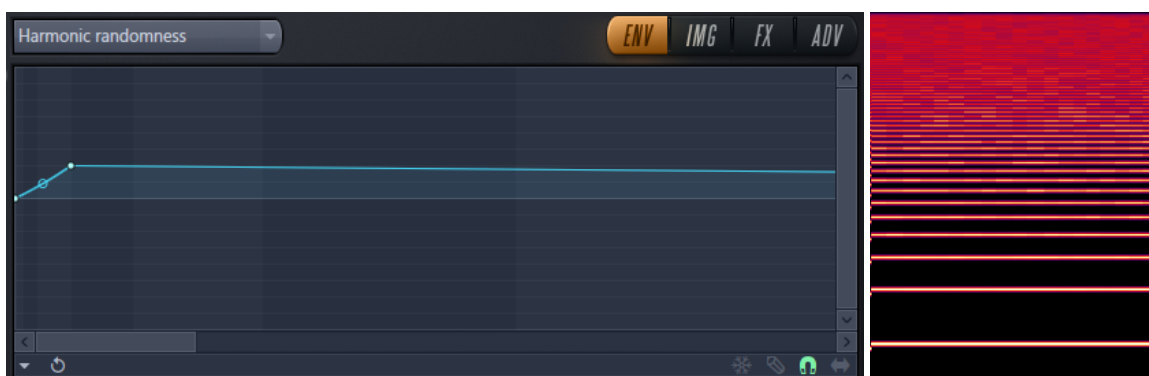
This graph also modifies the timbre because these partials are amplified and made more quiet. In this case the timbres becomes a little “pointier” than normal, not a very big difference.





### Harmonic randomness

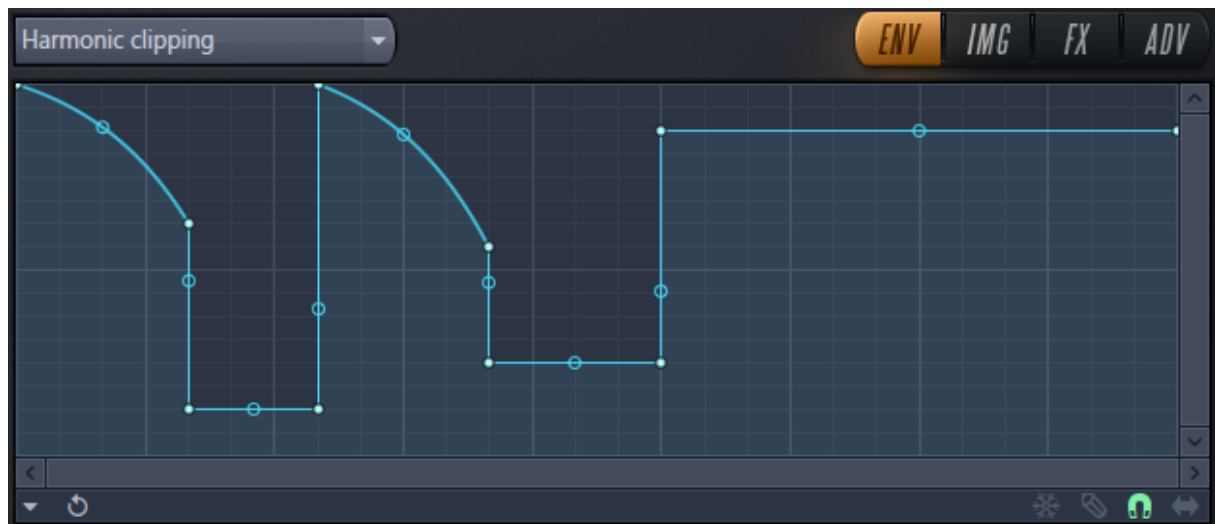
Suppose you want to introduce a little bit of randomness in the waveform. Then open the harmonic randomness editor target and edit the envelope.



Every time a note is played the partials are getting a little modification. The modification is not changing constantly but stays as long as the note is played. The next note has other random changes.

### Harmonic clipping

In *Harmonic clipping* threshold, harmonic clipping was explained in detail. In that chapter, we did not talk about the harmonic clipping editor target envelope window so let us do that now! Harmonic clipping works like a sort of filter and the default setting for this filter is a flat line. Here in this window it is possible to specify more elaborate clipping envelopes. Anything is possible.



### Harmonic prism

In *Chapter 8. Prism* we discussed everything about harmonic prism, except how to change the default curve so it is about time to talk about that here! Here below you see the default prism envelope and what it produces when we are changing the prism amount knob.

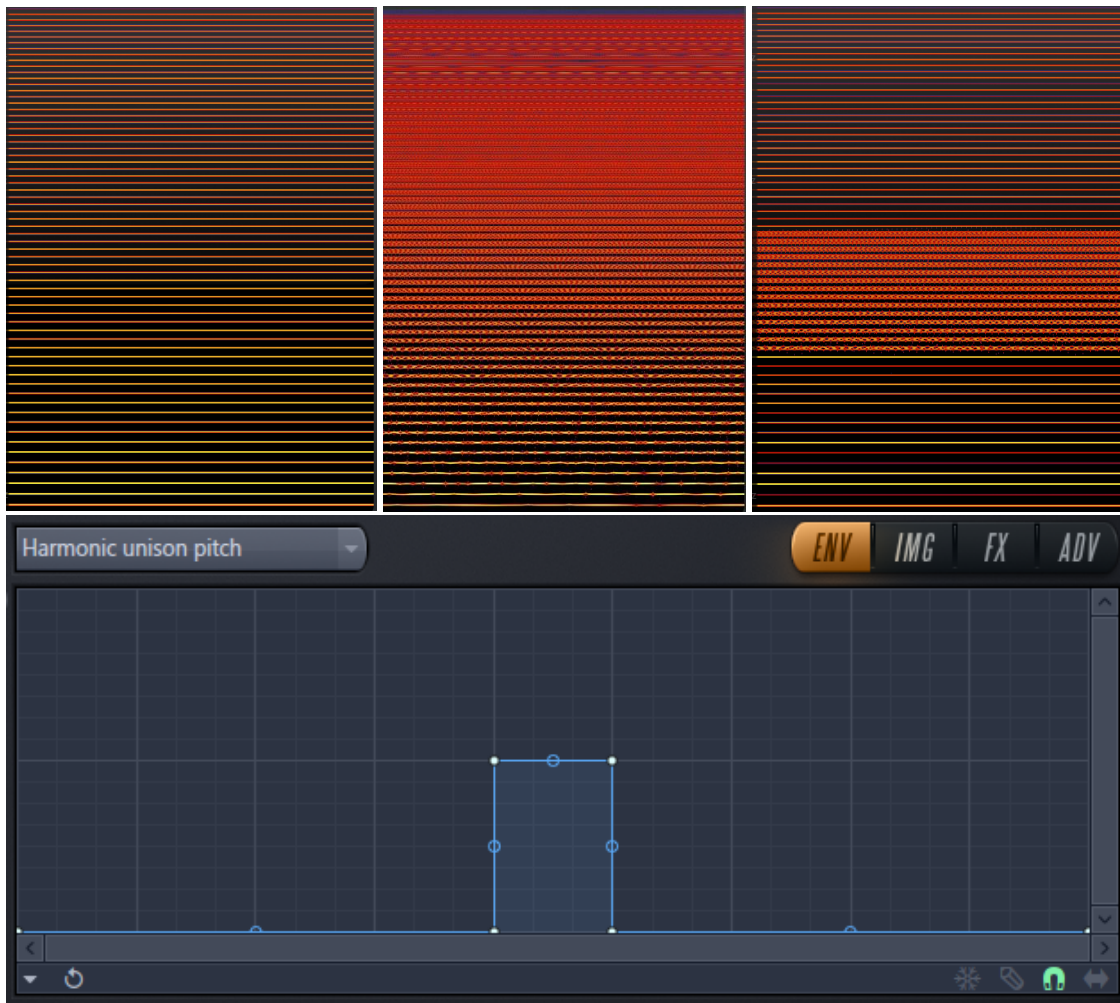


When we modify the prism graph it looks different.



### Harmonic unison pitch

In regular Unison, the effect is applied evenly across the frequency spectrum. We can change that. Here below to the left you see a saw waveform without Unison and in the middle with Unison, order 4, and evenly applied pitch. In the image to the right unison is only applied to a mid-section.



Here above Unison is applied to a small section of the frequency spectrum.

### Image formant mutation

In the *Image formant shift* section about formant mutations the formant shift knob moved all formants up or down in the frequency spectrum based on the format shift knob. With the image formant mutation envelope, it is possible to move formants of a specific frequency up while formants from another frequency are moved down.



I found that “Laurie Webb Ooh C” has plenty of formants to play with. With the graph modified the knob, formant shift, also moves all the formants but it is done in relation to the graph.



## Filter shape



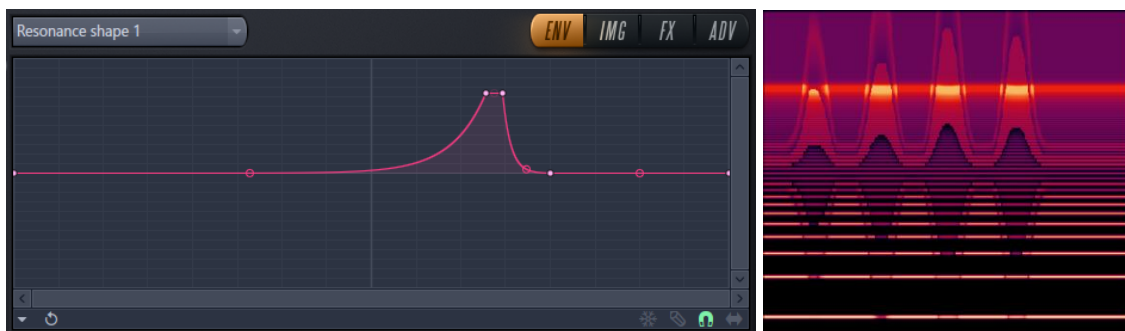
The filter shape drop-down menus got two custom shape options, one and two. When selecting a custom shape the editor target window will display for that shape.



Here I made a custom filter shape and while it I modify the filter width knob.

## Resonance shape

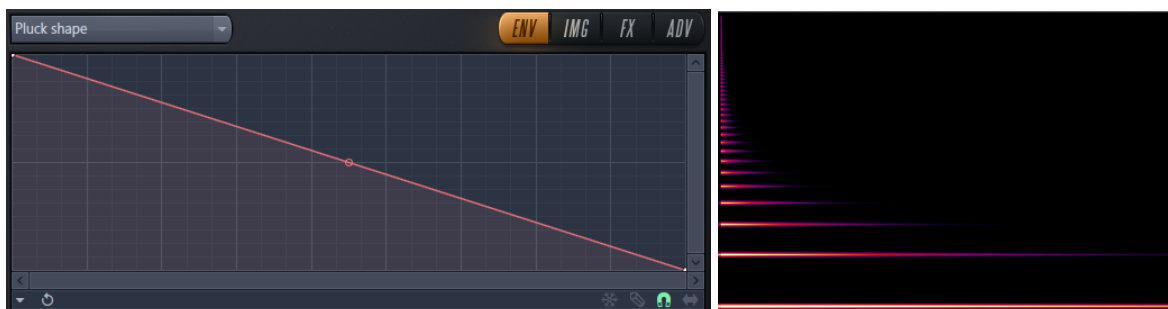
It is possible to define two custom resonance shapes as well.

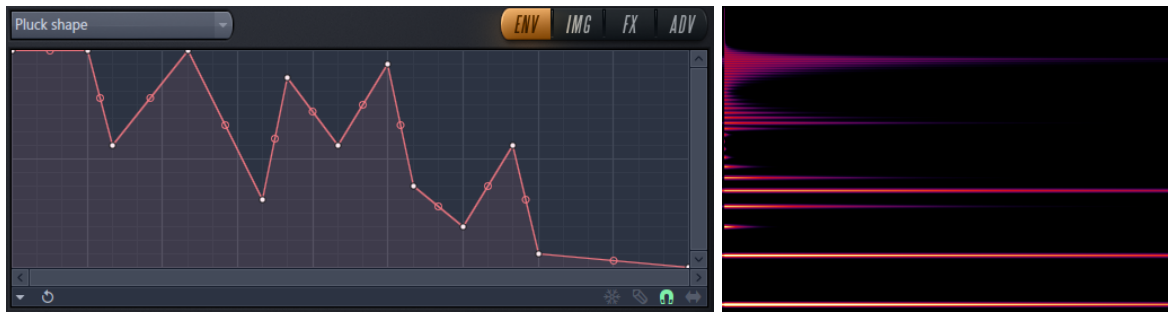


In the image, here above, the peak is lighting up in the visual feedback when played.

## Pluck shape

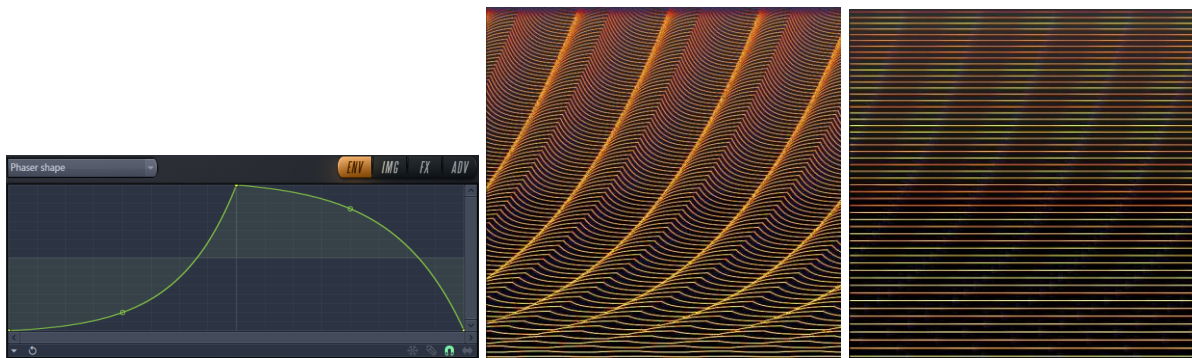
The default pluck shape is a straight line from the highest partial frequency and the shortest amount of time down to the lowest frequency and the longest amount of time. This is also normally how for example a string on a string instrument sounds ideally when being plucked. It is possible to change this behavior. Here below you see the straight line of the default pluck and how the visual feedback displays it and then below that a modified pluck shape and how that looks like in the visual feedback.





## Phaser shape

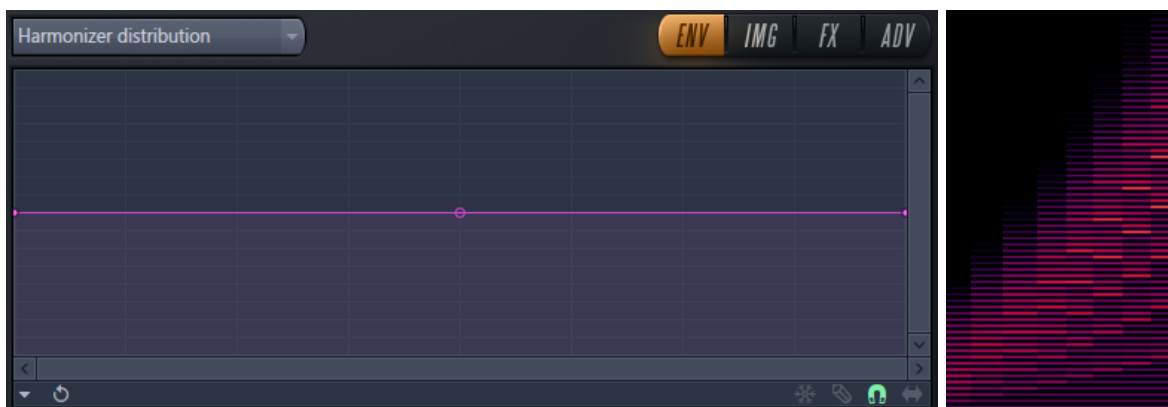
When you select Freq from the phaser drop-down menu then the phaser shape editor target window will open.

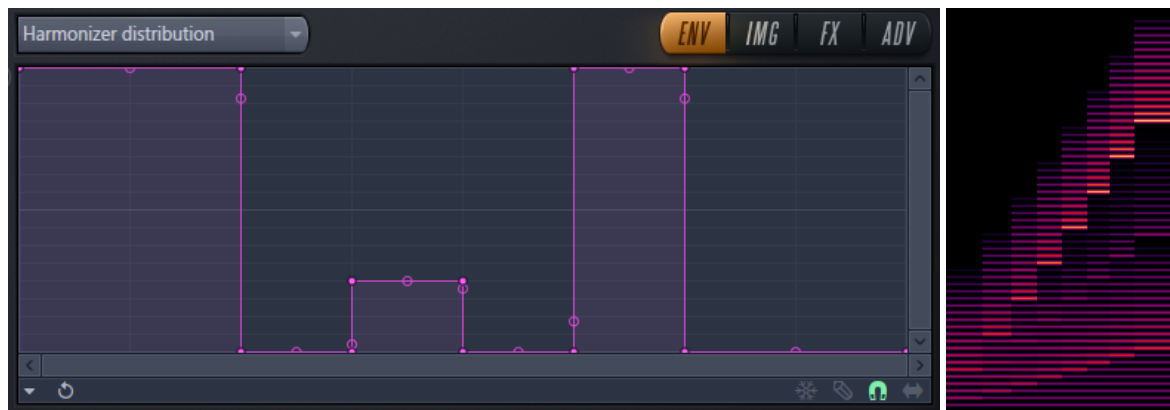


This graph here above to the left creates a sound that is decorative to look at in WaveCandy (middle image) but it is perhaps not so useful for regular music. The *freq* diagram is bipolar. When selecting the custom phaser shape you get the same diagram but then it is unipolar and far more modest and useful for music making, see the rightmost image here above.

## Harmonizer distribution

The Harmonizer effect, described in *Chapter 9. Harmonizer*, is duplicating and transposing harmonics based on the shift and gap controls. The different modes of '+' and 'x' gives various transposing at various steps. The default Harmonizer distribution "engage" nine steps at the same volume. This editor target graph makes it possible to increase or decrease the amplitude of any of the steps of the Harmonizer algorithm. Here below you find the default Harmonizer graph and on the right side of it you find the visual feedback of playing a note with gap at 0x1 and shift starting at 0+1 and gradually increasing this to 0+8. Below the default graph I placed a modified graph. Also this graph is played from shift 0+1 to 0+8.



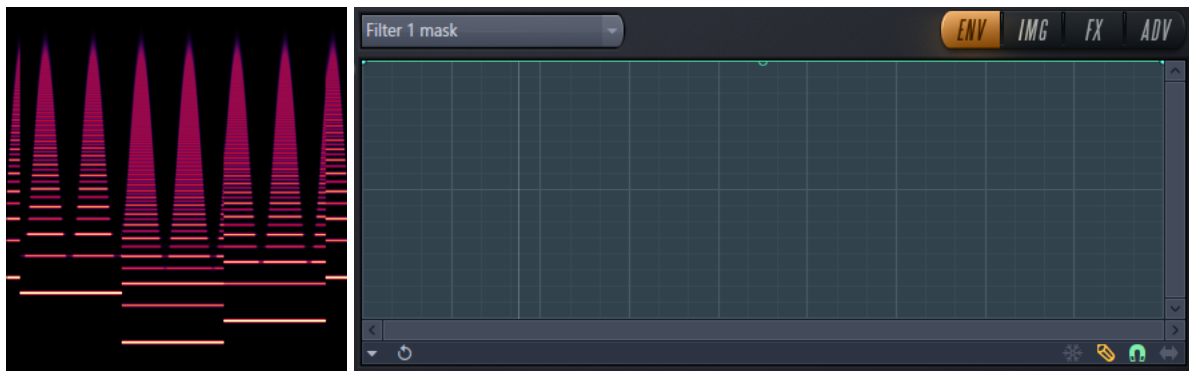


## Filter mask

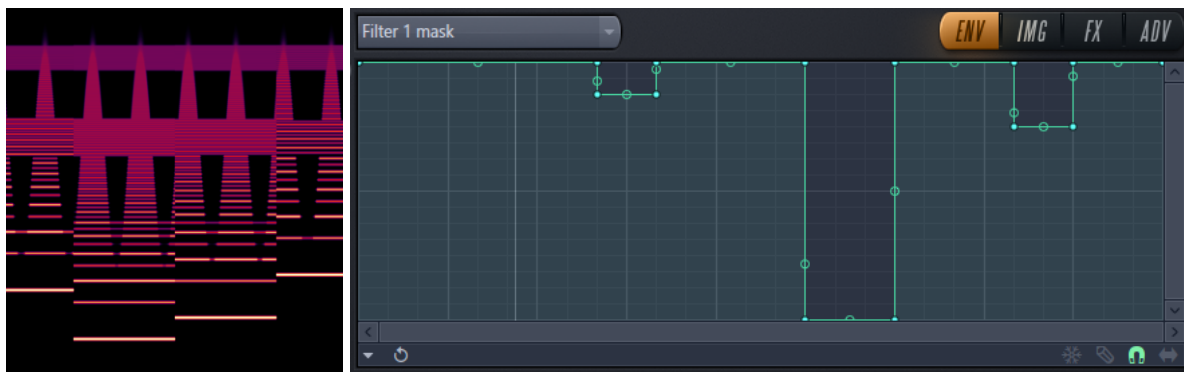
The filter mask specifies a frequency range where the original unfiltered signal should stay in the output. You could also see it as a graph that can gradually mix between dry and wet. To try this you can create a simple pattern playing for example a saw waveform. I created four long random chords. Then



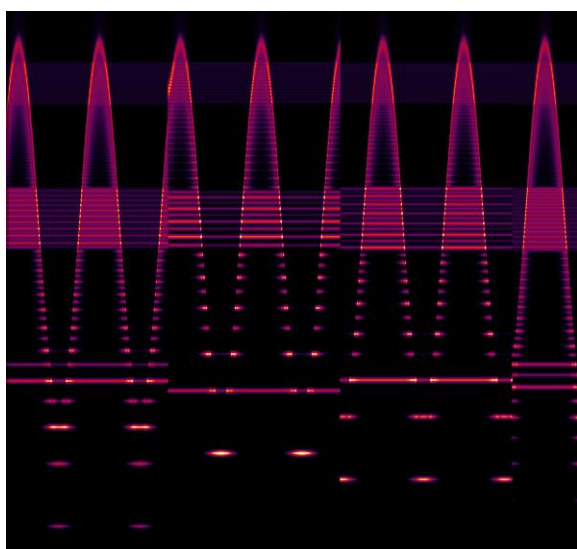
This project can be found at <https://www.malmgren.nl/Harmor/Harmor-51.flp>. I programmed the filter frequency to an automation clip. With a lowpass filter and the default mask this is the result here below to the left. On the right side you can see the default filter mask.



When the line is at the top all of the frequencies are assigned to the filter. Here below the filter mask has changed. The upper part is the “wet” filter side and the bottom part is the “dry signal” side. The lowest frequencies at the left and the highest frequencies to the right. Where the line is nearing the dry signal in the frequency range there you can see the unfiltered waveform.

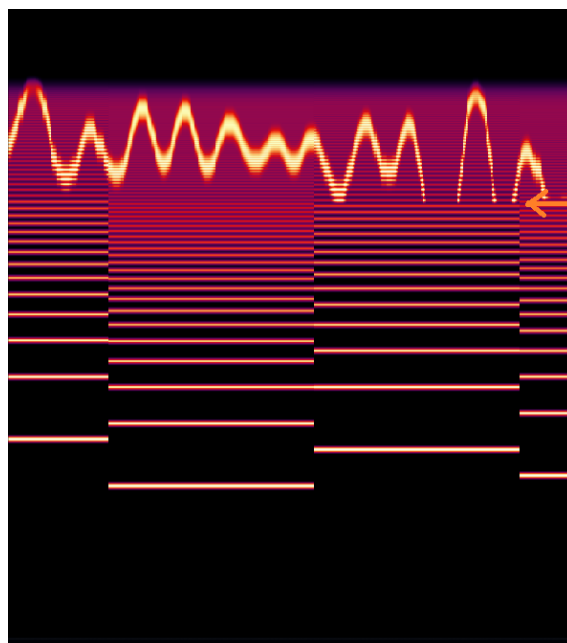
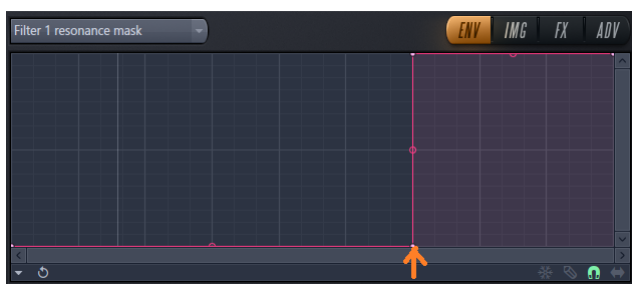


There is more to this functionality! Here below I changed to classic bandpass, increase the resonance a bit. That will fade out the dry part of the mask in the “background” but keep the mask in the filtered part.



### Filter resonance mask

The resonance mask works like the filter mask except resonance is a part of the filter effect. One way to illustrate this is to set an abrupt mask and then adjusting the resonance offset. For this I created the project <https://www.malmgren.nl/Harmor/Harmor-52.flp>. In the image to the right you can see how I change the offset manually. I inserted an

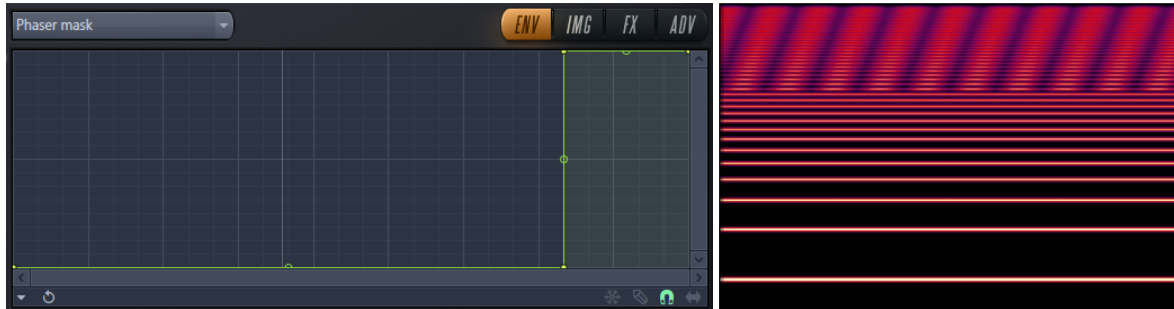


orange arrow to indicate where the mask cut off goes is.

In the image here above you can see the editor target of the resonance mask. Also here I drew an arrow for the cut off point.

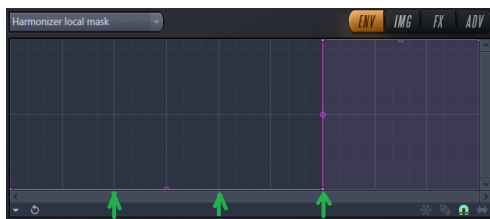
### Phaser mask

The phaser mask editor target can turn off the phaser effect partially. Here below you can see the editor target graph to the left and the result this gives to the right.

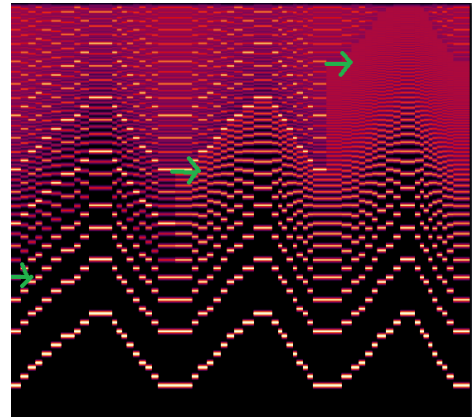


### Harmonizer local mask

The harmonizer mask editor target can be used to turn off the harmonizer effect partially. To the right, you can see an example that I played on the keyboard starting at C4 playing up to E5 and then down again. I played this pattern three times and each time I changed the



Harmonizer mask as indicated by the green arrows. Local means that this mask is relative to the note.



### Blur local mask

To illustrate the blur local mask, I created a more elaborated setup. First, I analyzed a sine wave shape so that only the fundamental frequency was played.

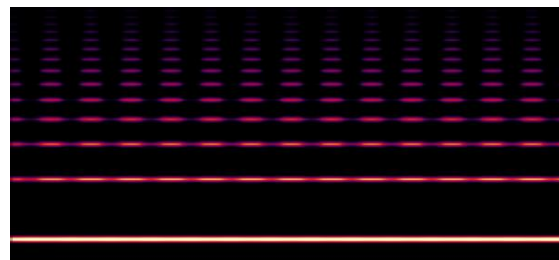


Then I enabled the blur settings according to the image to the right. In addition to this I right clicked the harmonic blur amount knob, *harm*, and selected edit articulator and then I



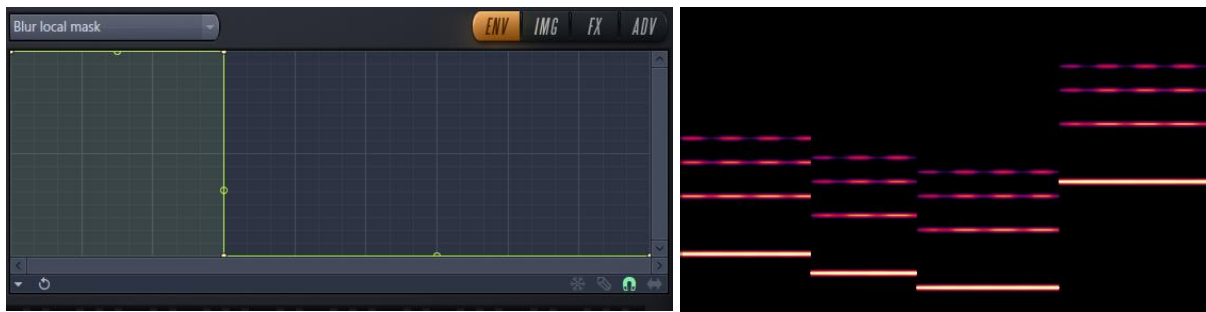
selected the LFO articulator part and programmed a nice sine wave.

Now when playing notes, they were provided with a distinct blur look. Then it was time to open the blur local mask editor target and enter





a graph like the image here below to the left. When playing this the top of the blur is masked and this is easy to see in the image here below to the right.



## Chapter 24, ADV Tab

The advanced tab in Harmor has both the power to redefine everything we learned so far and at the same time it is a little bit like the rests tab where the scrap features are collected; Both awesome and mundane at the same time.



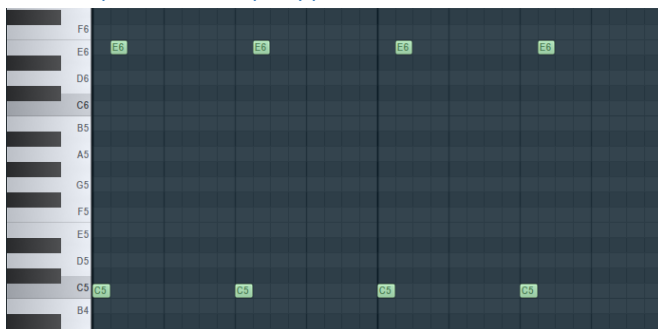
Notice the tag “Current part” referring to part A or B depending on what you currently have active. Global is the same for both A and B.

### Predelay

In the ADV tab you find the knob *delay*. The tooltip says *predelay*. This knob adds a delay from the moment that the midi note starts playing until Harmor is really starting to play. In the middle the knob is set to 0. The knobs left side gives the time for the delay in milliseconds and the right side gives the time for the delay in project time.

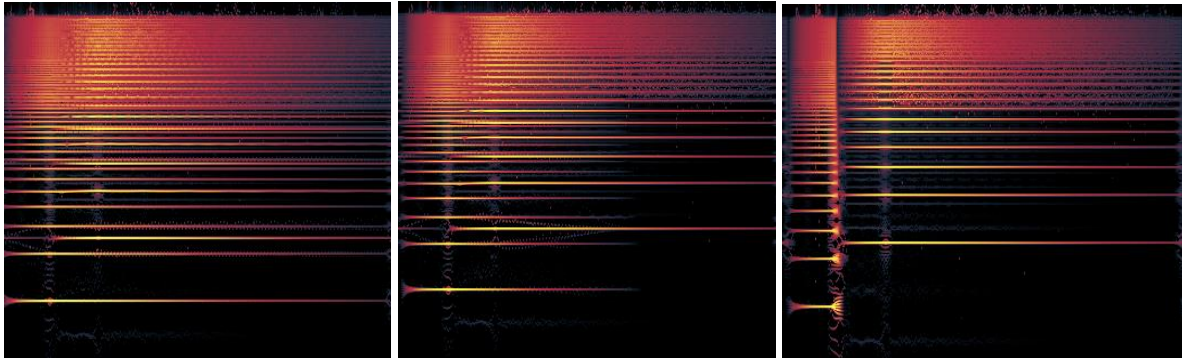
If you right click the knob and select edit articulator the editor target window opens. The upper part is used for the right side of the knob. The lower part of the graph is used for the left side of the knob. It is not useful to mix the upper part graph with the knob set to the left side.

### Envelope release polyphonic scale



This knob, *poly rel*, controls how the release section of an envelope interacts with other notes. When set all the way to the right to 100% one notes release can be played with other notes. When set to 0% all the way to the left the release of a note is stopped when other

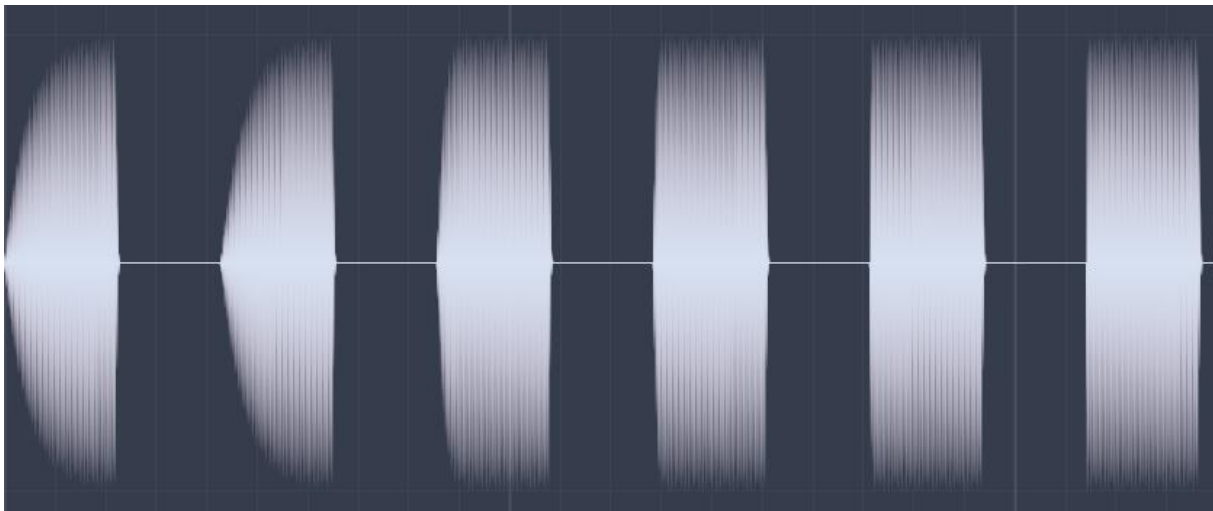
notes start playing. To experiment with this I created <https://www.malmgren.nl/Harmor/Harmor-53.flp> where I made five Harmor patches and for each patch a pattern with first a C5 directly followed by a E6. For each Harmor I set different polyphonic release scale. Here below to the left displays the result at 100% and the middle at 50% and the right the result at 0%.



In the left most image, it is clear that when the second notes start playing the release section of the first note is abruptly stopped, at 0%. In the left 100% and middle image 50% this perhaps not entirely noticeable but it is possible to hear.

### Ramping time

The *Ramp* knob is “kind of” doing the same to the partials as the strum mode up functionality are doing to the midi notes. Here below I play a note repeatedly. At first ramp is set to 200ms absolute, that is the setting when the knob is all the way to the left. Then I turned the knob and played the notes until the knob was at top position 0. You can see how that nearer to 0 the notes are started at full volume.

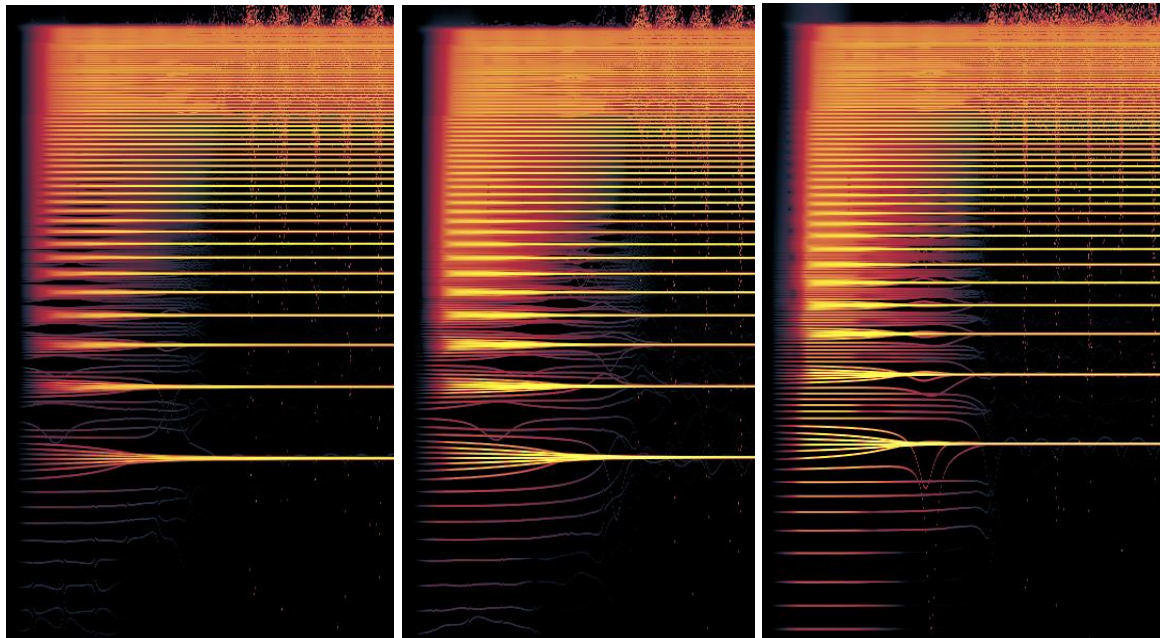


The right side of the *ramp* knob sets partial relative ramping time. It is not as evident as the left side of the knob but I will try to explain in anyway.

Here below to the left you can see the partial relative ramp set to 200 ms. The lower frequency partials are engaged with less volume is appears. In the middle image ramp is set to 25 ms partial relative and this is also the default value. The rightmost image is ramp of 0.

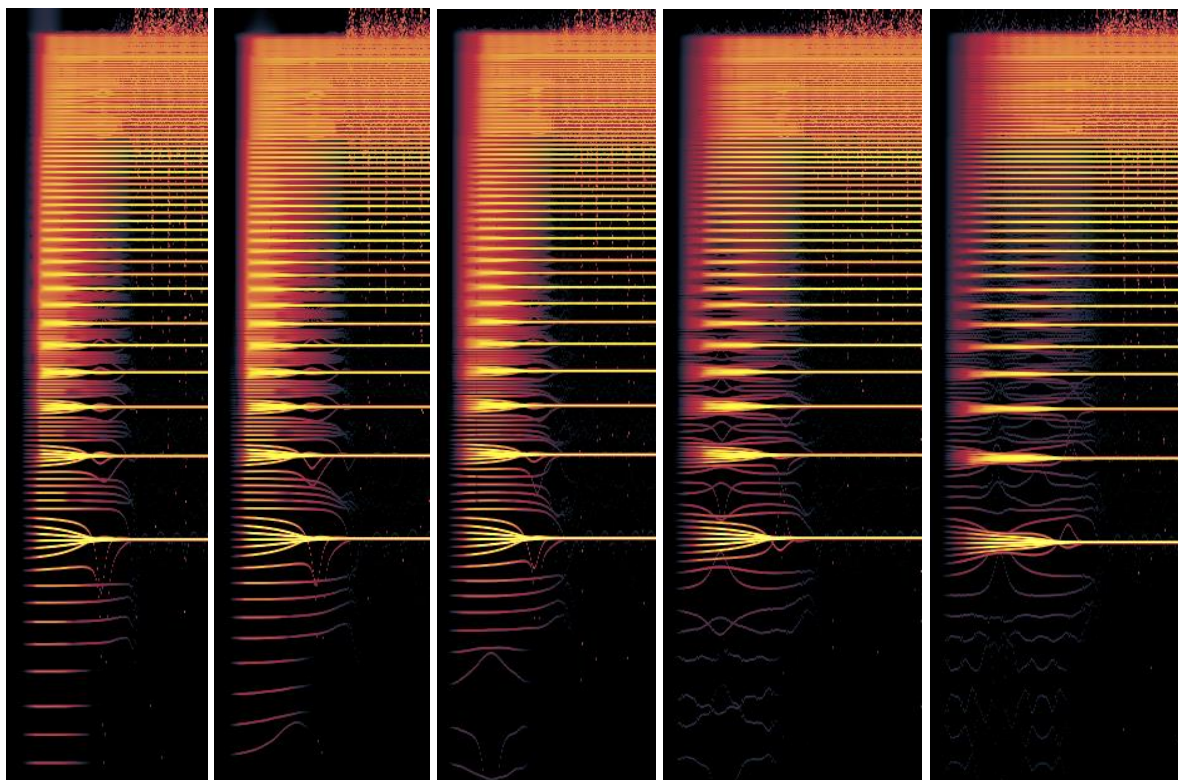
It is the space between the partials being influenced. At 200 ms there are holes between the partials from the start. In the 25 ms situation the holes are starting at the lower right from the beginning and for higher frequencies the holes are following sequentially.

The situation with ramp set to 0 gives that the partials are starting all at the same time.



### Declick

When starting to playing a sinewave a quarter into the cycle the engine need to produce volume at “full throttle” from start and this is tricky. The result is often a click sound. With additive synthesis, we add together all sorts of sine waves and with this partial processing it can be convenient to start gently. This is what the declick functionality is doing. Here below is an example of how it looks when playing with various declick settings. The declick knob goes from 0 ms all to the left to 25 ms all to the right. Here below I played the same notes with the following declick settings from left to right: 0 ms, 2 ms, 7 ms, 15 ms and 25 ms.



Notice how there are more room in-between the partials introduced and that the higher frequency partials are starting playing later and later. In the example images here above I set ramp to 0 ms.

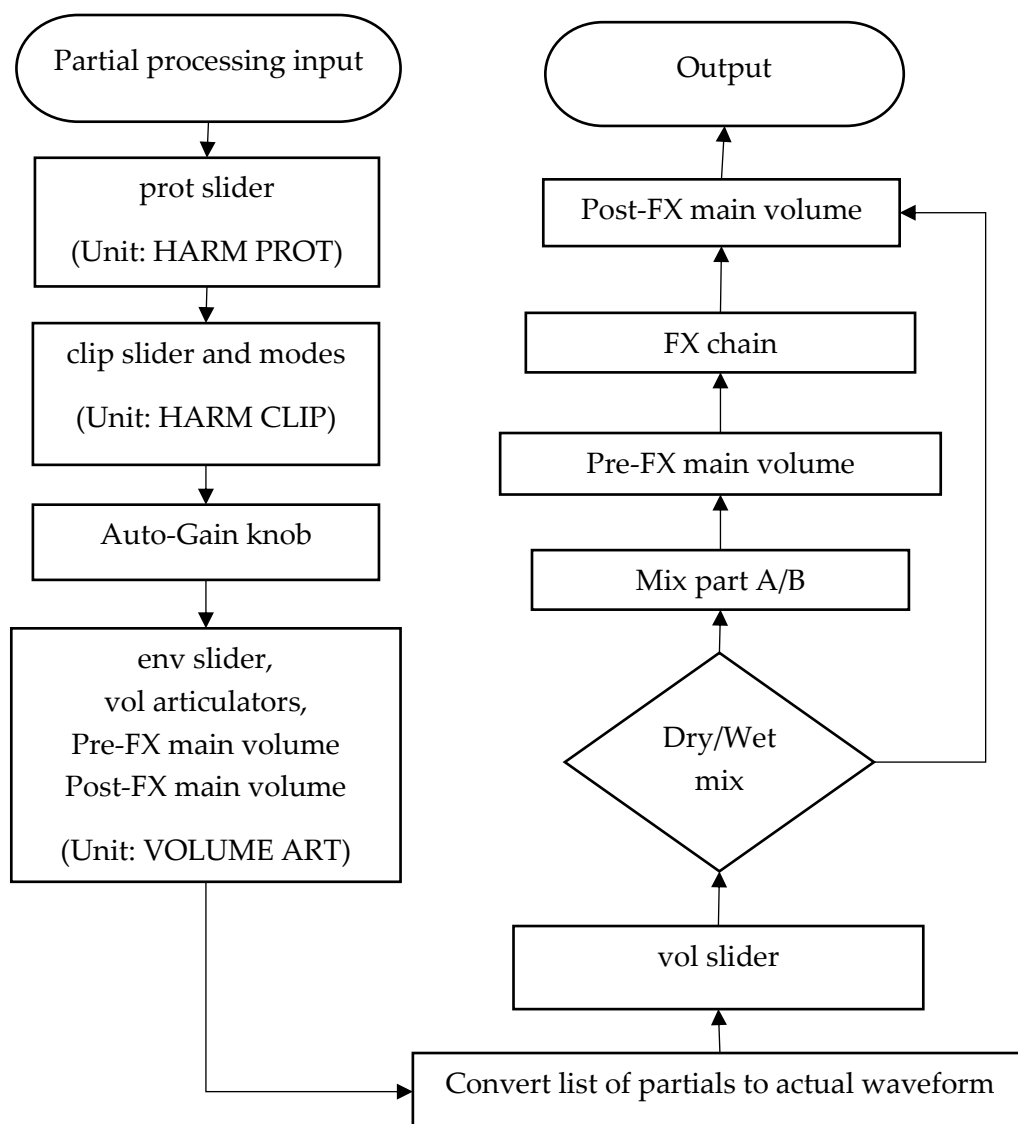
### Unit Order

The unit order section makes it possible to reorder the units of Harmor. Right click on an item in the list to move it up, earlier in the processing, or left click to move the item down later in the processing. Unit order applies to one part.

For your convenience, the items in the image to the right are made clickable (in the PDF version) so that if you would like to jump to the chapter and section about an item then click in the list.







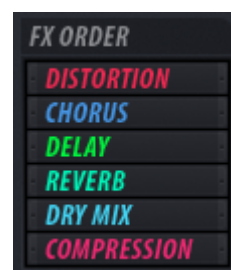
The *VOLUME ART* unit and the *Auto-Gain* step interact such as volume articulation will be applied after *Auto-Gain* when *VOLUME ART* is at the very end of the unit order list. If *VOLUME ART* is not at the end of the unit order list, then *Auto-Gain* happens after everything in the list. *Auto-Gain* itself cannot be directly moved around in the list.

### Effects order

The effects of the FX tab can also be reordered. The interface works the same as for the unit order. Right click for higher and earlier in the chain and left click for lower and later.

Also here if you are reading this document as a PDF you can click the item in the list to jump to the section talking about the effect.

The effects order is a global ordering applied to both parts.





## Auto-gain mode

This is a drop-down menu with two options:

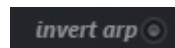
- Relative
- Absolute

Previously we talked about gain in the section *Auto-Gain* in chapter 4. I have not been able to figure out what the auto-gain mode drop-down menu is doing. I think that this mode moves the auto-gain module in the unit order but I have not been able to figure out how exactly.

I will one day talk to the developers at Image-Line about auto-gain. When I know more I will come back here and rewrite this section. Until then I would like to share with you the thought that in the “bigger picture” it does not matter so much how auto-gain works because in most realistic scenarios there will be a compressor after Harmor anyway and that compressor will among many things suppress challenges with gain that you might have.

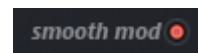
## Invert arp

In chapter 20 section *Arpeggiator points* we talked about the arp. This switch here in the ADV tab makes it possible to invert the arpeggiator points so that up becomes down and down becomes up, etc.



## Smooth mod

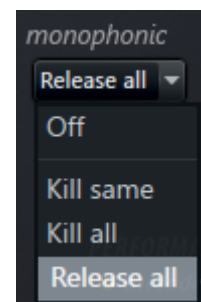
In chapter 21 section *Articulator part: Modulation X, Y and Z mapping* we talked about how to program the X,Y and Z MOD control to different articulators. The MOD control has a built-in smoothing function turned on by default. Suppose you want to make a “percussive” automation via the MOD control, in that case you might want to turn off the smoothing here.



## Monophonic

This drop-down menu specifies how midi notes will interact with each other, especially how notes are stopped from playing. These are the options:

- Kill same. When the same note is played again then the previous note is stopped first.
- Kill all. When any note is playing then they are all stopped before playing a new note, from one Harmor.
- Release all. New notes forces all playing notes to enter the release phase of their envelope. Look in section *Articulator part: Envelope* of chapter 21 for how to program the volume envelope.
- Off. This is the default monophonic value. No notes are stopped from playing.

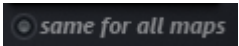


I understand that it is called “kill” but could we not find another expression that sounds less violent?

## Randomness

Random values are in general produced in such way that given a starting value, called “seed”, the randomization unit can produce a series of values that are random. With another seed another series of values are produced. With the same seed the same series will be produced. It is possible to extract a “good” seed-value from the built-in computer clock. The randomness drop-down menu specifies when the seed for random values should be produced. These options are available:

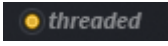
- Per voice, each note has a new seed.
- Per part voice, each part has its own randomization seed.
- Per unison voice, each unison voice has its own randomization seed.
- Off, Harmor has extracted one seed and from there all random values are produced.

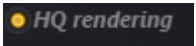
 Below the drop-down menu, you find the switch *same for all maps* and this option sets one seed for all maps as specified in the editor target drop-down menu - the second column “Mapping”.

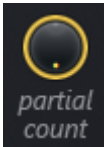
Now if you read the manual on this topic you will find a completely different story. What I write here is in this section is my best “educated” guess. I will need to double-check this with Image-Line.


## Performance versus Quality

All plug-ins by Image-Line has a sound designer mode and a render mode. When manipulating the user interface while making the sound, the music etc., then the quality is good but it doesn’t need to be excellent. When the piece is rendered FL-studio will switch to render mode and in that mode the quality is the best possible.

 The switch *threaded* is on by default and it allows Harmor to work with multithreading. Normally you keep multithreading on. If you have performance issues, then you could try to turn this off and in some cases, it might improve the situation.

 The switch *HQ rendering* enables high quality rendering when you start rendering. This switch is on by default and probably you want the result of your work to be produced with the highest possible quality so you leave this switch on.

 The knob *partial count* can be used to decrease the number of partials. The default setting is 516 partials when the knob is turned all the way to the right. When turning the knob to the left partials are removed starting at higher frequencies. This has the effect that less CPU is necessary to produce the sound but the sound will lose the higher frequencies. The minimum value is 12 partials.

 The knob *safety* is used to specify the safety ceiling. This knob should determine how many partials above the root note are played but I see no influence of this knob at all. I suggest leaving this knob at the default value.



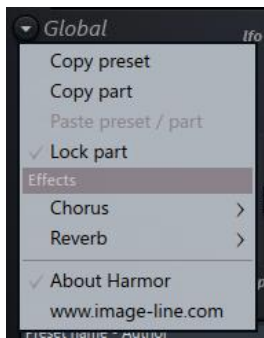
Envelope granularity, *grain*. This knob should specify how smooth envelopes are processed but I cannot see this. Finer granularity means smoother changes to sounds at the cost of higher CPU. I suggest leaving this knob on its default value.

The drop-down menu *computation precision* has three choices:

- Average, has almost no impact on your CPU load. This is the default value and it works fine for sound design.
- High, produce a better result but costs slightly more CPU load.
- Perfect, produces the best result it has the highest CPU load.

I suggest reading the online manual about these options, it is great fun reading!

## Chapter 25, Global menu



With the global menu, it is possible to copy the entire preset into the clipboard. It is also possible to copy only one part. In another Harmor you can then paste either the entire preset or the part.



With the lock part feature, it is possible to avoid changing a part from overwriting it.

You can set chorus and reverb effects from the global menu. These effects are covered in *Chapter 16. Chorus* and *Chapter 18. Reverb - Reverberation*.

The menu item about Harmor brings up the Harmor mascot! She is hanging with her tail in the “o” of Harmor. If you move the window she will swing around a bit. In this window, you can also read the names of the people making Harmor. In this menu, you also find a link to the Image-Line website.

## Chapter 26, Presets

Presets are premade settings. Most plugins in FLStudio can load presets. The default presets come with the standard installation of FLStudio but you could also get presets or buy presets. There are different ways to approach presets based on your own situation. Here below I outline possible phases for how you could think of presets:

1. If you are new to producing music I think that you should not fear using presets at all. On the contrary, play with them, have fun with them, learn how they sounds like!
2. After some time, a time when you produced a couple of songs if you like, you could start adjusting the existing presets to fit your own needs. This is useful technique if you are about to present your music to a “preset aficionado”. Even small differences to a preset might do wonders in creating a unique sound. When doing this, it can be nice to know a couple of tricks for decomposing presets to see how presets are made.
3. As your experimentation proceeds and you produced a few more songs then you should consider making your own presets from scratch that you can use for your own needs. This is the foundation for creating your own sound! It is common to create a library of presets and then when you work on a song you might do changes to a preset and save a new version and so your sound and the presets evolve with you and your progress in music making.
4. When you made presets from scratch a couple of times then you might want to give away or sell presets to others and then it could be a good idea to consider if the presets you created are well explained and that they make efficient use of the CPU.

On a Windows machine the normal location for preset files of Harmor can be found here:

```
\Image-Line\FL Studio  
xx\Data\Patches\Plugin  
presets\Generators\Harmor
```

If you installed FL Studio on the default location, you need to look in Program Files to find the Image-Line folder. Then xx is the main version number.

At this location, you will find a preset file Default.fst. This IS the default preset file that you find in the upper right corner of the presets menu.

When you create your own presets, and save them to the same location they will show up next to the Default preset. If you create a directory and place your own presets in there then they will show up in their own section in the menu with the name of your directory. During an upgrade to a new main version of FLStudio you will need to migrate the presets to the new version.

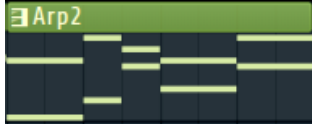
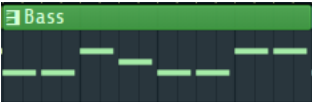
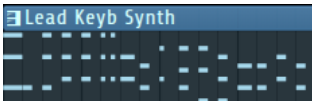

Default	Fairy bells	Rhodes	Slutty space pirate	Short viking
Asp	Gong	Steampunk organ	Soft & warm	Shorty
Chip	Ice bells	Lead	Starlight	Shorty 2
Cubist	Labyrinth	Aural Psynapse	Subspace	Shorty 3
Cubist 2	Light gong	Garner	Sun rays	Shorty 4
Cubist pumpkin	Massive gong	Hard lead	Sunroom	Space dust
Invasion of pumpkins	Melodic bell	Jade	Third fury	Stab fall
Machine wriggle	Possessed hammer	Monolith	Voice of mist	Stratosphere
Overeffected	Shield hit	Morning	Percussion	Trance pluck
Swinging pipes	Strange bell	Robot lead	Forest	Tranquility
The ball	Synthetic bell	Waspinator	Metal thingy	Tunnel bass
Tronic			Moon drum	Viking
Tronic 2			Spicy tuna	Voice synth
Tronic 3				Template
Ulaanbaatar				Bass
Zombie				Loop timestretching (1 bar)
Zombitronic				Loop timestretching (2 bars)
				Loop timestretching (4 bars)
				Plucked
				Resynthesis
				Song synced resynthesis (1 bar)
				Song synced resynthesis (2 bars)
				Song synced resynthesis (4 bars)
				Synthesis
				Tutorials and tricks
				General
				Image synthesis
				Resynthesis
				Randomize
				Hybridize
				Tree display



I created a test project to play with the presets. You can find it here <https://www.malmgren.nl/Harmor/Harmor-54.flp>. It is a naïve simple melody with nothing to it but pretty fun to use when playing around with the presets. But first let me explain the sections of the presets.

### The default preset sections

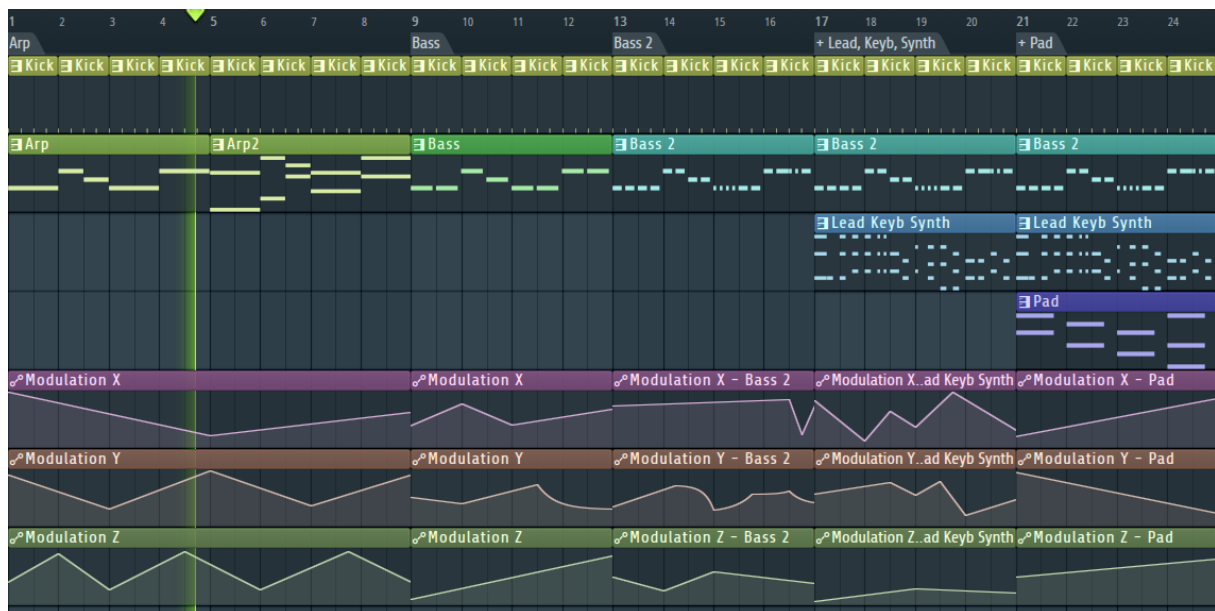
The preset menu has the following default sections:

- **ARP:** Arpeggiators. These presets are unique because they are made to jump between notes in chords as explained in the section Arpeggiator points. I suggest making a simple song over four bars with two or three simultaneous notes (chords) over one whole bar or half a bar. Also in another pattern add a kick drum four times per bar. Now you can more easily try the arpeggiators and especially try to change the X, Y and Z controls while playing. In the experiment project, I programmed the X, Y and Z with automation clips.
- **Bass:** This is all sorts of presets working well in the bass frequencies about 80 Hz up to 252 Hz. Many times, bass sounds will not play well with chords. For experimenting with Bass presets, I suggest making a simple melody with two notes per bar and a kickdrum with four notes per bar. Be aware of that not the same note height / frequency serves all basses equally well. You might need to transpose the notes of your test tune up or down to get the most out of a specific bass preset. A good advice is not to take anything for granted with Bass presets, for example “Fate oh fate” is not a regular bass but a bass drum. Why is it not placed in the percussion section?
- **Bell:** Various kinds of bell sounds. These sounds can be tried as they are.
- **FX:** Effect sounds.
- **Lead:** These presets takes the lead, they are leading the song and as such they are quicker in building up the character of the sound, allows rapid note movement, sometimes with evolving tone or character and a narrower stereo image or mono. I suggest that you try the Lead presets with the almost the same notes as with Keyboard and Synthesizer presets.
- **Keyboard:** These presets are often designed to play chords very well. In this group, you find patches that sounds like regular pianos and organs etc. If you already got experiments with a kick and a bass, then you could try adding the keyboard on top.
- **Synthesizer:** A group of presets in a Synthesizer where the group is called Synthesizer must be quintessential or just hard to categorize otherwise.
- **Pad:** These presets also play chords very well but they are in general designed to support the music, slower in building up the sound, consistent tone, designed for slower

note movement with a wider stereo image. To try these presets, I suggest making a test song with longer notes ending a bit before the end of each bar. When trying these presets, it is useful to move all notes of the Pad preset up or down a couple of octaves to experiment with how well the preset sounds. In the piano roll hold down the ctrl key and press arrow up or down to do that.

- Percussion.
- Sequence: This group contains presets with elaborate sequence programming, almost entire songs can be found here.
- String.
- Template: These presets contain template presets useful as a starting point for creating your own presets. More about these presets in a section later in this chapter.
- Tutorials and tricks. Here below you will find a specific section about these presets.

This is how the test project looks like.



You could select bar 1 until 8 and play this and open the ARP preset and switch between different ARP presets to see how this sounds.

Next you can select the Bass bars from 9 until 12 and press play and switch between different Bass presets.

From here on this project is building up a melody starting with Kick and Bass2 in bar 13 until 16. In the next steps, bar 17 until 20, the Lead Keyb Synth pattern is added on top of the Bass2 pattern. Finally, the Pad pattern is added at bar 21 until 24.

### How to decompose presets

After reading a book like this it much is easier to figure out how things are made. When I created the sample project of the previous section I got interested in how Cubist was made so

we will use Cubist as an example of how to decompose a preset to get a better understanding of how it is made.

To do this load Cubist into a Harmor. I suggest making a simple melody playing long notes. For example, four notes in one bar.



When turning off part B we notice that when Z is set to 0% there is sound and when it is set to 100% there is no sound. Then we turn off A and turn on B and change the Z knob again. At 0% no sound and at 100% there is sound. This correspond nicely to the description that Z is timbre A/B. How is this achieved?

To find out how Z is configured please open the Editor Target menu and notice that there is a triangle at each menu item where the settings is different compared to the default preset. When we list all these modified Editor targets and what parts are used for, then that list looks like this:

- Volume. Parts: Envelope<sup>2</sup> and Modulation Z<sup>1</sup>.
- Filter 1 frequency. Parts: Envelope<sup>2</sup>, LFO<sup>4</sup>, Modulation X<sup>5</sup>
- Filter 1 resonance amount. Part Modulation Y<sup>6</sup>
- Pitch. Part: Envelope<sup>2</sup>
- Global LFO amount. Part: Modulation X<sup>3</sup>
- Filter 1 resonance mask

In the list here above, you can see that I marked parts with a number for what functionality it has in the Cubist preset:

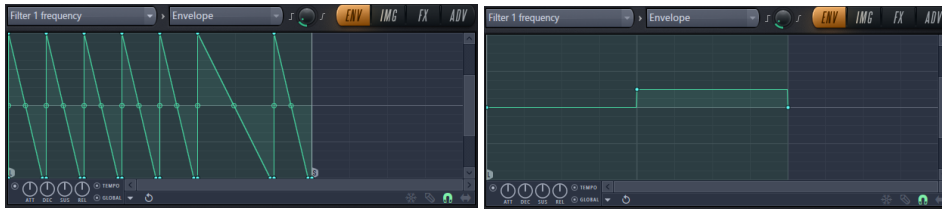
1. Knob Z. When Z is all the way to the right only part A is turned on and when turning it to the right part B is gradually turned on. Here is how that is done. Part A to the left and B to the right.



2. Sequence settings. The sequencing features are configured by volume, filter frequency and pitch. Try turning these off and then turn the knob Z from 0 to 100% to listen to both parts without the sequencing. A to the left. B to the right.



The volume envelopes here above also has arpeggio markers. Part A goes up, up and up. Part B has one marker going down for the rest it is up only. Arpeggio markers makes it useful for playing chord notes so that the patch can jump between notes.



All the frequency envelopes has the articulator output smoothing turned on just a tiny little bit. The filter 1 frequency here above, both A and B, has 30 ms of articulation output smoothing.

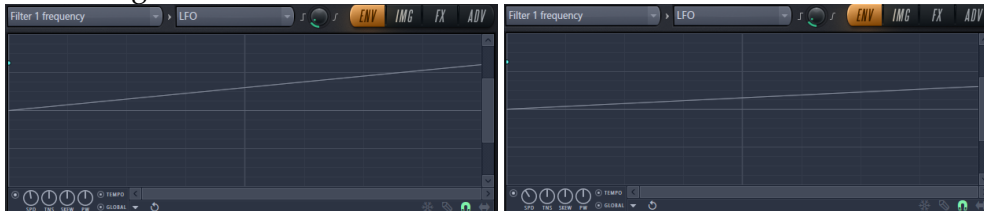


The pitch envelope has articulator smoothing set to 15 ms.

- Global LFO amount. This is controlled by Modulation X. Both A and B has the same graph. The usage of the LFO is explained in the next point.



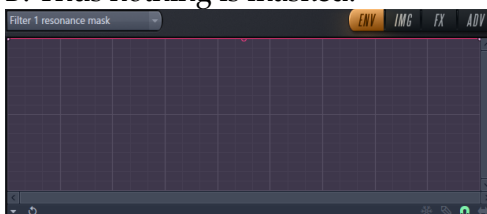
- LFO is also set to control the filter frequency. Part A has phase set to 002:00:000 and part B has phase set to 004:00:000. A is quicker than B. Here below is A to the left and B to the right.



- The Modulation X envelope of the filter frequency is the last part modifying the filter frequency. Part A has a straight line and part B has a curved line.



- Filter 1 resonance is controlled by the Y modulation knob. Normally the default setting displays a curve here but in this case Cubist has a straight line for both A and B. Thus nothing is masked.



The first screenshot here below is marked with orange rectangles displaying where the Default preset is different compared to the Cubist preset. Then below in the images of part A and B of Cubist I marked all knobs different between part A and part B.

Default preset:



Cubist A:



Cubist B:



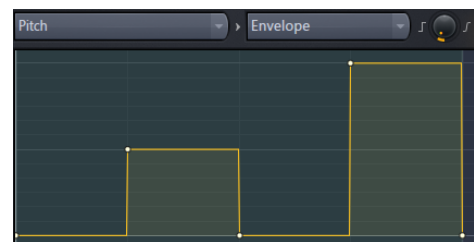


In this table here below, I listed the values of the knobs of the Cubist setting.

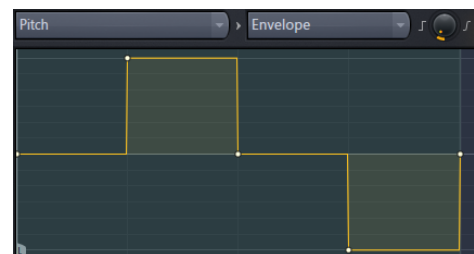
Knob / Dropdown	Default	Part A	Part B
Timbre 1 & 2 mix	Saw waveform	Square waveform	Saw waveform
Phase randomness	0%	100% spectral random	0%
Harmonic protection	-INF dB 0.00	-7.2 dB 0.44	-3.3 dB 0.69
FX dry/mix	0%	23%	0%
Time domain volume	0 dB 1.00	0 dB 1.00	-2.0 dB 0.80
Auto-Gain	0% absolute	100% absolute	100% absolute
Filter 1 resonance type (drop-down)	Lowpass	Classic	Classic
Filter 1 resonance / noise scale (switch)	Oct	Hz	Hz
Filter 1 envelope amount	100%	7%	38%
Filter 1 width	50%	50%	62%
Filter 1 frequency	100%	60%	75%
Filter 1 resonance / noise amount	0%	100%	100%
Filter 1 resonance width / noise length	50%	25%	25%
Filter 1 resonance self-oscillation	0%	43%	0%
Pitch frequency	1.000/1	1.000/4	1.000/2

The most notable difference between A and B is the waveform, saw vs square, and you also had that figured out by now.

The square waveform of Part A is lowered two octaves given by the frequency set to 1.000/4. On top of this Part A has a pitch envelope starting at its “home frequency” and then the pitch frequency envelope is jumping up 1200 cents, and back, which is one octave. Then it jumps two octaves up and back.



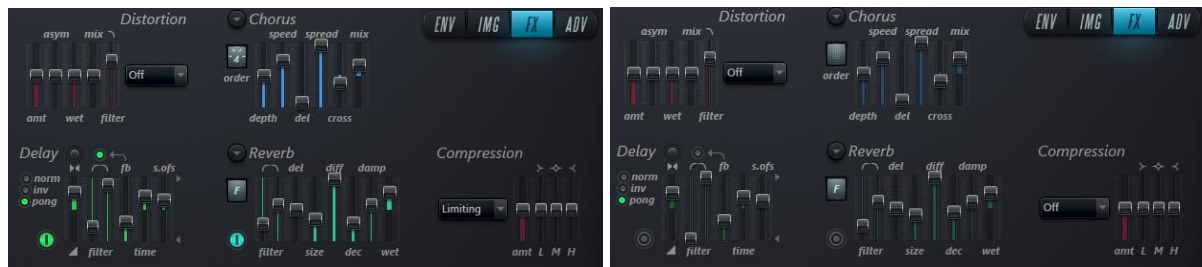
The saw waveform of Part B is lowered one octave due to the frequency set to 1.000/2. The pitch frequency envelope is jumping up and down one octave.



Part A has more harmonic protection compared to part B. Part B has a slightly lower time domain volume making it a little more silent than part A. Since saw waveforms usually comes with more sound energy this is a useful trick to balance the two parts.

Part A has a FX dry / wet mix of 25%. This is so that more of the original sound is mixed into the final sound whereas part B goes entirely through the effects. This brings us to the effects

section. Here below you see the FX section of Cubist to the left and the Default preset to the right.



No distortion or compression. Chorus is of order 4 and for the rest almost entirely the default setting. The same with delay and reverb. Turned on but not changed much. The end result of the effects tab and that part B has more effects is that part B is more present in the echo and the reverb than part A.

There are no changes in the ADV tab except that ramp is changed to 43 ms instead of the default 24 ms.

Now you should have a good understanding of the Cubist preset!

### Template, tutorials, and tricks presets.

The template section of the presets menu contains a number of presets to use as a starting point for your own preset creation.

It is a good exercise to decompose these presets to figure out what the intention is with these.

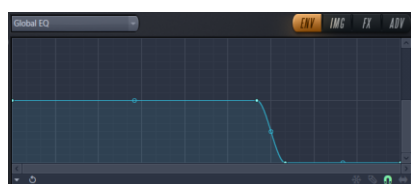
To help you on this let us start with Bass. What is this for? It has this information:

Preset name - Author  
This spares CPU if your preset doesn't have high-frequency content.

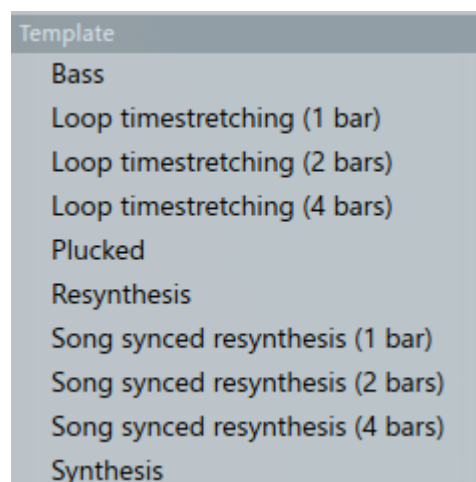
Legit. It sets this frequency:



The Global EQ has this envelope:



In the tutorials preset section you need to read the information window carefully of each preset. There are bits of information in these information windows that you might not find anywhere else. I expect this section to grow in each release of FL Studio.



It had been nice to provide more information about the template, tips, and tricks presets but I feel that I don't know enough to write about these presets in a useful way. Perhaps there is a series of corresponding YouTube movies about them or there are sections of the online help. In later versions of this book I might come back to these presets.

### Create your own presets

If you looked through and listened to some of the presets you will still wonder how these presets are created, at least that is what I wondered many times. Obviously, some of this are secrets of the creator that I don't know so much about. If I knew I would read it here.

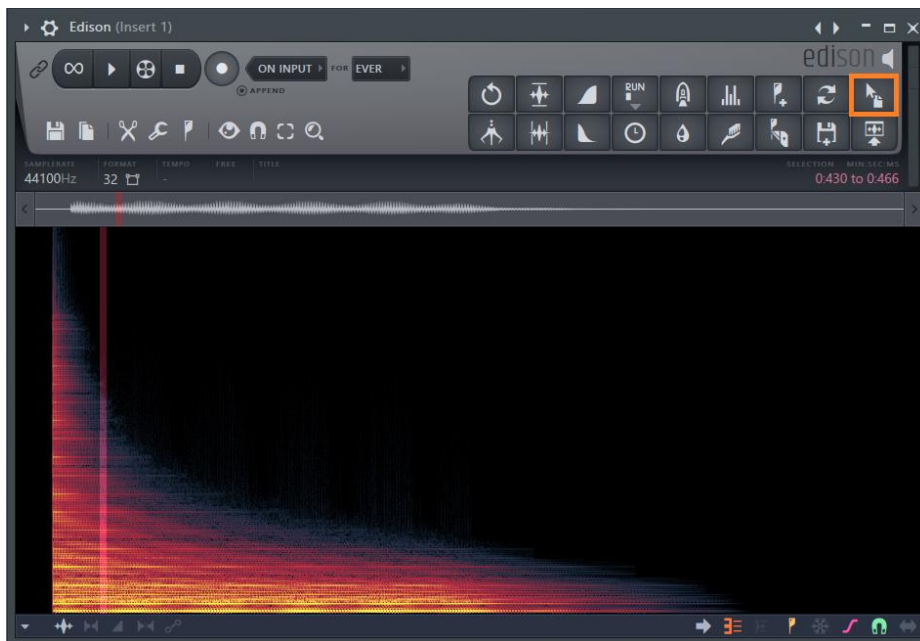
Take for example harmonic levels of timbre 1 in the preset *Analog bass 2*. Even if I could plot a graph like that it would take me days to do it manually! The obvious answer is that more experienced sound designers do not plot these things manually. I believe that many times the sound designer creates a sound and then sampling this and dragging it into a new Harmor and then the sound is modified and changed and inserted again in the next Harmor. A sound can be the result of several generations of manipulations. Many times, the end result is created by a mixture of luck and experience.



To experiment with this preset creation method I created <https://www.malmgren.nl/Harmor/Harmor-55.flp> with two Harmor patches in the channel rack. In "Harmor a" I created the sound that I routed to Edison.



It is a square waveform with blur, prism, pluck, phaser and a pinch of sub protection. To this a little bit of Hill distortion as topping.



Then I recorded a C5 and cut out a tiny little slice and dragged this (with the drag copy / sample / selection tool marked with an orange rectangle) into the Timbre 1 window of Harmor b.

The result is a graph that it would take days to plot out manually!





This serves just as an example. For further refinement, I suggest analyzing Analog Bass 1 and that is what we will do in the next section.

### Create presets for others

The manual from Image-Line emphasizes that if you plan to distribute presets to others then you should try to make these presets as CPU efficient and well explained as possible. Here in this section we will explore these objectives and what you can do with this.

When you create presets yourself that you are using yourself then if you do it in a CPU intense way then you yourself are the one to suffer. When you create presets for others, often less experienced sound designers, and if you do it in a CPU intense way then perhaps they will not be able to make good use of Harmor. Probably you will not be blamed but Image-Line will and they don't like that.

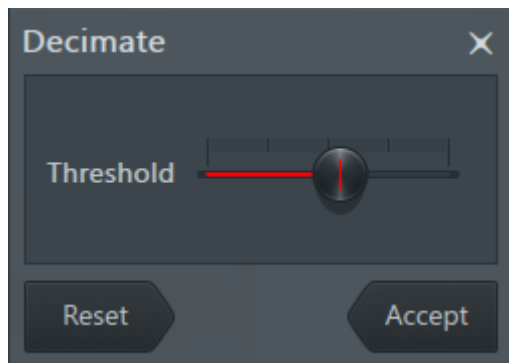
In the previous section, we created sounds with great complexity with little effort but is it necessary? To explore this idea, I will use the preset *Analog Bass 1* as an example, not as a critique to the maker but as an example. Actually, *Analog Bass 1* is the reason that I bought Harmor myself. I loved it so much I had to have this synth. The idea here is to decimate points and disable features to see at what point the gist of the sound starts to deteriorate. Load *Analog Bass 1* in a channel of Harmor. Then we will simplify it in a series of steps. If you like you can try it yourself or you can load an example project that I made for this at: <https://www.malmgren.nl/Harmor/Harmor-56.flp>. In this project the first Harmor in the channel rack is the untouched original *Analog Bass 1* preset. The next Harmor is called *Modified #1* and this is the version I explain about here below.

We start looking at the original *Analog Bass 1* preset. Here is the original timbre 1 of *Analog Bass 1* in part B.



This is probably a couple of thousand control points. In the lower left corner select the Decimate points function.

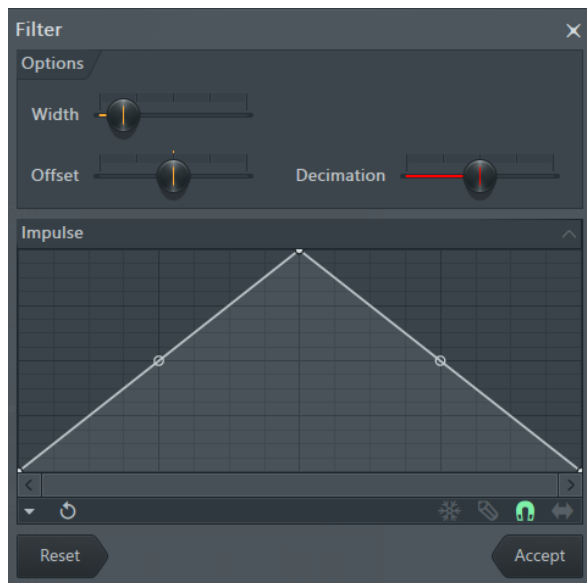




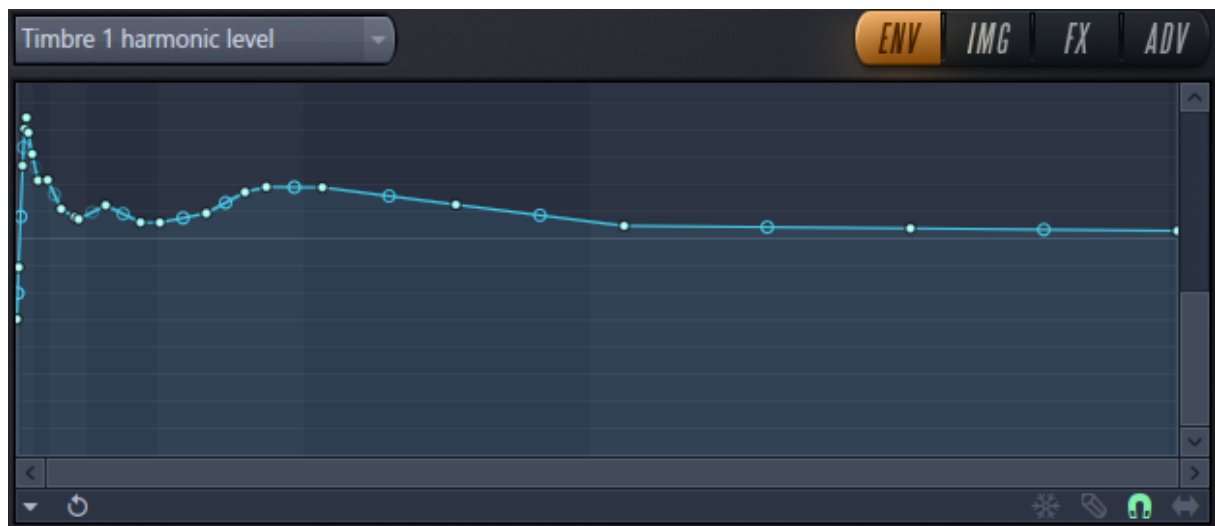
12 points. Much simpler. It still sounds very similar though. Now we switch to part A and Timbre 1.



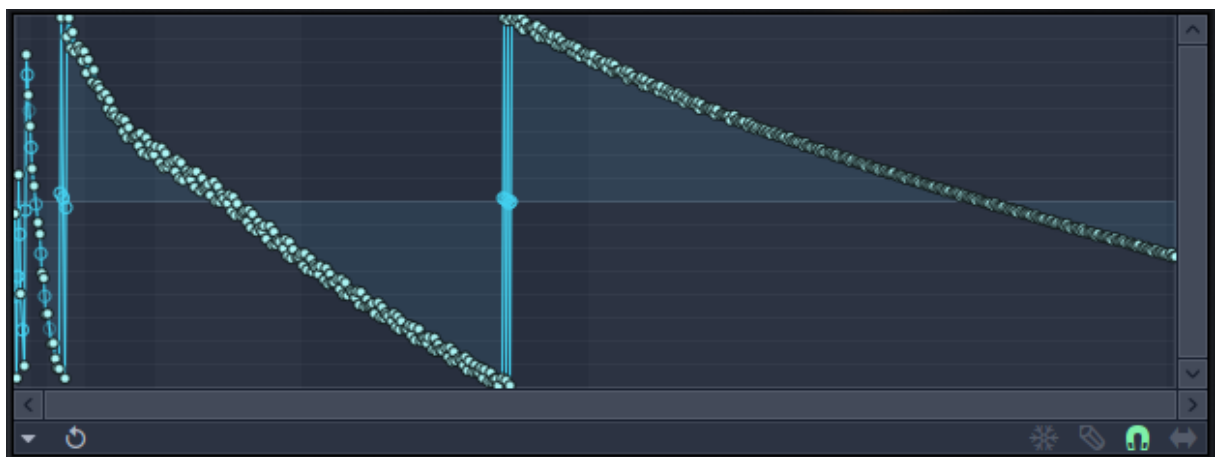
Again, a couple of thousand control points. This time you select Filter.



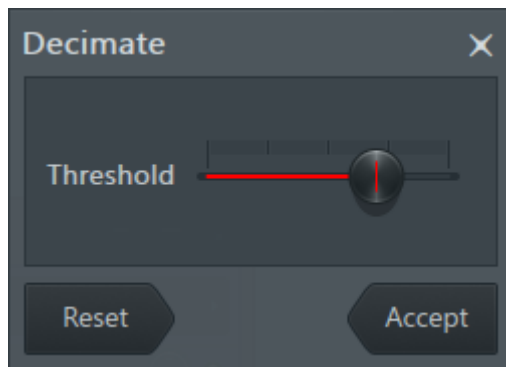
That will give the following Timbre 1 graph:



It still sounds very similar to the original! Now we turn to the harmonic phase window. In the editor target drop down select Harmonic phase.



Now we will Decimate points also in this graph.



Giving this result:



At first I tried using the Filter functionality but this resulted in a “ringy” tone being added but that could be avoided with Decimate.

It still sound similar to the original sound! It is amazing that we removed a couple of thousand points and it still sounds very similar to the original. Is this giving any result on the CPU? No, not much. We talk about CPU later. What this is about is getting to the gist of the sound. Where is the gist?

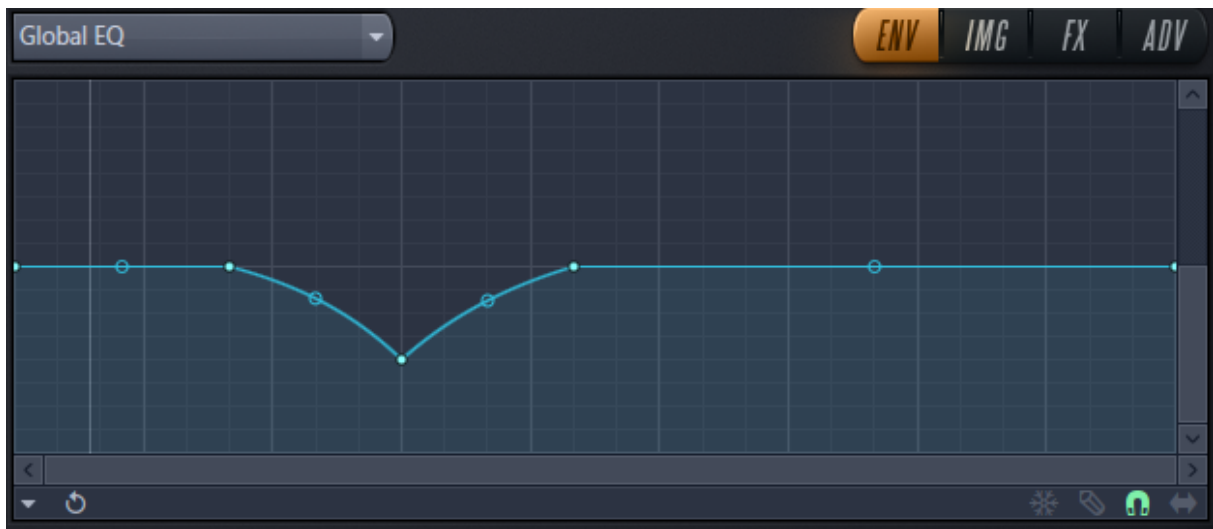
If we switch to the FX tab and turn off distortion, reverb, and compression. Are these effects providing the essence of the sound? Distortion gives the sound the feeling it has been routed through slightly distorted bass amplifier cabinet. The reverb gives the sound the room feeling. Compression provides a slightly more bass to the sound but overall the effects will not provide the gist of the sound, so we turn on the effects on again.

Part A and part B play different roles in this preset. B plays a higher pitch. If we turn that off, then we can concentrate on the bass-part of the preset. Open the editor target menu to see what else are used in this preset. Here below (A=B) indicates that both A and B are used and they are identical. (A<>B) indicates they are both used and different compared to each other. (B) indicates that only B is used.

- Filter 1 frequency (A=B). Parts: Envelope, Velocity mapping, Modulation X.
- Pluck amount (A=B). Parts: Modulation Y.

- Local EQ (B).
- Harmonic phase (A<>B).
- Timbre 1 harmonic level (A<>B).
- Filter shape (A=B).
- Global EQ (A<>B)

Somewhere the important ingredient can be found. For example the Global EQ of part A looks like this:

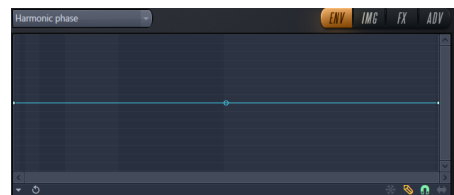


This frequency dip makes a significant contribution to the bass sound. If you grab the lower point and move it around and play notes you will notice that this is an important factor of how this preset sounds like and that you can make different versions just by moving this point.

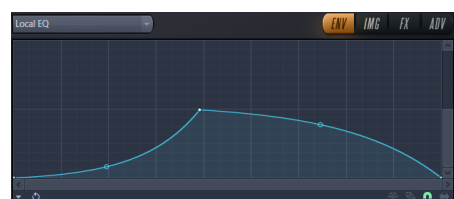
Now we will talk about part B for a moment. Surely the harmonic phase of part B is impressive. To the right, you see an image of the original harmonic phase. In this graph, you see a regular pattern that would take much time to create by hand.



Suppose we destroy this harmonic phase completely and replace it with a line like the graph displayed here to the right. Before we do clone Will this have a negative impact of the sound? To try this I turned off part A and played only part B in the original preset and compared this with the simplified version. In the test project this is "Modified #1" and "Modified #2". I noticed very little difference. Why is that?



This is because the purpose of part B is to serve the higher frequencies and when pluck is engaged you only hear high frequencies during the moment when the attack of the note is played. Part B also got a local EQ



following the fundamental tone cutting out the low frequencies and a global EQ cutting out a bit but that is not all.

The filter has a custom shape configured like kind of low pass filter and the filter frequency is also configured to “pluck”. The harmonic phase is impressive but it is plucked, filtered, and filtered again. There is simply very little left. On top of this the A/B mix slider is giving A more sound than B.



You could skip the filter frequency drop but keep the pluck. You could skip the global EQ but keep the local EQ. When doing these changes clearly the characteristics will change but not so much.

There is one more thing with part B worth discussing and that is Unison. Are we starting up the Unison engine just for the pluck effect? If we need a more CPU efficient bass, then we removing Unison could help. Another thing to consider is the compression in the FX tab. This preset has the compression set to “Heating” to boost the bass slightly. In many situations, you use compression in music and if you are going to compress anyway then compression inside Harmor could be avoided to save CPU. If the only reason is to boost the bass a bit, then you can go to A and boost some of the lower frequencies instead without starting up the entire compression engine within Harmor.

There are tricks to lowering CPU usage, although most of the CPU-saving settings are on by default.

Consider the following:

- If you use just a part of an image or a sampled sound, consider cropping it and include only that part. That makes the preset file smaller.
- Harmonic blur is CPU intense.
- If you use resynthesis you do not need cubic interpolation.
- Using only the number of partials that you need for the sound and nothing more.
- Only activate a feature if it is really audible.



## Chapter 27, Examples

### Make white noise

White noise is a hissing sound to be found almost everywhere in modern music. It is used for sound effects, build ups, fade ins, sweep in, sweep out. In Harmor there is no ready-made white noise preset so we will make it ourselves in this exercise and in this exercise, make a good use of the randomization of phase (to the left).

1. Load the default preset.



2. Set partial pitch distribution to “oct” and frequency to 1.0000/9 and detune to 1.0000/6. At this point it sound like some sort futuristic laser in an old movie.



3. Then activate Unison order 3. No panning and Unison pitch thickness and phase set to maximum. Set blurred distribution. At this point we can start to hear the white noise effect.
4. Now we want the white noise centered around note C4. To do that we need to modulate the pitch. Set the upper point to +4 octaves. The lower point at -5 octaves at the last vertical line B8.



This way we centered the pitch around note C4 so no matter what key we are pressing the result will be only a note C4.

5. Set the rand of phase to -100%.



6. To compensate for the built-in Brownian filter, we increase the harmonic levels at higher frequencies. We start at -24.08dB and end at +15.92 dB.



7. Switch to the ADV tab and set ramp to 0.



Ramp is by default set to 25ms partial relative. This delays the engagement of the partials. The different partials are started from bottom and up. First the fundamental frequency is started and one by one the partials are started and the last partial is started 25 milliseconds later. When we change this to 0 the attack is more prominent. In most of the examples that I have created so far where we investigated the phase I set the ramp to zero so that it would be more apparent what is going on.

And there you have it, white noise produced with Harmor!

## Chapter 28, Open questions

In this chapter I collected questions raised in the book. I contacted Image-Line about these questions and when I have an answer I will explain the features and update the book.

Page 28. The default phase value is 180. Why is phase jumping to 0 when importing a regular wav audio file? (When I import a wav of a single cycle waveform created in Sytrus then phase is not jumping.)

Page 41. In the section *Unison alternate distribution*, I found out that Uniform and Random are identical. Why is that? Is this a known bug?

Page 57 Blur attack and delay, the knob *time blur decay*. What is this knob doing?

Same section another question: Time blur attack and decay knobs, what is 100% representing?

Page 60. Pluck replaces blur decay but what is it doing? I cannot figure out how this works.

Page 132. Image sharpening, the knob *sharp*, loses its setting when playback reaches the end of the window. How can this be avoided? Is this a known bug?

Page 140. Create a bouncing loop time envelope. What is the difference between this function and Prepare image time envelope?

Page 146. Filter envelope amount. Why is the editor target not bipolar?

Page 163. Auto-gain mode in the ADV tab. I don't know how this works. Any hint to explain the working of this feature is welcome.

Page 164. Randomness. I contemplated about the explanation in the manual and to me it does not make any sense. Instead I figured out how it could work but am I closer or completely off?

Page 164. Performance versus quality, knob *safety*. What is the knob doing and are there any suggestions for how I can see that it works?

► Global LFO amount  
► Global LFO speed

Why are modified Editor targets sometimes giving a full triangle and sometimes an empty triangle?