



---

# HARMOR: 加减合成器

---

Jens Malmgren (瑞典)

(2017)



翻译: 王付豪 (Rich)

翻译时间: 2020 年 11 月

# Contents

译者的话.....	8
前言.....	10
第一章: Harmor 介绍.....	13
合成器一览 (The synths landscape) .....	13
音量 (Volume) .....	13
Harmor 音色 (Harmor timbres) .....	19
音色窗口 1 的谐波水平 (Timbre 1 harmonic level) .....	21
谐波相位 (Harmonic phase) .....	21
音色 2 谐波水平 (Timbre 2 harmonic level) .....	22
音色 1 和 2 的混合 (Timbre 1 & 2 mix) .....	24
让-巴蒂斯特·约瑟夫·傅里叶 (Jean-Baptiste Joseph Fourier) .....	24
音色混合模式 (Timbre blending modes) .....	25
相位随机 (Phase randomness) .....	27
第二章 加法/减法合成 (Additive/subtractive synthesis) .....	30
音调频率 (Pitch frequency) .....	32
声调 (Pitch) .....	33
左右声道 (Panning) .....	34
谐波失谐乘数 (Harmonic detuning multiplier) .....	34
Oct 模式 VS Hz 模式.....	35
第三章 齐奏 (Unison) .....	36
齐奏左右声道 (Unison Panning) .....	36
齐奏声调厚度 (Unison pitch thickness) .....	37
齐奏相位 (Unison phase) .....	37
齐奏的分布 (Unison distribution) .....	40
齐奏交替的分布 (Unison alternate distribution) .....	41
第四章亚音, 剪切, 音量, 增益等 (Subs, clipping, volumes, gain etc) .....	42
亚谐波 1,2 和 3 (Sub harmonics 1, 2 and 3) .....	42
谐波保护 (Harmonic protection) .....	43
谐波剪切阈值 (Harmonic clipping threshold) .....	43
谐波剪切模式 (Harmonic clipping mode) .....	43
干湿混响效果 (Effects dry/wet mix) .....	44
自动增益 (Auto-gain) .....	44
时间域音量, 频率域音量 (Time domain volume, frequency domain volume) .....	45
速度开关 (Velocity switches) .....	45
第五章 . 基础滤波器 (Basic filtering) .....	47
滤波器 1&2 混合 (并联-串联) Filter 1 & 2 mix (parallel – serial).....	47
滤波形状 (Filter shapes) .....	48
滤波器频率 (Filter frequency) .....	49
键盘跟踪, 从琴键到滤波器 1 频率 (Keyboard tracking, Key to filter 1 frequency) ...	49
过滤器宽度 (Filter width) .....	50
过滤器共振量 (Filter resonance amount) .....	52
过滤器共振宽度 (Filter resonance width) .....	53

过滤器共振类型 (Filter resonance type) .....	54
过滤器共振适应宽度/自由噪音 (Filter resonance adaptive width/free noise) .....	55
过滤器共振偏移 (Filter resonance offset) .....	56
过滤器共振自振荡 (Filter resonance self-oscillation) .....	57
过滤器 oct/Hz 开关 (Filter oct/HZ switches) .....	57
第六章 . 模糊 (Blur) .....	57
谐波模糊量 (Harmonic blur amount) .....	57
顶部和底部张力 (Top and bottom tension) .....	58
模糊冲击以及衰减 (Blur attack and delay) .....	58
第七章 . 弹拨效果和移相器 (Pluck and Phaser) .....	59
弹拨效果 (Pluck) .....	59
弹拨效果量 (Pluck amount) .....	60
弹拨效果代替模糊衰减 (Pluck replaces blur decay) .....	60
音色混合模式: 弹拨效果 (Timbre blending mode: Pluck) .....	61
移相器 (Phaser) .....	62
移相器种类 (Phaser type) .....	63
移相器混合 (Phaser mix) .....	64
移相器宽度 (Phaser width) .....	64
移相器偏移 (Phaser offset) .....	64
移相器偏移速度 (Phaser offset motion speed) .....	65
从琴键到相位器偏移 (Key to phaser offset) .....	66
移相器比例 (Phaser scale) .....	67
第八章棱镜 (Prism) .....	67
棱镜效果大小 (Prism amount) .....	67
棱镜相加/相乘模式 (Prism add/multiply mode) .....	68
谐波水平到棱镜效果大小 (Harmonic level to prism amount) .....	69
第九章 . 谐音器 (Harmonizer) .....	71
谐音器混合 (Harmonizer mix) .....	71
谐音器宽度 (Harmonizer width) .....	71
谐音器的 shift 和 step 参数 (Harmonizer Shift Step) .....	72
谐音器的 gap 和 step 参数 (Harmonizer Gap Step) .....	72
第十章 . 颤音 (Tremolo) .....	74
颤音速度 (Tremolo speed) .....	75
颤音间隔 (Tremolo gap) .....	75
颤音深度 (Tremolo depth) .....	76
第十一章 . 颤动 (Vibrato) .....	76
Vibrato 音调深度 (Pitch vibrato depth) .....	76
Vibrato 音调速度 (Pitch vibrato speed) .....	76
第十二章 Legato 和 Portamento.....	76
Legato .....	77
Legato 模式开关.....	77
Legato 时间.....	78
线性/对数滑音 (Linear/Logarithmic legato) .....	78
固定/可变滑音时长 (Fixed/variable legato time) .....	78

滑音限制 (Legato limit) .....	79
滑音音效的释放时间/滑音速度 (Velocity / Release to legato time) .....	79
延音 (Portamento) .....	80
延音模式开关 (Portamento mode switch) .....	80
延音时长 (Portamento time) .....	80
线性/对数延音 (Linear/Logarithmic portamento) .....	81
固定/可变滑音时长 (Fixed/Variable legato time) .....	81
延音音调限制 (Portamento pitch limit) .....	81
延音的起始速度/释放速度 (Velocity / release to portamento time) .....	82
第十三章 弹拨 (Strum) .....	83
弹拨时间 (Strum time) .....	83
弹拨模式 (Strum mode) .....	83
弹拨张力 (Strum tension) .....	84
第 14 章 . A/B 部分 .....	85
A/B 选择 (Part selector) .....	85
A/B 部分激活开关 P (Part enable switches) .....	85
A/B 部分混合 .....	85
瞬间复制用户的设定更改 (Instant replication of the user's changes) .....	85
第十五章 失真 (Distortion) .....	86
Waveshaping 理论知识 .....	86
Waveshaper 中的单极以及双极模式 .....	87
WaveShaper 中的 Pre amp .....	87
WaveShaper 中的混音水平 (Mix level) .....	87
WaveShaper 中的 post gain .....	87
失真类型 (Distortion type) .....	87
失真度 (Distortion amount) .....	88
失真不对称 (Distortion asymmetry/extra) .....	88
失真湿声音量 (Distortion wet volume) .....	88
失真混合 (Distortion mix) .....	88
失真高切 (Distortion high cut) .....	88
失真效果示范工程 .....	89
Harmor Classic Distortion .....	90
Harmor Hill Distortion .....	90
Harmor Flat Hat Distortion .....	91
Harmor Cube Distortion .....	91
Harmor Log Distortion .....	91
Harmor Softsat Distortion .....	92
Harmor Rubber Distortion .....	92
Harmor Ribbon Distortion .....	93
Harmor Sincrush Distortion .....	93
Harmor Bitcrush Distortion .....	93
Harmor Bitjam Distortion .....	94
第 16 章 . 合唱 (Chorus) .....	94
Fruity Flanger .....	95

Fruity Flangus.....	97
Harmor Chorus.....	99
合唱预设 (Chorus presets) .....	100
第 17 章 . 延迟 (Delay) .....	100
开启延迟效果 (Enable delay) .....	101
延迟输入音量以及延迟左右声道输入均衡.....	101
延迟反馈模式: (Normal, Inverted) 以及 (Ping Pong) .....	101
延迟反馈水平 (Delay feedback level) .....	101
延迟时间 (Delay time) .....	102
低切, 高切, 反馈以及反馈衰减.....	103
延迟时间立体声偏移 (Delay time stereo offset) .....	104
第 18 章 . 混响 (Reverb) .....	105
混响颜色 (Reverb color) .....	105
混响输入滤波器 (Reverb input filter): 高切以及低切.....	106
混响预延迟 (Reverb pre-delay) .....	107
混响空间大小 (Reverb room size) .....	107
混响衰减 (Reverb decay) .....	107
混响高衰减 (Reverb high damping) .....	107
混响扩散 (Reverb diffusion) .....	108
混响湿声音量 (Reverb wet volume) .....	108
混响选项 (Reverb options) .....	109
第 19 章 . 压缩 (Compression) .....	109
压缩类型 (Compression type) .....	110
压缩量 (Compression amount) .....	110
压缩带 (Compression band): 低, 中, 以及高.....	110
第 20 章 . 曲线编辑器 (Line editor) .....	110
点模式 (Point mode) .....	111
曲线的基本操作 (Basic line operations) .....	111
线段类型 (Line segment types) .....	112
琵琶音点 (Arpeggiator points) .....	112
第 21 章 . 目标的基本知识 (Target basics) .....	112
Editor target 下拉菜单.....	113
合成模式: 单极乘积 (Combining mode: Unipolar product) .....	114
合成模式: 双极乘积 (Combination mode: Bipolar product) .....	116
合成模式: 双极加和 (Combination mode: Bipolar sum) .....	116
发生器部分 (Articulator part): 包络线 (Envelope) .....	116
Envelope 窗口选项菜单: Make all loop.....	117
Envelope 窗口选项菜单: Insert points at markers.....	117
Envelope 窗口选项菜单: Flip vertically.....	117
Envelope 窗口选项菜单: Scale levels.....	117
Envelope 窗口选项菜单: Normalize levels.....	118
Envelope 窗口选项菜单: Decimate points.....	118
Envelope 窗口选项菜单: Filter.....	118
Envelope 窗口选项菜单: Smooth up.....	119

Envelope 窗口选项菜单: Smooth up abrupt changes .....	120
Envelope 窗口选项菜单: Turn all points smooth .....	120
Envelope 窗口选项菜单: Create sequence .....	121
Articulator part: 低频振荡器 (LFO) .....	124
Articulator 部分: Keyboard mapping .....	125
Articulator 部分: Velocity mapping.....	125
Articulator 部分: Modulation X, Y and Z mapping.....	125
Articulator 部分: Random mapping.....	125
Articulator 部分: Unison index mapping.....	126
Articulator 部分: Held index mapping .....	126
Articulator 部分: Mod X speed mapping.....	126
Articulator 输出平滑化 (Output smoothing) .....	126
第 22 章 . 图像合成 (Image synthesis) / 再合成 (Resynthesis) .....	127
查看模式 (View mode): plane(s) to work on.....	128
图像微速度 (Image fine speed) .....	129
图像过程速度 (Image course speed) .....	129
图像时间偏移 (Image time offset) .....	129
图像时间偏移平滑化 (Image time offset smoothing) .....	130
循环模式 (Looping mode) .....	130
图像改善 (Image sharpening) .....	130
Level 分区 .....	131
图像增益像素比例 (Image gain pixel scale) .....	131
图像增益混合 (Image gain mix) .....	131
频率分区 (Frequency section) .....	131
图像频率像素比例 (Image frequency pixel scale) .....	132
图像频率模式 (Image frequency mode) .....	132
图像共振峰漂移 (Image formant shift) .....	133
图像共振峰漂移混合量 (Image formant shift mix) .....	134
图像复制 (Image copy), 图像粘贴 (Image paste) .....	134
图像频率补充曲线以及图像增益补充曲线 (Image frequency interpolation curve and Image gain interpolation curve) .....	134
显示比例 (Display scale) .....	136
ADV image / resynthesis 设置: 质量模式 (Quality mode) .....	136
ADV image / resynthesis 设置: 去噪声 (Denoise) .....	137
ADV image / resynthesis 设置: Side processing / Invert right channel - stereo.....	137
图像选项菜单: 基本知识 (Basics) .....	138
图片选项菜单: 将地图音频标记/地区映射到键盘 .....	138
图像选项菜单: 准备鼓组的时间包络线 (Prepare time envelope for drum loop) ....	139
图像选项菜单: 准备图像时间包络线 (Prepare image time envelope) .....	140
图像选项菜单: 创建有弹性的循环时间包络线 (Create bouncing loop time envelope) .....	141
图像选项菜单: 在循环区域中慢下来 (Slow down within loop region) .....	141
图像选项菜单: 时间偏移的音调 (Tune pitch for time offset) .....	141
图像选项菜单: 将看到的区域设置成循环区域 (Set viewed zone as loop region) ..	142

图像选项菜单：将上一个演奏的区域设置成循环区域 (Set last played zone as loop region)	143
图像选项菜单：将时间映射到刚刚按下的键 (Map time offset to last hit key)	143
图像选项菜单：创建随机云 (Generate random cloud)	144
图像选项菜单：横向降解 (Degrade horizontally)	144
图像选项菜单：纵向降解 (Degrade vertically)	144
图像选项菜单：交叉渐变 (Crossfade) 10%, 25%和 50%	144
图像选项菜单：剪切顶部 (Crop top)	145
图像选项菜单：反转 (Invert)	145
图像选项菜单：垂直反转 (Flip vertically)	145
剪切板监视 (Clipboard monitoring)	146
编辑图片 (Edit image)	146
图片选择列 (Image selected column)	146
Image freq local mask	146
第 23 章 . 高级目标 (Advanced targets)	147
过滤包络线量 (Filter envelope amount)	147
全局包络线 ADSR 设定 (Global envelope ADSR settings)	148
全局包络线 LFO 设定 (Global envelope LFO settings)	149
全局 Global EQ	149
局部 Local EQ	150
和声随机 (Harmonic randomness)	151
和声剪切 (Harmonic clipping)	152
和声棱镜 (Harmonic prism)	152
和声合唱音调 (Harmonic unison pitch)	153
图像共振峰突变 (Image formant mutation)	154
滤波器形状 (Filter shape)	155
共振形状 (Resonance shape)	155
弹拨音形状 (Pluck shape)	155
移相器形状 (Phaser shape)	156
谐波器分布 (Harmonizer distribution)	156
滤波器面罩 (Filter mask)	157
滤波器共振面罩 (Filter resonance mask)	159
移相器面罩 (Phaser mask)	159
谐波器局部面罩 (Harmonizer local mask)	160
模糊局部面罩 (Blur local mask)	160
第 24 章 . ADV 标签	161
预延迟 (Predelay)	161
包络线释放复音比例 (Envelope release polyphonic scale)	161
斜坡时间 (Ramping time)	162
杂音消除 (Declick)	163
单位顺序 (Unit Order)	164
效果器顺序 (Effects order)	165
自动增益模式 (Auto-gain mode)	166
倒转琵琶音 (Invert arp)	166

Smooth mod.....	166
Monophonic.....	166
随机 (Randomness) .....	167
表现 (Performance) VS 质量 (Quality) .....	167
第 25 章 . 全局菜单.....	168
第 26 章 . 预设.....	169
如何分解预设。 .....	171
模板, 教程, 以及窍门预设。 .....	176
创建你自己的预设.....	177
为他人制作预设.....	179
第 27 章 . 例子 .....	184
制造白噪声.....	184
第 28 章 . 开放式问题.....	186

# 译者的话

作为一个刚刚接触电音制作的小白，在斥巨资买下了 FLStudio 的正版之后，实在没有资金去继续购买功能更强大，教程更加完整的正版 Serum 或者 Sylenth 合成器插件，所以无奈只好仔细研究 FLStudio 自带的免费合成器 Harmor，但是好在这个合成器的功能对于我这样一个新手来说完全足够了。

然而我立即就遇到了一个十分巨大的问题。在中文世界里，对于 Harmor 功能的介绍好像也就只有那本仅仅 42 页的《Harmor 中文手册》。对于 Harmor 这样复杂的合成器来说，42 页仅仅只对 Harmor 进行一个十分粗略的介绍，而且对于其中的许多用语，我个人完全不知所云。当然，理论上讲，我们不可能将各种音效带给耳朵的感觉“写”进书里，这也就导致了书籍这种承载形式上的缺陷。所以我又尝试在网络上寻找 Harmor 相关的视频教程。

但是情况依然十分类似，在知名的学习视频网站，Bilibili 中，除了 up 主 Fanki 的那个 1 小时的教程，剩余的就只是几个零零散散，5、6 分钟左右的 Youtube 搬运视频。整体而言，中文世界缺少对于 Harmor 这个合成器插件，一个全面的，系统的讲解教程。因此我不得不将注意力转向外网。

幸运的是，我找到了来自瑞典的 Jens Malmgren 先生写的这本《Harmor: 加减合成器手册》。在读完了前言之后，我就已经决定了，尽管本人英语水平一般，对电音制作的了解更是处在入门的阶段，我都要尽力将这本书翻译成中文。第一是中国有越来越多的年轻人开始爱上电音，也有越来越多的年轻人喜欢上电音制作，其中不乏有许多高中生以及大学生。他们中间一定有许多像我一样，买不起昂贵的合成器插件的人。如果由于没有好的教程，而导致他们放弃了对于电音制作的梦想，那是多么遗憾啊！第二点是 Malmgren 先生坚持将本书免费开放的精神完全感动了我，我想参与到传承这种精神的行动中来。因此决定利用在大学毕业后，等待毕业证书下发的这段时间，无偿翻译并分享我的翻译作品。

这本书其中的翻译必然是存在许多瑕疵的，有些翻译的原则也没办法坚持贯彻下去，比如我一直想将所有的英文都翻译成汉字，但是考虑到大部分电音爱好者使用的是英文版的 FLStudio，所以对于一些专业词语，不翻译反而会帮助英语不好的同学找到某个旋钮或者设置的所在位置，然而如果不翻译这些词语，则会造成理解上的困难。因此这也许会让许多强迫症患者，比如我，感到十分难受。但是我译这本书的宗旨是希望读者能以最方便，最快速的方式理解作者想要表达的内容。因此有些地方不得不作出妥协。

另外，对于电音学习者来说，掌握一定的英语是必要的，这种中文夹带着英文的翻译，会帮助你熟悉 FLStudio 中的界面。这样的话，阅读这本书也可以成为你学习电音相关英文的一个十分不错的机会。

如果这本翻译作品不但没有帮助你更好的使用 Harmor，反而让你感到更加困惑的话，那么我将感到十分抱歉，希望你可以通过阅读英文原文来帮助你理解。

在逛 Malmgren 先生的个人网站的时候，我发现他的网站标题中有一句话，或许是他的座右铭吧：I create, that is my hobby. 我希望使用一个在句型结构上相似的一句话来结束这部分：I share, that is my hobby.

# 前言

我原本希望有人可以为我写这本书，可惜没有人。所以我来了，我来自己创作出这本书。写这本书很难，而且非常耗费时间。我本身就是一名开发者，因此我完全理解，当 Image-Line 公司的开发人员必须写一份手册时沮丧的心情。那感觉就像是在浪费时间。也许真的就是在浪费时间。其实我更希望那些开发人员可以制造出更多新的特色，而不是在写手册这方面浪费时间。所以现在我就是用自己的空闲时间替他们做这件事情。

有一天我突然明白了，并不是对一事物无所不知的人才应该去写关于这个东西的书，而是那些能够与初学者感同身受的人才应该写这本书。嗯，那就是我了。现在当我开始写这本书的时候，我对 Harmor 其实一无所知，因此我决定我会与读者们一起共同学习 Harmor。

我讲述的方式是先宏观。首先我们会先整体过一遍 Harmor，跳过大部分的下拉菜单和包络线。在第二遍的时候，我们再打开所有的下拉菜单，仔细探讨其中的各个选项。

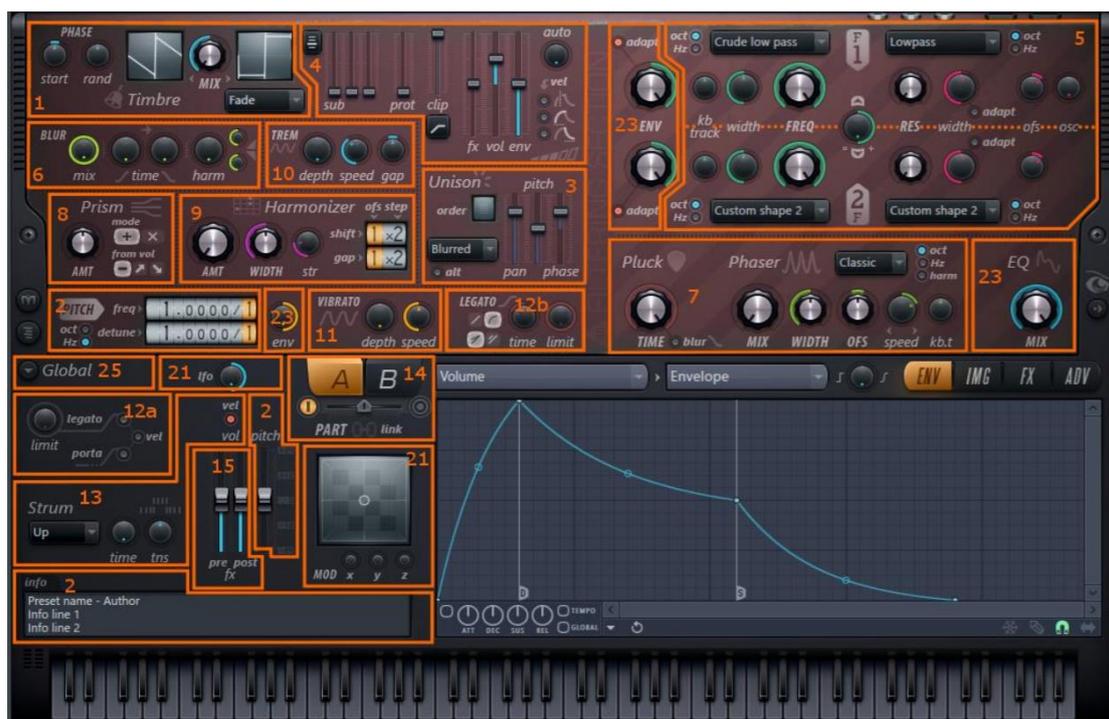
……7 个月过去了……

有人建议我去卖这本书，但是我觉得他们不太理解为什么我写这本书，在此我来进行一番解释。我写这本书是因为我十分喜欢 Harmor 这个插件，以及 Harmor 的工作原理，我也乐意将我所积累的知识传播出去。售卖书籍跟音色设计没有任何关系。另外，如果我要去卖这本书，也许我得去做市场，而且价格本身会成为人们阅读这本书的障碍。我不需要这些东西，况且我在全世界发展最快的翻译公司中有一份软件开发的工作，所以我在经济方面没有什么问题。

我向他们这样解释之后他们依然不肯放弃说服我，他们告诉我，我应该去找 Image-Line 公司要这本书的稿费，但是我想他们可能不太清楚我和 Image-Line 公司之间的关系，所以请让我解释一下。我仅仅只是购买了 Image-Line 公司的产品，除此之外我和 Image-Line 公司没有任何联系。虽然我不想引起这家公司的不爽，但是我希望做我自己，因此当我觉得有哪些功能不太好的时候，我会写下来。如果 Image-Line 公司付给我钱的话，事情就会变得复杂起来，而我看来，这种复杂本身毫无意义。

如果你想因为这本书而给我钱的话，那我建议你把这笔钱捐给你们国家的风湿病慈善机构，然后发 twitter，加上 #JensHarmorBook。那样我会很高兴的，因为我自己有一点风湿病。如果你想打印这本书，那么只能为你自己学习所用，请勿用于商业。你不能将这本书，或任何章节以任何电子版的形式发行。这本书完全由版权法案所保护。

书的下载地址：<https://www.malmgren.nl/Harmor.pdf>



如果你阅读的是这本书的 PDF 版，那么你可以点击这张图中的内容，就可以直接跳转到相应的讲解部分。



# 第一章：Harmor 介绍

这本书包讲解的是如何在 FL studio 中使用 Harmor 来制造音色。书中使用的 FL studio 的版本是 12.3 以及以上。

什么是 Harmor? Harmor 是 Image-Line 公司的旗舰合成器插件，可以在 FL studio 中使用，也可以以 VST 格式安装在其他数字音频工作站 (DAW – Digital Audio Workstation)，Harmor 的发布时间是 2011 年 9 月。

我会从 Harmor 使用界面的左上角开始讲起。Harmor 制造出来的声音是由左上角的音色 (timbre) 区域所控制。我们可以直接去了解 Harmor 制造声音的原理，但是在此之前，我们先来看一看其他几个更加简单的合成器。

## 合成器一览 (The synths landscape)

FLStudio 中，入门阶段的合成器包括：

- 3XOsc，这个合成器有三个振荡器 (oscillator)
- Wasp，有 2+1 个振荡器，2 个低频振荡器 (LFO)，6 个滤波器 (filter)
- Wasp XT，也有 2+1 个振荡器，2 个低频振荡器 (LFO)，6 个滤波器，以及多个包络线 (envelope)

由于振荡器个数的不同，这些合成器合成的声音也不一样。振荡器发出来的声音可以进行混合，得到的声音可以被过滤，过滤本身还可以添加 LFO。这些合成器包含一些常见的过滤器比如高通 (high pass)，低通 (low pass) 等等。具体到什么是 LFO，什么是高通以及低通，这本书的后面会谈到的。

顺着这个合成器的发展规律，我们可以将 3 个振荡器增加到 6 个，再添加 1000 个旋钮。是的，这就是 Sytrus。在 Sytrus 中，振荡器的名字变成了操作器 (operator)，因为他们中的每一个都能做比振荡器更多的事情，所以再管他们叫振荡器就有点不公平了。

## 音量 (Volume)

常见的波形有“正弦波” (sine wave)，“锯齿波” (saw wave)，“三角波” (triangle wave) 以及“方形波” (square wave)。波的名字反映了这个波在示波器 (oscilloscope) 的形状。

播放一个音的声音越大，音量 (volume) 就越大，或者你也可以说波的振幅越大。当声音越小的时候，波幅也就越小。在 Harmor 中，这个操作就很简单，把旋钮转下来就行了。



你可能会傻乎乎地认为振荡器就是一切，然后坚信示波器是在音色制造过程中分析声音最好的方式。但事实并非如此。比如 Sytrus 中的 6 个操作器。这些小东西十分精巧，你可以对他们进行各种变更传输路径（reroute）的操作，你也可以慢慢地在标准波形之间改变波的形状。



图 1. 音色窗口

那么 Harmor 有多少个振荡器呢？很明显你可以看到两个音色窗口！这两个窗口的确会对 Harmor 中的音色产生影响。如果你刚刚开始了解 Harmor 或者音色设计我建议你先忽略第二个音色窗口，我们以后会回来再来看第二个窗口的。

在 Sytrus 中你可以右键波形，然后选择“转换到正弦和声”（convert to sine harmonics）来打开其波形公式的“内部运行方式”。这样做之后，你可以让 Sytrus 中所有的操作器发出任何可能的声音。现在我想请你思考一下，假如仅仅一个 Sytrus 的操作器的功能可以这样强大的话，我们为什么需要 6 个这样的操作器呢？这不是太多了吗？

具体到什么是音色（timbre），以及音色的制作过程，是作为一个初学者容易迷的地方，所以在此我将把他们分成能够理解的小区块进行解释。

我们首先通过一个简单的思想试验来解释 Harmor 中的振荡器工作原理。声音可以有多种角度来诠释。虽然示波器不是最好的理解声音的途径，但是示波器却是我们最早接触到的视波工具。示波器向我们展示了喇叭的震动方式，什么时候震动，以及什么时候停止。

FL studio 自带一个 Wave Candy 工具，我们可以用这个工具来对声音进行不同的视觉反馈操作。这里是一个 FL studio 工程的链接，工程里的 Wave Candy 已经被设定好来分析不同的合成器：<https://www.malmgren.nl/Harmor/Harmor-01.flp>

首先，我会向大家展示在 3xOsc 中，锯齿波长什么样。



图 2.在 3xOsc 中弹奏 C5。蓝色的 Wave Candy 窗口展示了振荡器的波形。右侧是一个频谱图。左下角是 Fruity parametric EQ2。

用 3xOsc 制作的锯齿波不是非常“干净”。这个锯齿波的频谱向上一直到大概 18kHz 的位置。接下来让我们看一个在 Wasp 中合成的锯齿波。



图片 3. 使用 Wasp 的锯齿波弹奏 C5。频谱较 3xOsc 更加清晰，但是大于 18kHz 的频率却逐渐消失。3xOsc 的波形像 N，Wasp 的波形则与之相反，但依然是一个锯齿波。

下面是 Sytrus 中合成的锯齿波：



图片 4. 这是一个用 Sytrus 弹奏的 C5

Sytrus 合成的锯齿波频谱很清楚，其频谱中的频率高达 20kHz

我更喜欢用 Wave Candy 的频谱图来分析声音。频谱比示波器能提供更多的信息。如果我们用 Wave Candy 来分析那几个经典波形的话，你会看到这样的图像：



图像 5. 使用 Sytrus 的经典波形来演奏 C5。你可以下载这个 FL studio 文件来仔细看一看：<https://www.malmgren.nl/Harmor/Harmor-02.flp>

我们可以发现：

- 正弦波只有一条在底部的线。这是基频（fundamental frequency），或者又叫做基音（fundamental tone）。剩下的频谱都没有声音。
- 三角波底部有一个频率非常响，频谱越往上声音越弱。
- 方形波与三角波中被激活频率（active frequency）的数量一样，但是方波中频率的振幅更大，声音更响。
- 锯齿波中被激活频率的数量是三角波的两倍。

我们可以看到在合成器中，正弦波几乎没有太大的变化空间，因为正弦波只有一个频率。三角波更好一点，但是两个最好的基础波形是锯齿波和方波。他们有很多不同的频率来进行变化，而且我认为这也是为什么这两种波形在 Harmor 中享有如此重要的地位——在默认情况下，音色窗口 1 (Timbre 1) 是一个锯齿波，窗口 2 (Timbre 2) 是一个方波。

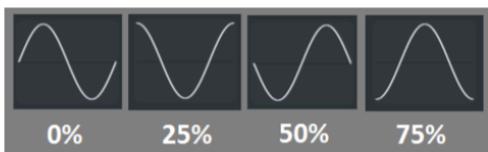
频谱中所有的频率都是正弦波（译者注：数学家傅里叶证明了所有的波都可以被分解为正弦波，因此三角波，锯齿波以及方波其实都是不同频率的正弦波的组合）。我们可以过滤掉所有频率，保留仅仅一个频率来进行观察。



图 6.我们过滤掉了所有的声音，除了在 1577Hz 处的声音。这是一个什么波形呢？你可以通过这个工程文件 <https://www.malmgren.nl/Harmor/Harmor-03.flp> 来看一看。这里我们把 Fruity parametric EQ2 放在了 Wave Candy 之前，所以只有一个频率的声音被展示了出来。虽然 EQ 切的不是很完美，有一些其他频率的声音泄露了出来，但并不影响展示效果。

如果要去创造某一个特定的波形，有 3 个成分是必须的：正弦波 (sine wave)，正弦波特定的频率 (frequency)，以及振幅 (amplitude)。但这并不是全部。每个频率振荡的起始位置也同样对塑造声音很重要。这个起始位置叫做相位 (phase)。

如果我们可以看到每个频率的相位的话，那会对我们的理解会有非常大的帮助，但是 Wave Candy 中没有这个功能。相位在创造波形的过程中也是非常复杂的。另一个展示相位的方法可以在 Sytrus 中进行，在 Sytrus 中我们可以看到波的相位，然后我们也可以在 Sytrus 中把玩这个参数。我们过一会儿会展示，但是现在让我们先看一看正弦波中不同的相位。



相位不仅非常晦涩难懂，在对其的语言描述上也常常令人摸不着头脑。相位的定义是振荡的起点，但这东西到底是什么呢？在 Sytrus 中是通过一个周期的百分比来描述相位的。当一个声音与另一个声音混在一起的时候，相位的影响就十分明显了。在 Harmor 和 Sytrus，以及大部分的合成器中，相位被定义成，波形中振荡器开始振荡的点。

Harmor 的音色窗口 1 是一个示波器，展示了当前的波形。现在我们先不去管音色窗口 2。音色窗口 1 的波由不同的音色水平 (timbre level)，以及谐波相位 (harmonic phase) 数据构成。音色水平指的是，有多少个振荡器在同时工作，以及振幅分别是多少——声音有多大。在 Harmor 中一共有 516 个小振荡器 (译者注：英文原文是 partial，有“部分”之意，这里译为小振荡器，以揭示这 516 个 partial)，这些小振荡器只发出正弦波。一个特殊的波形可以由多个小振荡器发出的正弦波组成，有时并不需要这 516 个小振荡器都同时工作。图 7 展示了有多少个小振荡器在工作。

比如如果你制造一个正弦波，那么就只有一个小振荡器在工作。谐波相位数据指的是每个正在工作的小振荡器的相位。总的来说，在 Harmor 中我们需要三个元素来组成一个波形：

- 已被激活的小振荡器的个数，以及他们的振荡频率
- 每个小振荡器制造的正弦波的波幅
- 每个小振荡器的相位

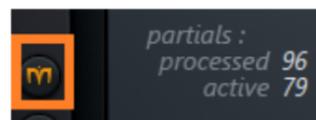


图 7. 在 Harmor 的左侧。你可以看到有多少个小振荡器正在被加工以及激活。

这三个元素以一种极其巧妙的方式结合在一起，使我们可以创造出任何波形。如果你没有领会上一句话，那么请不断地反复阅读上面这句话，直到完全理解为止，因为这一点极其重要。谢谢。

请观察 Sytrus 是怎样用正弦波谐 (harmonics) 波来构成一个三角波的：

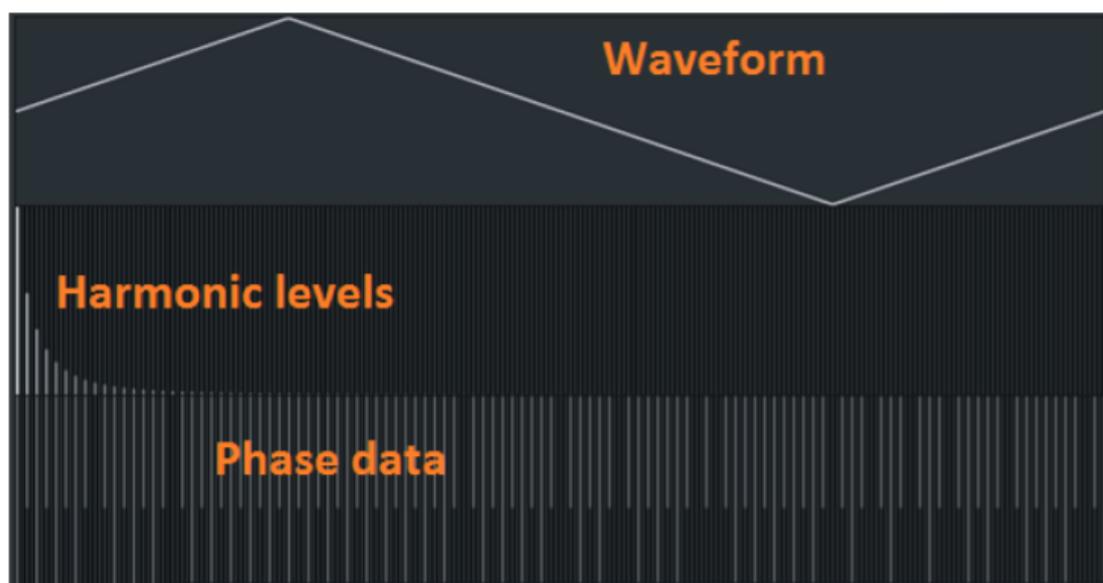


图 8. Sytrus 中的三角波

对于那些没有被激活的小振荡器来说，其谐波相位数据对波形的合成来说不会产生任何影响。

## Harmor 音色 (Harmor timbres)

音色窗口 1 有一个内置的滤波器，叫做“布朗噪声滤波器” (Brownian noise filter)。该滤

波器会逐渐的削弱音色中高频的声音。在使用 Harmor 的时候你是看不到这个滤波器的，但是这个滤波器的的确确存在。只有当我们与其他合成器进行比较的时候，我们才能发现这个滤波器，这个我们过一会儿再谈。之所以 Harmor 的设计者加入了这个滤波器是因为高频的声音有非常高的能量，而这个滤波器可以削弱这种能量。如果你回头去看图 5（那个展示了 4 种经典波形的图片），你会发现高频部分比低频部分更加稠密。对于一般的音色设计来说，把音色中高频部分的音量调的很高是没有太大必要的，正因为这样，设计者才在 Harmor 中添加这种自动削弱高频能量的内置滤波器。

Sytrus 没有内置的布朗噪声滤波器。如果我们在 Sytrus 中使用正弦谐波来构建一个波形的话，它看起来像图 9 这样：



图 9. 这是一个在 Sytrus 中使用正弦波谐波编辑器制造的波形。

上半部分是音色水平，也就是每个正弦波的振幅。下半部分是每个正弦波的相位数据。请注意这个 Sytrus 版本中有个小小的 bug，有一些相位信息被最下面的左右滚动条挡住了。

在图片 9 的相位信息中我把相位信息做成了一个阶梯。第一个台阶是 10%，第二个是 20%，然后一直到 70%。然后大于 74% 的相位信息不幸地被下面的滚动条挡住了。

Sytrus 一共有 255 个小振荡器。当你从 Sytrus 中导出一个仅包含一个周期的波形，然后再导入到 Harmor 中的时候，只有 255 个小振荡器被激活。另外我们还会发现。Harmor 还可以改变相位信息。结果就是我们会得到相同的波形，但是整个周期被移动了一半。也就是说，这个由 Sytrus 制造的单周期波形的终点是 Harmor 中波形的中间点。



图 10. 导入进 Harmor 的波形。如果你想打开谐波水平窗口，就点击这个图标

## 音色窗口 1 的谐波水平 (Timbre 1 harmonic level)

当你点击音色窗口 1 之后, Harmor 的下方就会出现导入波形的谐波水平。



图片 11. 导入波形的谐波水平 (harmonic level) 数据

在这里你可以看到布朗滤波器的水平在导入波形的过程中被补偿了。在 Sytrus 中我们有一条水平线, 水平线上有一些三角形的突起。在 Harmor 中这条线是逐渐上升的 (译者注: 这样的话被布朗滤波器削过之后就变成了水平的)。在第 255 个小振荡器那里, 谐波水平变为了 0。我们可以看到 Harmor 中的三角形变得更尖, 这是布朗滤波器的原因。

## 谐波相位 (Harmonic phase)



图片 12. 谐波相位窗口。相位偏移了半个周期。要想打开谐波相位窗口的话, 就打开包络线窗口 (envelope window) 上方的下拉菜单, 选择谐波相位。

谐波相位窗口中的相位被偏移了。这里你可以看到从 10% 开始的那些台阶。Harmor 的工具提示窗口 (tooltip window) 中显示的是 +0.10。在显示 +0.50 的时候, Harmor 跳到了 -0.5 然后继续一直到上升 -0.09。那是在 Sytrus 中被挡在滚动条后面的部分。除了这 255 个小振荡器之外的小振荡器的相位被设置到了 +0.24? ! 只要是小振荡器没有被使

用到，被导入之后这些小振荡器的相位就会被设置成+0.24 或者-0.25。我想这就是“特色”吧，因为在 Sytrus 中如果谐波水平是 0 的话，那么相位就不会显示。虽然不会被显示，但其实在其内部，相位还是有一个随机值的。在导入到 Harmor 中之后则会变得可见了。

下面是一个可以帮助我们理解 Harmor 音色的试验。先用 Sytrus 制造出一个看起来像“W”的曲线。



图 13. 这是一个用 Sytrus 制造的 W 波形

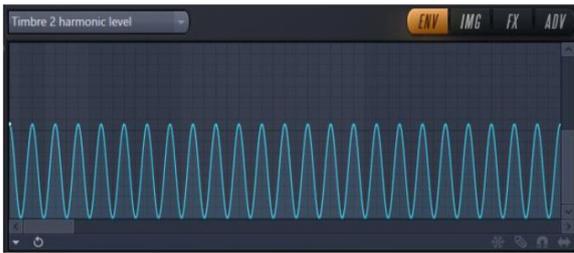
然后我把这个波形从 Sytrus 中导出，并导入到 Harmor 中。上半部分会被记录到音色 1 的谐波数据当中。下面的部分会被记录到谐波相位的数据当中。在导入的过程中，音频数据会被布朗滤波器过滤。

## 音色 2 谐波水平 (Timbre 2 harmonic level)

直到目前为止我还没有讲任何关于音色 2 (Timbre 2) 窗口的内容。现在我们已经做好了知识上的准备来开始讨论音色 2 了！

音色 2 有它本身的谐波数据，但是它跟音色 1 共享谐波相位数据。我们可以右键音色 1 窗口，然后选择“分析单周期波形……” (Analyze single-cycle wavefore)，然后相位数据就可以被导入，并与音色 2 共享。

在 Harmor 中音色 2 默认的谐波水平看起来像一个正弦波



图片 14. 音色 2 的默认谐波水平。每一个位置在偶数上的谐波都被关掉了

如果你还能回忆起来的话，频率越高，布朗滤波器对振幅的削弱就会越强。如果我们在 Sytrus 中反向设计音色 2 的话，它看起来会是像下面这样的。



图片 15. 这是我们在 Sytrus 中反向设计 Harmor 的音色窗口 2.

如你所见，我把谐波振幅（Sytrus 中是这么叫的）以一种我认为布朗滤波器会工作的方式来削弱谐波振幅。相位信息与刚才的 W 波形保持相同。

下面是 Harmor 的两个音色窗口！看，这两个波形还是挺像的。



在示范工程 <https://www.malmgren.nl/Harmor/Harmor-04.flp> 中 Sytrus 的 OP3 窗口你可

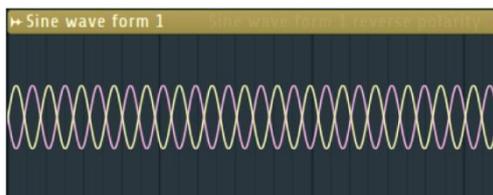
以找到刚才提到的“W”波形。在 OP4 窗口中你可以找到我反向设计的音色 2 中的波形。

这个“W”波形听起来有点像教堂里的管风琴！

## 音色 1 和 2 的混合 (Timbre 1 & 2 mix)

音色之间的混合是通过两个音色窗口之间的旋钮来完成的。旋钮的最左边是 0%，这时只有音色 1 会发出声音。最右边是 100%，这时只有音色 2 会发出声音。中间是 50%，这时你可以同时听到两种音色。之所以我们让这两个窗口共享相同相位，是因为这样你就可以在不担心相位抵消 (phase cancellation) 的情况下将两个音色进行混合。虽然这个设计非常巧妙，但是却不好理解。到底什么是相位抵消呢？这个现象最好是通过示波器来解释。

通常来讲，当我们在混合声音的时候，你一定会遇到相位抵消的问题。



一个波的波幅，比如说正弦波，总是会有正值和负值。当我们把这些正负值的信息输送给扬声器的

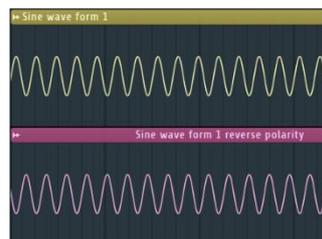


图 16. 正弦波 1 以及其倒置形态

时候，扬声器的纸盆就会向内向外运动。当我们把两个具有相同波形但相反相位的波进行混合的时候，他们就会完全抵消。

当一个声音使纸盆往外运动，另一个却在使纸盆往内运动时，结果就是纸盆会保持静止不动。

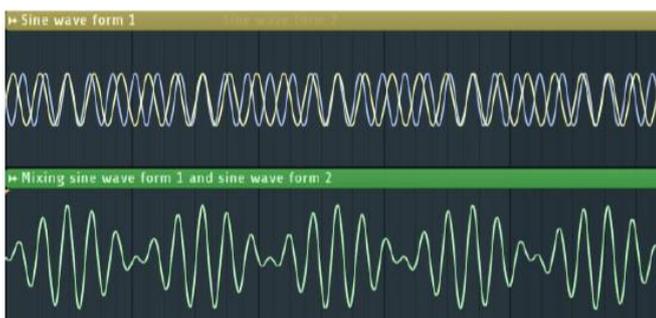


图 17. 混合两个频率不同的正弦波

在图 17 中，我们把两个波形混合在一起。黄色的是第一个正弦波，蓝色的是第二个正弦波。当他们相互抵消的时候，你几乎听不到任何声音，而当他们重叠起来的时候，振幅就会翻一倍。绿色的波是两个波叠加的结果。当相位抵消造成这种现象的时候，这叫做节奏抵消 (rhythmic cancellation)。

## 让-巴蒂斯特·约瑟夫·傅里叶 (Jean-Baptiste Joseph Fourier)

在 1807 年的时候，傅里叶证明了任何波形都可以被分解成正弦波的叠加。204 年之后，也就是 2011 年，Image-Line 公司基于这种思想，发布了 Harmor 这种加法合成器。

## 音色混合模式 (Timbre blending modes)

在音色窗口的下方你可以找到混合模式 (blend modes)。默认的混合模式是 fade (逐渐消失)。这个模式就是我们刚才一直在谈到的混合方式。

混合模式就是对音色 1 以及音色 2 的谐波水平进行数学运算操作。如果你把混合模式设置成 fade, 然后在左侧音色 1 窗口分析三角波; 在右侧音色 2 窗口分析正弦波。



音色窗口 1 中三角波的谐波水平长下面这样:



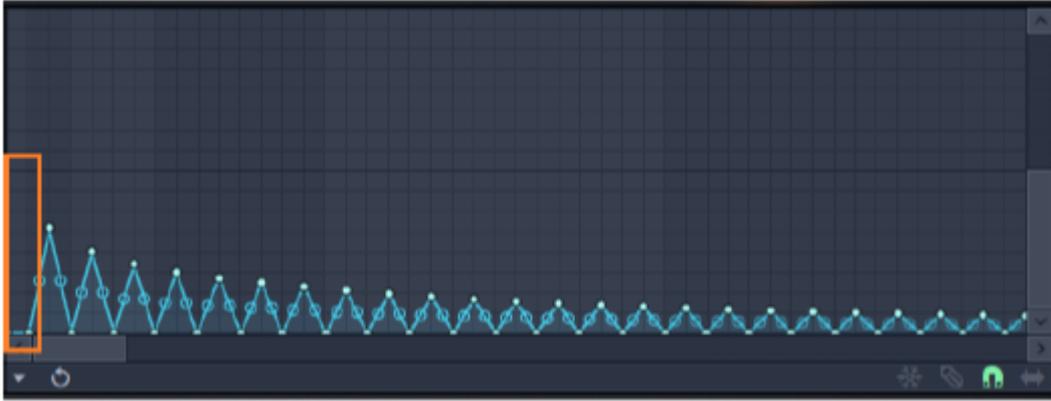
音色 2 的正弦波的谐波水平长下面这样:



如果我们把混合模式改成 subtract (减法) 的话, 音色 2 就会被替换成减法运算的结果。在 Harmor 中, 音色 2 窗口依然在那里, 只不过我们在音色 2 窗口中看到的是减法运算的结果。



我们是怎么得到这个结果的呢？其实就是三角波的谐波水平失去了处在第一个位置的谐波。如果我们手动去掉音色 1 的第一个谐波，我们就会得到相同的结果



现在我们获得了相同的图像，上面的解释好像是说得通的。但是为什么现在我们改变了音色 1，而音色 2 窗口的图像不改变呢？现在三角波的第一个谐波已经被我们手动调成了 0，然后混合模式是相减，那么 0 减 1 我们应该得到 -1 才对。这是因为振幅没有负值，所以结果保持不变。

另一个问题是，为什么在正弦波跟三角波共享相位信息的情况下，正弦波依然不受影响呢？这是因为在正弦波中，只有第一个谐波被激活。其他的谐波都没有被激活，因此相位信息不会造成任何影响。

在我所有的试验中，我发现混合的结果是音色 1 减去音色 2。然而 Harmor 的官方文档却说是音色 2 减去音色 1。

multiply（乘法）模式的工作原理比较好理解。比如，任何数乘以 0 都等于 0。任何数乘以一个小于 1 的数都会得到一个比原数更小的值。我个人觉得是音色 1 乘以音色 2，不过对于乘法来说，哪个乘哪个都无所谓。

下面是音色混合方法的选项：

- fade（逐渐消失） --- 当音色 1 淡出的时候音色 2 淡入（译者注：从音色 1 过渡到音色 2）
- subtract（减法） --- 音色 1 的谐波水平减去音色 2 的谐波水平
- multiply（乘法） --- 音色 1 的谐波水平乘以音色 2 的谐波水平。如果有任何一方的谐波水平等于 0，那么结果就是 0
- maximum（最大值） --- 选择音色 1 和音色 2 谐波水平中的最大值

- minimum (最小值) --- 选择音色 1 和音色 2 谐波水平中的最小值
- pluck (弹拨) --- 混合音色 1 和音色 2。两个音色窗口中的图显示的是在刚开始演奏音符时的波形。你可以简单的理解成是把音色 1 复制粘贴到音色 2 中，但是实际情况要更复杂一些。我们会在 *第七章弹拨与移相器 (Chapter 7. Pluck and Phaser)* 的时候再深入探讨，到那时我再解释这个模式到底是干嘛的。

## 相位随机 (Phase randomness)

在左上角的地方，你可以找到相位旋钮 (phase control)。



图 18. 相位旋钮

开始 (start) 旋钮设置的是波形的全局相位 (global phase)，数值大小是从 0 到 359 度。这个指的是所有小振荡器的起始相位。类似于一个全局起始数值。起始相位的默认值是 180 度。当我们使用图像合成 (image synthesis) 的时候这个值会自动跳到 0，我也不知道为什么。我觉得这个旋钮不会被用的十分频繁。

随机 (rand) 旋钮对于这两种音色来说都是相同的。而不是说不同的音色会有不同的随机相位 (当然这样也有可能很有趣)。

随机旋钮非常有意思。当它被旋转到左侧的时候，每个小振荡器的相位都分别被随机化了。这意味着波形被改变了。这是理所当然的，因为相位本身就是构成波形的一个要素。越是往左拧，随机值之间的方差或者是变动就会越大。

当我们把旋钮拧到右侧的时候，小振荡器的相位就被调整到了不同步模式 (free-running mode)。在这个模式下，整个波形的起始相位被设置成了一个随机的值。越往右拧，随机值就更加分散。

在这个工程中 <https://www.malmgren.nl/Harmor/Harmor-06.flp>，我们深入地看一看相位以及相位随机到底都发生了什么。

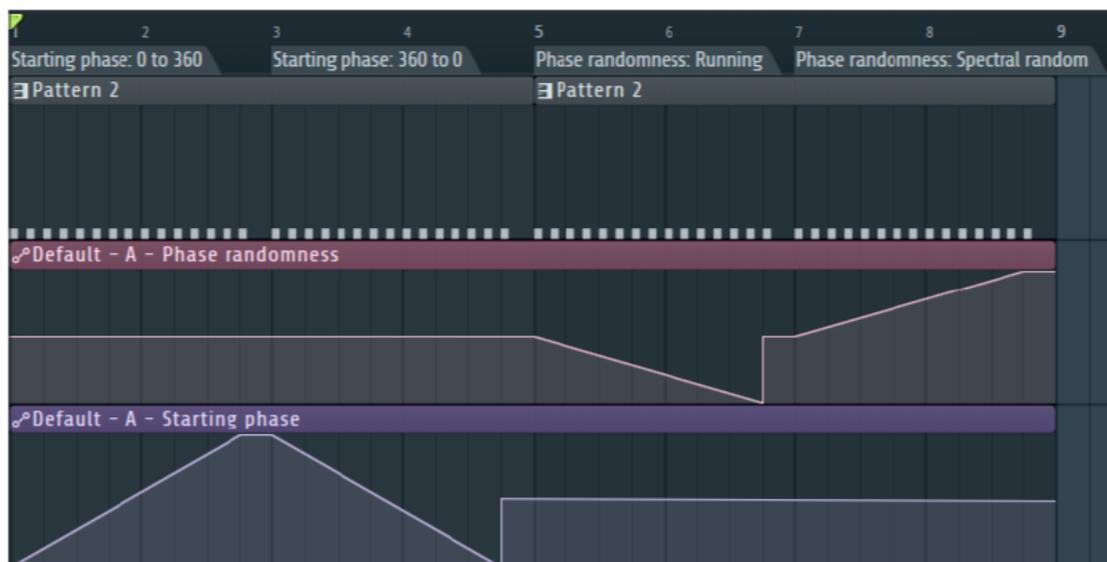


图 19. 相位以及相位随机对锯齿波的影响。在样式 2 (Pattern 2) 中, 我们弹奏 C4。然后我对这两个旋钮创建自动化包络线 (automation clip), 在播放的同时拧动这两个旋钮。然后我们可以把声音录下来, 进行分析。

这是缩小之后, Edison 中的结果:

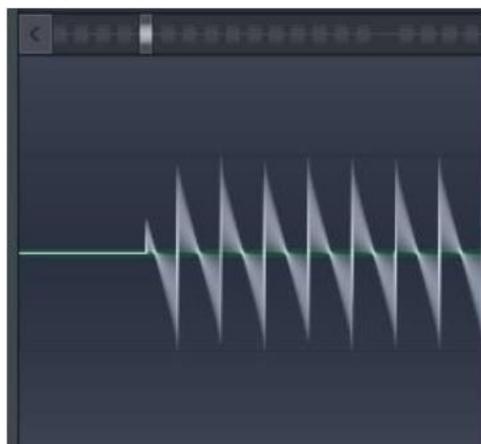


图 20. 使用 edison 录制下来的工程

接下来让我们放大仔细看看。



这是我们已经熟知的锯齿波。当我们往前滚动的时候我们可以观察到相位是如何变化的。



在从 0 增加到 360 的过程中，锯齿波的开端越来越多的被略过。在从 360 减少到 0 的过程中，结果相反。

下面的是相位随机部分。相位随机完全的扭曲了波形。



最后的部分是频谱随机 (spectral random)。我们可以看到相位随机越大，起始相位就越会前后反复地在音与音之间跳跃。

## 第二章 加法 / 减法合成 (Additive/subtractive synthesis)

现在我们已经储备了足够的知识，接下来我就要开始解释到底是什么使得 Harmor 与其他合成器相比如此不同。

在一个经典的合成器中，声音总是先以某种方式制造出来。可以通过振荡器或者波表 (wavetables)。然后合成器的主要部分都是用来处理声音的，比如使用过滤器消除某些特定频率的声音。于是处理声音的操作就被称为减法合成。消除声音的操作常常是在声信号上进行的。当我们一步接着一步处理声信号的时候，有可能每一步都会产生不同的伪迹 (artifact)。有些伪迹是我们想要的，有些不是。也许有些伪迹太常见了，以至于我们根本没有注意到他们

然而 Harmor 中的声音是由 516 个小振荡器叠加而成的。他们的振幅，频率以及相位结合起来构成了最终的声音。其他传统的合成器是先合成声音，然后再对声音进行编辑，然而 Harmor 是先使用有着极高自由度的过滤器在小振荡器上进行编辑，然后再合成声音。这样使得我们可以以一种无损的方式来处理声音，这在现代数字声音设计中是十分独特的



你可以把 Harmor 的上半部分理解成是对小振荡器进行无损处理的工作台。声音通过各个小振荡器产生以后，其大部分的声音特征是由 Harmor 交互界面的上半部分决定的。然后这些被处理过的一个个声音组合在一起，形成一个整的音频信号，最后再由下半部分的设置进行处理。甚至在下半部分，依然有处理单个小振荡器的设置选项。

Harmor 的工作原理是通过对小振荡器进行调整。Image-Line 公司也制作了许多经典滤波器来处理这些小振荡器。当我们把两种对声音的处理方式结合起来的时候，我们得到的就是加法/减法 (additive/subtractive synthesis) 合成。如果你把这个名字念得更快的话，最好说成令人上瘾的/吸引人的 (addictive/attractive synthesis) 合成，你觉得呢？



在无损音色处理结束后，你可以看到处理的结果。在使用界面的右侧有一个靠近眼睛符号的按钮。当你点击的时候，视觉反馈窗就会被打开。点击两次可以最大化。视觉反馈窗提供了许多有用的信息，为我们了解 Harmor 的工作原理很有帮助。

Harmor 自带的视觉反馈工具和 Harmor 外的视觉反馈工具，比如 WaveShaper 之间有一个巨大的不同。



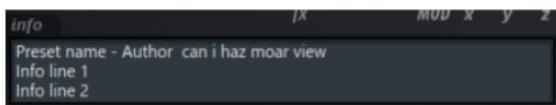
图 21. 这里我弹奏了两个锯齿波音符来向你展示不同的视觉反馈窗口。

在图 21 中，你可以看到不同的视觉反馈窗口。在工具栏中的母带监视器 (master monitor)，A，展示了一个小的示波器以及一个峰值电度表 (peak meter)。这个示波器与调到示波器模式的蓝色 WaveCandy 窗口，F，很像。红色的 WaveCandy 窗口，E，被设置成了一个矢量显示器 (vector scope)。它可以被用来很好地展示立体声信号 (stereo signal)。窗口 D 是一个频谱表 (spectrum meter)，它与在右上角的 Fruity parametric EQ2，也就是窗口 C，以及 Harmor 自带的视觉反馈窗口，B，很像。

在示范工程 <https://www.malmgren.nl/Harmor/Harmor-05.flp> 中，我已经把图 21 中的东西都设置好了。

Harmor 自带的视觉反馈工具完全显现不出来我是在同时奏响两个音符。他只能单独显示一个音的分析结果。

当你的光标在视觉反馈窗口中移动的时候，左上角的提示信息里会显示光标划过的频率值。



如果你想让视觉反馈窗口变得更大的话，你可以在左下角信息框中输入“can i haz moar view”

## 音调频率（Pitch frequency）

在 Harmor 中我们有三种方式来调整音调(pitch)。第一个调整的地方是根音设置(root note setting)。几乎每一个合成器都可以通过这个方式改变音调，点击扳手图标你就可以找到。



图 22. 各种各样的功能标签。右键一个钢琴键以设置根音。

把根音设置成 C4 可以让 Harmor 合成一个你希望得到的音符，换句话说就是，如果 Harmor 的根音被设置成了 C4，那么当你按下钢琴键盘上的 C5 键的时候，Harmor 也会合成 C5。而如果你把根音设置成 C5，那么当你按下钢琴键盘上的 C5 键，Harmor 则会合成 C4。然而现在先假设 C4 是根音吧。

第二个调整音调的地方是 Harmor 中的音调标签。



图 23. Harmor 中的音调频率

频率控制器 (frequency control) 可以以相对于根音的方式来调整音调。频率控制器展示了一个分数。当分子, N, 增加到 2 的时候, Harmor 就会制造一个比正常音调高八度的音。当增加到 4 的时候, Harmor 会制造一个比正常音调高 3 个八度的音, 以此类推。

当保持分子不变, 分母, D, 增加的时候, 就会制造低于根音的声音。

## 声调 (Pitch)

在声调标签下面你会找到两个单选按钮 oct 和 Hz。赫兹是合成声音最常见的模式。在这个模式下, Harmor 是按照我们所学的关于小振荡器是如何叠加在一起的方式来制造音色的。而如果我们现在有一个锯齿波, 然后把模式调成 oct, 所有的小振荡器都会从根音开始, 均匀的向上排列分布。这时我们得到的声音就不再是一个典型的锯齿波了。只有正弦波在当该模式在 oct 和 Hz 之间切换的时候才会听起来一样。

(译者注: 在 Hz 的选项下, 小振荡器的排列是按照谐波的排列方式。假如根音是 C3, C3 的频率是 130.31Hz, 那么第二个小振荡器的频率就应该是  $130.31 \times 2$ Hz, 第三个就是  $130.31 \times 3$ Hz, 以此类推。如果选项是 oct, 那么小振荡器就会从根音开始, 以一个音程单位的方式往上叠加。假如根音是 C3, 那么第二个小振荡器的频率就是 138.59Hz, 第 3 个就是 146.83Hz, 以此类推)

八度	音调 ( Hz )											
	C	#C	D	#D	E	F	#F	G	#G	A	#A	B
0	16.351	17.324	18.354	19.445	20.601	21.827	23.124	24.499	25.956	27.50	29.135	30.867
1	32.703	34.648	36.708	38.891	41.203	43.654	46.249	48.999	51.913	55.00	58.270	61.735
2	65.406	69.296	73.416	77.782	82.407	87.307	92.499	97.999	103.83	110.00	116.54	123.47
3	130.81	138.59	146.83	155.56	164.81	174.61	184.99	195.99	207.65	220.00	233.08	246.94
4	261.62	277.18	293.67	311.13	329.63	349.23	369.99	391.99	415.31	440.00	466.16	493.88
5	523.25	554.36	587.33	622.25	659.26	698.46	739.99	783.99	830.61	880.00	932.32	987.76
6	1046.5	1108.7	1174.7	1244.5	1318.5	1396.9	1479.9	1567.9	1661.2	1760.00	1864.7	1975.5
7	2093.0	2217.5	2349.3	2489.0	2637.0	2739.8	2959.9	3135.9	3322.4	3520.0	3729.3	3951.1
8	4186.0	4434.9	4698.6	4978.0	5274.0	5587.7	5919.9	6271.9	6644.9	7040.0	7458.6	7902.1



图 24. 主音调滑杆

第三个调整音调的地方是主音调滑杆。主音调滑杆的范围从-2400 分一直到+2400 分（译者注：一分是百分之一一个半音），加减两个八度的范围。

## 左右声道（Panning）



在窗口的顶部你可以找到左右声道的旋钮。通过这个你可以设置 Harmor 的左右声道输出。

## 谐波失谐乘数（Harmonic detuning multiplier）

失谐控制器（deruning control）也是以一个分数的形式来进行调整。默认设定是 1/1，也就是没有任何失谐。当我们增大分子时，基音的频率不变，但是其余的谐波都变得更加远离基音的频率。在图 25 中你会看到一个 C4 的锯齿波频谱是如何在失谐参数改变的时候改变的。

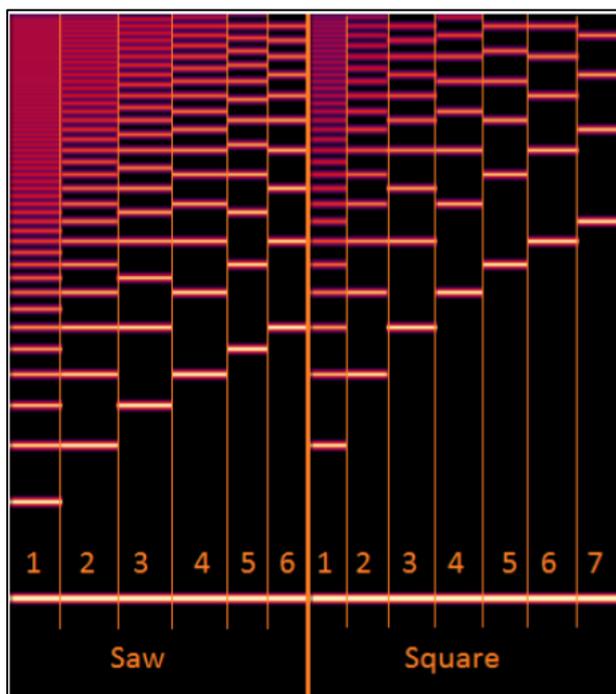


图 25. 失谐过后的锯齿波和方波。底部的横坐标是基音频率。底部的数字代表分子的值。

当我们增大失谐分母的时候，谐波会朝基音频率移动。



如果你想了解一下失谐更加进阶的用法的话，你可以看一看 Youtube 博主 SeamlessR 的这个视频：<https://youtu.be/-JUnNTFSekE> 每次当 SeamlessR 谈到 Harmor 的时候他总是说 Harmor 有 512 个小振荡器。为什么呢？Harmor 的官方文档可是说有 516 个啊。

## Oct 模式 VS Hz 模式

在 Harmor 的音调模式中，也同样有两个按钮 oct 和 Hz。这又是什么呢？



在整个 Harmor 中，你会在很多地方看到 oct 模式以及 Hz 模式。在大多情况下，默认是 oct 模式。这两个模式其实指的是小振荡器频率的分布。

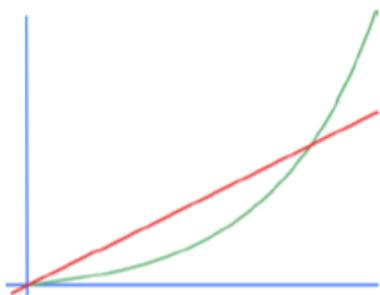


图 26. 对数 VS 线性。曲线是对数，脚踏实地。  
直线是线性。

总的来说，oct 模式是我们熟悉的频率刻度。在 oct 模式中，小振荡器的频率以对数的形式分布着。如果你回忆一下，前面在视觉反馈锯齿波的时候，高频的谐波数量要远远多于低频的谐波数量。

在 Hz 模式下，小振荡器是线性分布的。他们的频率在频谱中是均匀分布的。我发现在 Hz 模式经常会制造一个“处于外太空”的感觉，而 oct 模式则让人感到

许多时候，oct 模式是 Harmor 的正常设定，而 Hz 模式则是另一种选择。许多时候（但并不总是），Hz 模式会使频率更加分散。

## 第三章 齐奏 (Unison)

齐奏是一种让音色变得更厚的方法。其背后的道理跟传统乐器以及歌手一样。例如，一个管弦乐队中会有不止一位小提琴家，或者歌手们会同时一起唱。通常来讲，这背后的道理就是同时演奏同一个音符，但是演奏方法之间都会有一丢丢的不同。这就是齐奏 (Unison) 在做的事情。在电脑合成器中，这更加重要，因为电脑是可以预测的，除非我们故意加入一些随机性，或者用齐奏这种方式加入一些差异。

齐奏效果是通过复制一些合成器，然后再在亚音 (sub voice) 之间加入一些差异，最后再同时奏响他们。齐奏的控制器在 Harmor 用户界面中间部分的右侧



图 27. 齐奏

当阶数 (order) 右边的方框里没有数字的时候，齐奏效果是被关闭了的。你可以增大这个数字，从 2 到 9。数字指的是亚音的数量。你可以通过拖拽这个方块来改变数字的大小。

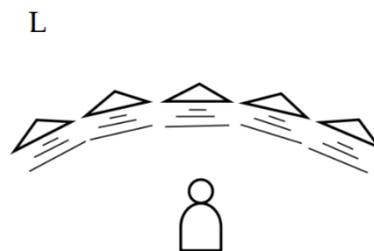


图 28. 包含了 5 个亚音的齐奏效果。你可以看到 5 个 Harmor 在同时演奏。

Harmor 中自带的视觉反馈窗口在改变齐奏数字的时候不会有任何差别。但在 Harmor 之外，你却有机会观察到其中的变化。

齐奏通过很多种方式来给声音之间加入不同：改变左右声道，音调，以及相位。

### 齐奏左右声道 (Unison Panning)

这个选项赋予亚音其自己的左右声道设置。滑杆可以从 0 到 100% 之间进行调节。调节后的结果可以在矢量显示器中查看

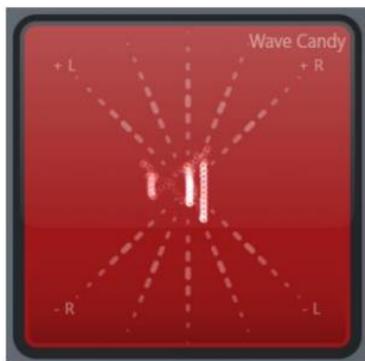


图 29. 锯齿波，齐奏参数 4，以及齐奏左右声道调节至 75%

当我们向上推左右声道滑杆的时候，矢量显示器中的白线分得更开，立体声变得更宽。

如果我们减小左右声道滑杆，他们就会变得更靠近中间，随之声音也会变得听起来更像单声道

## 齐奏声调厚度 (Unison pitch thickness)

我们可以通过调整齐奏声调厚度滑竿 (unison pitch thickness slider) 的方式在亚音之间加入声调的变化。滑杆可以再 0 到 100% 之间进行调整。在 100% 的时候，声调之间的差异会非常大，以至于听起来像是被失谐了。

## 齐奏相位 (Unison phase)

相位是各个亚音振荡开始的点。我们可以通过两个方式来使亚音之间的相位进行偏移：向上推滑杆调整的是亚音的相位会向前跳跃多少。向下推滑杆的时候调整的是各个亚音之间的相位会分散的有多么平均。当把滑杆完全推上去之后，就会产生一个特殊模式，也就是完全模糊模式 (full blur mode)。

要想解释向前跳跃模式 (jump ahead mode) 到底是什么非常难。为了说明这件事我创建了一个示范工程：<https://www.malmgren.nl/Harmor/Harmor-07.flp>。我在钢琴窗里创建了一组锯齿波音符，样式 1，在音符之间我添加了空白。这是因为我想看看每个音符的开端是如何根据不同的相位而改变的。然后我又给齐奏相位滑杆创建了一个自动化包络线。

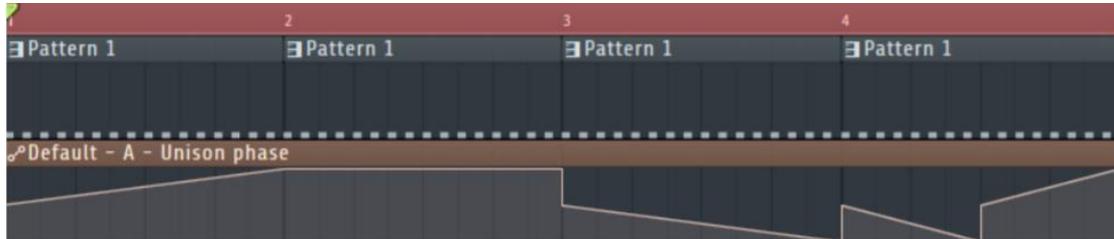


图 30。理解不同齐奏相位设置的试验。

第一小节以 50% 为开端，然后向上走。这是“向前跳跃模式”。在第二小节的时候停在 100% 的完全模糊模式。然后在第三小节的时候直接跳回到 50%，然后再向下走，这就是平均分散模式 (equal spreading mode)，一直到 0 为止。然后在第四小节的前半段，再次重复平均分散模式，后半段重复向前推进模式。

我将齐奏的阶数 (order) 调成了 3，也就是说有三个亚音。然后把结果录了下来，然后让我们来看看图 31 的结果。

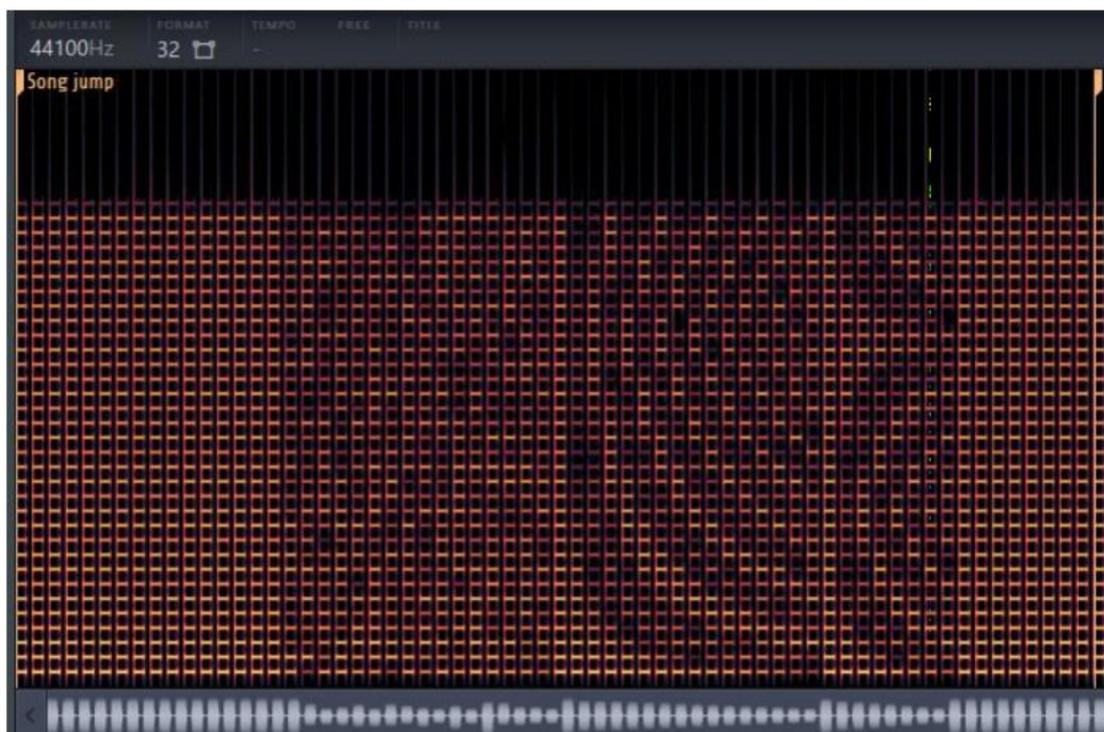
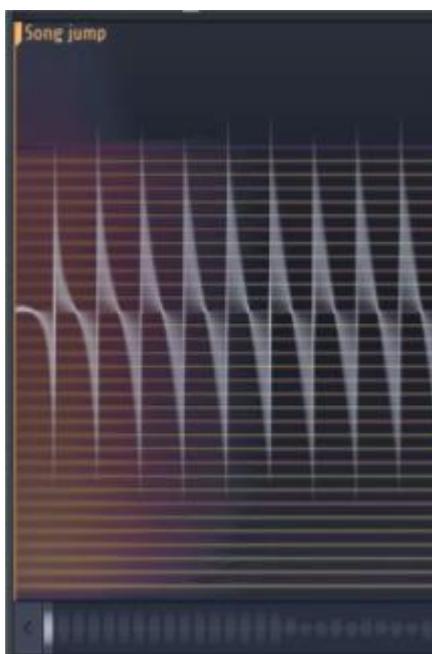
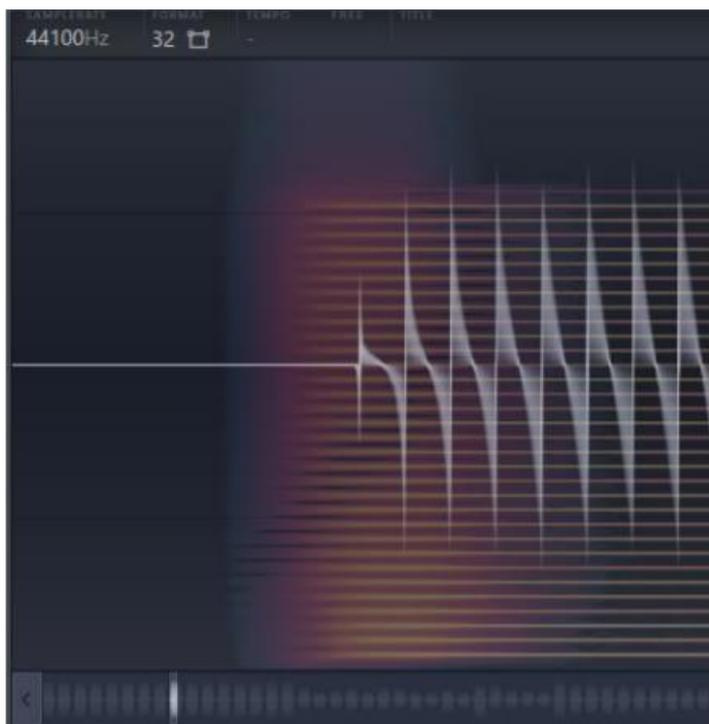


图 31。分析齐奏相位。用 Edison 展示结果

幸运的是，不同的小节在频谱中的分析结果看起来不一样。第一小节是最让人疑惑的。第一小节是向前推进模式，这个模式真的有一丁点作用吗？



为了看清楚到底发生了什么，我把 edison 调成了双视图模式。现在我们既有了一个频谱，也有了一个示波器。在最初的时候，在向前跳跃模式中，锯齿波是以向下走的趋势开始的。



当整段旋律向前运行的时候，锯齿波的起始点改变了，但是所有亚音的间隔依然是相同的。在这种模式下，我听不到有任何的声音效果。当齐奏的阶数增大之后，向前推进模式会增加振幅因为更多数量的波在同时被奏响，彼此会叠加，但是这是在向前推进模式下唯一可以被感受到的效果。

在第二个小节中，也就是完全模糊模式中，每个小振荡器的相位都被随机改变了



图 32. 齐奏，完全模糊模式

当对小振荡器进行改变的时候，整个波形就变得不规则了



图 33. 平均分散模式的相位

在第三个小节中，我们分析了“平均分散”模式。我们可以看到基本波形的个数与齐奏的阶数相同。这意味着各个小振荡器的相位之间产生了一些偏离，但是每个亚音其本身的相位不变。不然的话波形就要被改变了。当相位滑杆到达底部的时候，亚音也同时达到了分散的最大程度。当阶数为 2，并且波形是锯齿波的时候，音调就会翻倍。然后对于每一次阶数的上调，音调也会相对应的上升。对于一个方波来说，所有偶数阶数的亚音会互相抵消，最后的结果就是完全没有声音。（译者注：如果你是自己做这个试验的话，记得把滑杆 pan 拉到最低，不然的话还是会听到声音）

## 齐奏的分布 (Unison distribution)

Harmor 的官方文档上说齐奏的分布提供了音调以及左右声道的变化。大多数时候我能感受到音调的变化，而声道的变化以及相位的变化却很少。为了研究这个问题我创建了示范工程 <https://www.malmgren.nl/Harmor/Harmor-08.flp>。请看图 34 的结果。

这个下拉式窗口有 5 个选项：classic (经典)，uniform (均衡)，blurred (模糊)，random (随机) 以及 Hz (特殊)。classic 模式下的小振荡器看起来较其它选项有一个更宽的分布。uniform 模式有着最均衡的分布。random 以及 blurred 模式下的分布与 uniform 模式的分布一样，但是小振荡器的振幅却不同。在某些情况下，blurred 以及 random 是最有规律以及重复的模式，而不是像他们的名字说的那样。Hz 是一个完全不同的模式。Hz 模式看起来像是将小振荡器均匀分布。在这种情况下，你也可以认为 Hz 模式是把频率的分布点倒过来。基础频率是相同的，但是基础音调的 4 个频率（当阶数是 4 的时候）最为分散。所以对于每个频率来说，频率之间的间隔会越来越小。

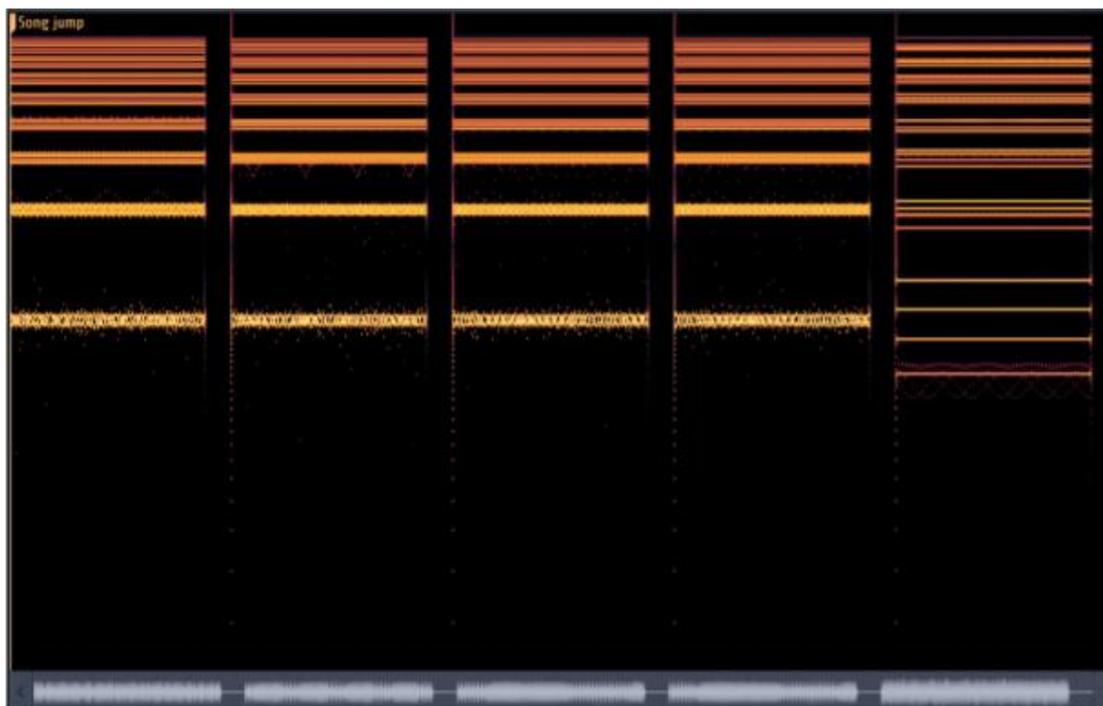
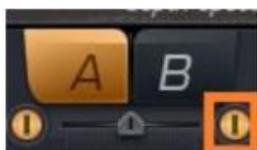


图 34. 为了研究齐奏分布之间的差别，我用不同的模式演奏了相同的音符，然后自己查找不同

## 齐奏交替的分布 (Unison alternate distribution)

在齐奏分布的下拉式菜单下面你会找到一个小的开关“alt”。这个开关提供了相位反转 (phase inversion)，一个加强立体声的技术。通过把一些右侧的信号添加到左侧，然后把一些左侧的信号添加到右侧。很明显，如果你想用这个东西，你必须先添加一点点声道平衡 (panning) 上的变化。这是我的猜测，因为官方文档在这方面解释的十分模糊。



Harmor 一共有两个部分，A 和 B。我们会用这个做一个试验。A/B 其实是两个不同的 Harmor，一个叫 A，一个叫 B。假设你把 Harmor-08 这个示范工程打开之后，把 B 下面的灯点亮，我们就激活了第二个 Harmor。现在我们会研究 Unison 的交替分布 (alternate distribution) 按钮。右键 A 中的齐奏声道滑杆 (pan) 并选择“复制到 B”。对阶数 (order)，声调厚度以及相位滑杆 (phase) 做同样的事情。在 A 和 B 中同时打开齐奏的 alt 开关。现在当我们播放音乐的时候，A 部分会自动化齐奏分布，但是 B 不会。当 B 被设置成 classic 以及 blurred 模式的时候，两个 Harmor 的相位会相互抵消。这对相位翻转来说很正常。当 B 被设置成 random，就不会产生相位的完全抵消。你可以得出一个结论，在随机模式下，的确有一定的随机性，使得随机模式的相位不能抵消。

现在有一个惊喜！当我们把 B 设置成 uniform 模式，那么 uniform 模式和 random 模式都能抵消。这两个模式之间的确有某种关联。我的结论就是这两个模式在某种程度上是完全一样的。

## 第四章亚音，剪切，音量，增益等 (Subs, clipping, volumes, gain etc)

在这一章我会结束对音色部分 (timbre section) 的讨论，也就是在图 35 中被虚线所围着的部分



图 35. 在这个图片中 Harmor 不同的部分被不同的橙色的线围绕着。在这一章，我们会探讨橙色虚线内的内容。

### 亚谐波 1,2 和 3 (Sub harmonics 1, 2 and 3)



在“亚” (sub) 标签的上方你可以看到 3 个滑杆。从左到右，他们分别对应着亚 1，亚 2 以及亚 3 (sub 1, sub 2 & sub 3)。这里也同样有个下拉菜单。这些滑杆以及菜单是用来增强基频的。增强方式有两种：

- 在基频周围 (Around fundamental)
- 在基频以下 (Below fundamental)

在图 36 中，亚滑杆以两种不同的模式被一个接一个地设置成了 100%。当然只要对滑杆进行一点点的调节就可以有明显的效果。

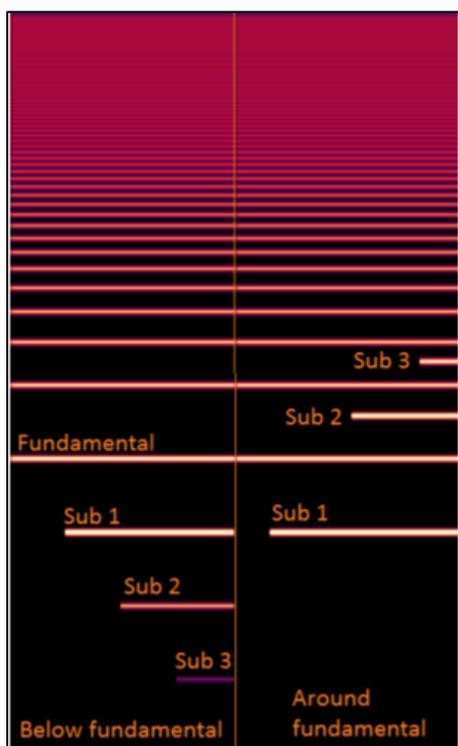


图 36。亚增强。这里展示了在两种模式下推动滑杆对 C5 锯齿波的变化

## 谐波保护 (Harmonic protection)

就像你已经注意到的那样，基频对于音色来说十分重要。一会儿我们会看到能够调整小振荡器的不同过滤器。我们有可能意外的弱化基频。为了防止这类事件的发生，我们有一个滑杆叫 prot。当增大这个滑杆的时候它会保护基频以防被破坏。

## 谐波剪切阈值 (Harmonic clipping threshold)

剪切滑杆控制的是在 clip 标签下的下拉菜单中，不同剪切类型的音频输入。当我们研究这些剪切类型的时候，把弹奏的结果可视化是十分重要的。



## 谐波剪切模式 (Harmonic clipping mode)

Harmor 一共有 5 种剪切模式：

1. high threshold (高阈值)。当剪切滑杆 (clip) 被推到顶部的时候，剪切是被关闭了的。当我们把滑杆向下推，剪切的效果就会越来越大，并且会减少基础频率的振幅。在默认模式下，这种剪切模式会忽略基频的保护。这是由于 ADV 标签中默认的单位阶 (Unit order)。在后面的章节中，我们会讨论更多关于单位阶的问题，以及如何通过改变不同的剪切方式也能使保护基频成为现实。
2. Soft low threshold (软低阈值)。当剪切滑杆在顶部的时候，剪切效果被设置成了最大，意味着整体振幅在此刻是最低的。当我们逐渐将滑杆向下移动的时候，越来越

多的小振荡器开始出现，从最低的频率开始，逐渐叠加更高的频率。对小振荡器的剪切不是那种突兀的剪切，而是十分温和轻柔。

3. Sharp low threshold (急剧低阈值)。这个方法和软低阈值很像，但是没有那么温和
4. Hard low threshold (硬低阈值)。这个方法和急剧低阈值很像，但是剪切更加突兀。
5. Subtraction (减法)。这个方法很像之前的三个剪切方法。剪切没有硬低阈值那么生硬，但也没有急剧低阈值那么柔软

## 干湿混响效果 (Effects dry/wet mix)

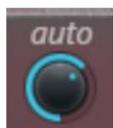


在剪切滑杆的右侧你可以找到 FX 干湿混合滑杆 (FX dry/wet mix slider)。默认位置是在中间的 0%。“FX”的意思是在 FX 标签中制造的声音。我们会在第 15 章失真 (*distortion*)，以及后面几章讨论这一点。有了这个滑杆，我们就可以对 FX 标签下的声音进行混音。

在 0% 的时候，所有的声音效果都会被送到输出端。在 50% 的时候，50% 的原声与 50% 的效果混合在一起。在 100% 的时候，声效被关闭，并且只有原声被输出。当滑杆被推到负值时，效果是完全一样的，只不过声信号被翻转了过来。如果你在示波器中将波型的视觉反馈窗口打开，你就可以看到，整个波形被上下颠倒了过来。

能够关闭效果器在 Harmor 中是十分重要的。也就是说，比如让 Harmor 的 A 部分拥有这种效果，而同时 B 部分没有这个效果。那么你就可以将 B 部分的滑杆设置成 100%。

## 自动增益 (Auto-gain)



自动增益 (auto-gain) 可以对声音的振幅进行“正则化”。其影响在当旋钮被设置到更高的值的时候才更加明显。被均匀分散开的小振荡器不会被自动增益所影响。当声音的平衡被打破时，自动增益才会开始显现出它的作用。比如如果我们的音色中使用到的小振荡器不多，那么当自动增益提高以后，音量也会增加。另外一方面，如果我们的音色中有许多小振荡器，使得声音变得很大，那么自动增益反而会减少音量。

当你对音色 1 以及音色 2 的默认波形进行试验的时候，你不会注意到任何自动增益的效果。你可以尝试用一个由很少小振荡器组合起来的音色来把玩一下，你就会发现音量被加大了。当你用由许多小振荡器组合起来的音色来做试验的时候，你就会发现音量被削弱了。

在第 24 章自动增益模式 (*Auto-gain*) 中，讨论不同的增益模式的时候，我们会再次回到自动增益模式这一点。

## 时间域音量， 频率域音量 (Time domain volume, frequency domain volume)

当声音还是一堆小振荡器的时候，声音的音量就可以被控制了。这就是被标记为 env 的滑杆，也就是频率域音量 (frequency domain envelope volume)。在 Harmor 的视觉反馈窗口中，你会发现当该滑杆被推到底部的时候，小振荡器的声音就会渐渐消失。信息栏显示的是 -59.2dB，然后一直到底显示的则会负无穷 dB (-INFdB)。在那一点你听不到任何输出，而且在视觉反馈工具中也看不到任何小振荡器。默认值是一半，也就是在信息栏上显示为 0dB 的时候。把滑杆一直推到顶部显示的是 12dB。

Vol 滑杆，也就是时间域音量滑杆 (time domain volume)，控制的是当小振荡器被组合成一个声音之后的音量。你在 Harmor 的视觉反馈工具中看不到变化，但是你可以听得到。

## 速度开关 (Velocity switches)

最后我们要讨论的是 vel 按钮。当最上面的选项被打开的时候，速度大小会决定有多少冲击 (attack) 会被跳过。如果第二个选项被点亮，速度大小会决定冲击的速度。第三个按钮把速度和释放 (release) 连续了起来。如果你想仔细研究研究的话，我创建了一个示范工程：<http://www.malmgren.nl/Harmor/Harmor-09.flp>



这个工程在 Harmor 中有一个默认的锯齿波。他的音量包络线被设置成像下图一样。



这个工程有一个样式 (pattern)，在该模式中有 3 个 C4 音符，音符的速度递减。



现在就是重点了。通过使用自动化包络线，我们就可以试验所有的组合了。



左侧是我们尝试 3 个选项所有的组合。第一个模式在冲击以及释放上有一个斜坡。很明显当我们看第 5,6,7 以及第 8 小节的时候，也就是当上图中的“release velocity to envelop release scale”这个自动化包络线被开启的时候，我们会发现释放时间被增加了许多。当我们比较“velocity to volume envelope attack scale”的第一和第三小节的时候，我们发现音量变大了。比较第一小节以及第二小节，似乎“velocity to envelop attack time”让冲击变得更快了。

在上面例子中，我们使用的是“velocity to volume”的默认设定。



这个默认设定是的速度可以影响主音量。如果我们关闭这个选项，然后再做一遍试验，那么每个音的音量都会变得一样，并且只有 vel 按钮会影响到速度，也就是图像右半边的情况。

## 第五章 . 基础滤波器 (Basic filtering)

在这一章中，我们会讨论图 37 中橙色线内的滤波器基础。这个区域中的上半部分跟下半部分是对称的。

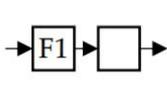
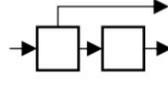


图 37. 基础滤波功能

为什么我们在这一章中会忽略 ENV 按钮以及 adapt 按钮呢？这些按钮在默认设置下什么效果也没有。如果想让 ENV 显现出效果的话，我们需要把他们与某个功能连接起来，我会在滤波器包络线量 (Filter envelope amount) 中再讨论这个问题。

### 滤波器 1&2 混合 (并联-串联) Filter 1 & 2 mix (parallel – serial)

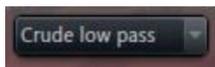
滤波器 F1 和 F2 可以单独运行，也可以是以并联 (parallel) 或者是串联 (serial) 的形式。默认混合量是 100%，并且这是串联模式：先是通过滤波器 1 然后通过滤波器 2，最后输出。下面的表格展示了所有可能的组合。我们从默认设定开始，然后把旋钮朝逆时针方向转动。

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
				

Harmor 中的过滤器作用在被合成之前的小振荡器上。这一点的好处就是声信号不会被滤波过程所影响，也就是说这样的滤波是“无损的”。

为了对滤波器进行试验，我们首先关闭 F2。也就是像上面的表格中第三列的场景一样。当我们这样做的时候，我们只是在观察改变滤波器上半部分的旋钮时的结果。然后打开视觉反馈窗口。现在我们就能够研究那些滤波器是怎样工作的了。

## 滤波形状 (Filter shapes)

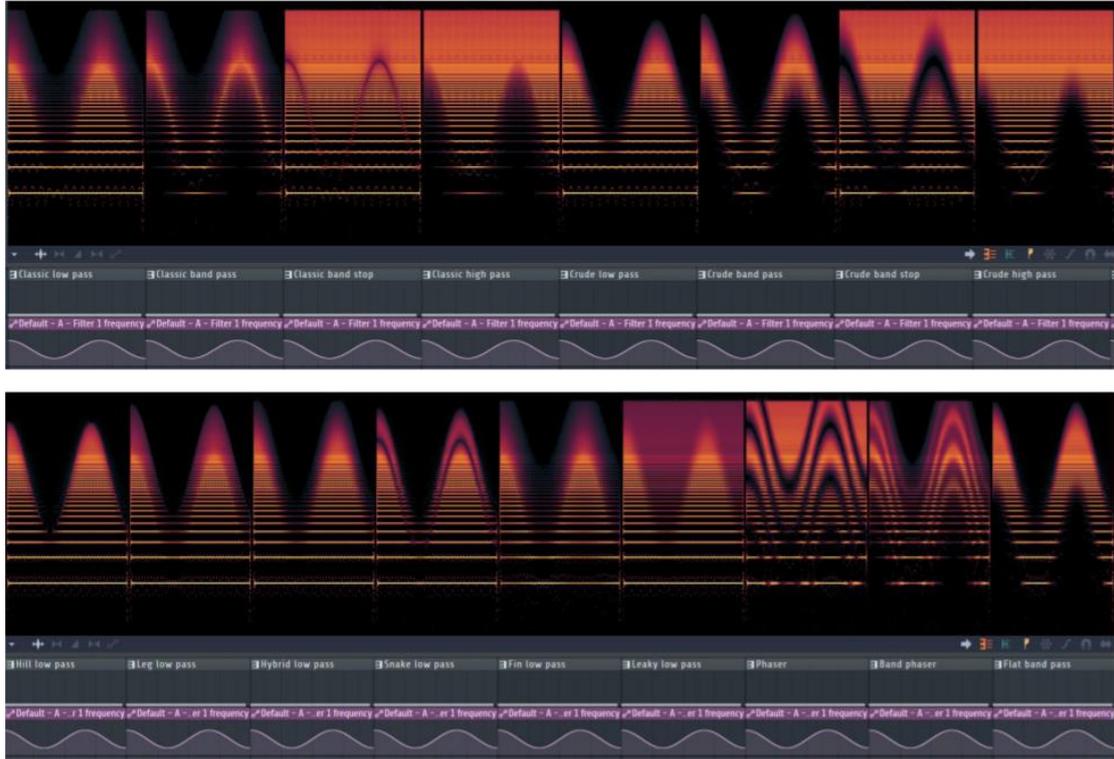


顶端的下拉菜单 Filter 1 shape 上写的是 crude low pass (粗糙低通)。这个下拉式菜单有 4 组过滤器：

- pass filter (通过过滤器)。通过过滤器会让一些频率通过，而另外的频率一点都通过不了。这种滤波器有 12 个
- stop filter (停止滤波器)。他们会阻挡频谱上的一部分
- phaser filter (移相器滤波器)。他们会阻挡频谱上的几部分
- custom (自定义)。我们会在更高级的过滤功能，Filter shape 中讨论
- 

在通过滤波器的例子中，按照传统习惯，频谱的排列方式是低频在下，高频在上。这也是 Harmor 以及 WaveCandy 的视觉反馈窗口的排列方式。在这种排列方式下，滤波器名字的由来就显得十分自然了。低通就是让低频通过。高通就是让高频通过。

在工程 <https://www.malmgren.nl/Harmor/Harmor-10.flp> 中，你会看到下面的图片。下面的图片展示的是下拉菜单中的所有滤波器的结果（除了自定义外）。



## 滤波器频率 (Filter frequency)



FREQ 旋钮控制的是一个“特殊点”。这个特殊点的功能取决于滤波器的类型。对于低通来说，特殊点的作用是，高于该点的频率不予通过。

## 键盘跟踪，从琴键到滤波器 1 频率 (Keyboard tracking, Key to filter 1 frequency)



因为滤波器是对小振荡器进行加工的，所以我们可以将滤波器与基频进行关联。当然，我们需要结合自动化包络线来研究。在工程 <http://www.malmgren.nl/Harmor/Harmor-11.flp> 中你会看到图 38 的试验结果。在这个工程中，我选择了 crude low pass，并且把 FREQ 旋钮调成了 50%

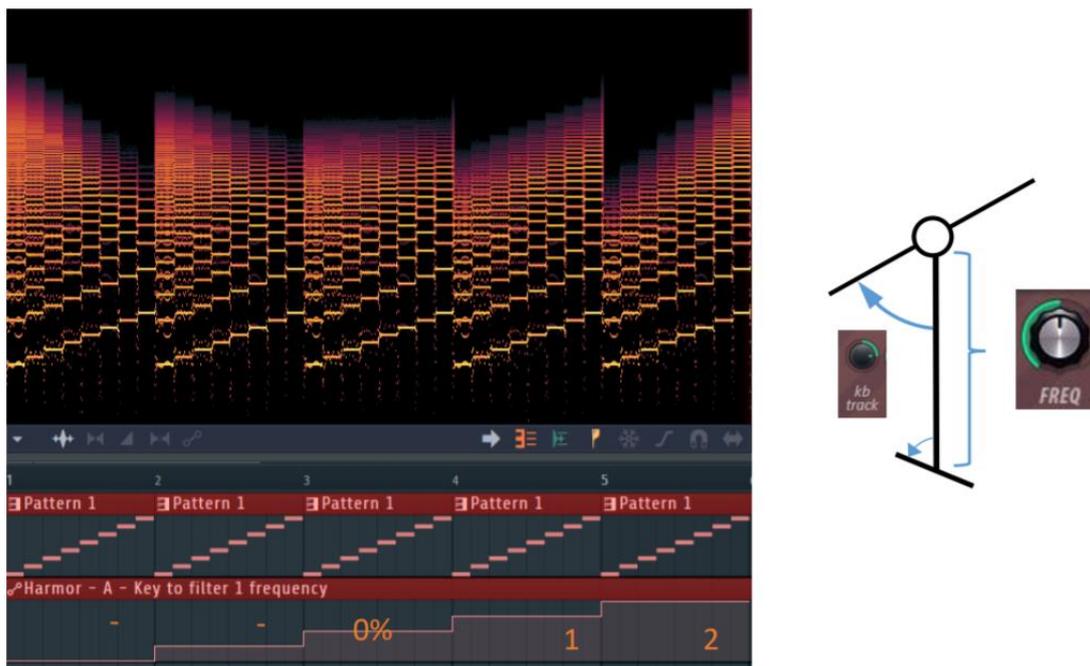


图 38. 键盘跟踪试验。将 key to filter 1 frequency 旋钮设置成不同的值。

(译者注：在 Harmor 中，kb track 旋钮就是 key to filter 1 frequency，把鼠标光标放在 kb track 旋钮上，你就能在左上角的工具提示信息中看到 key to filter 1 frequency 这个名字)

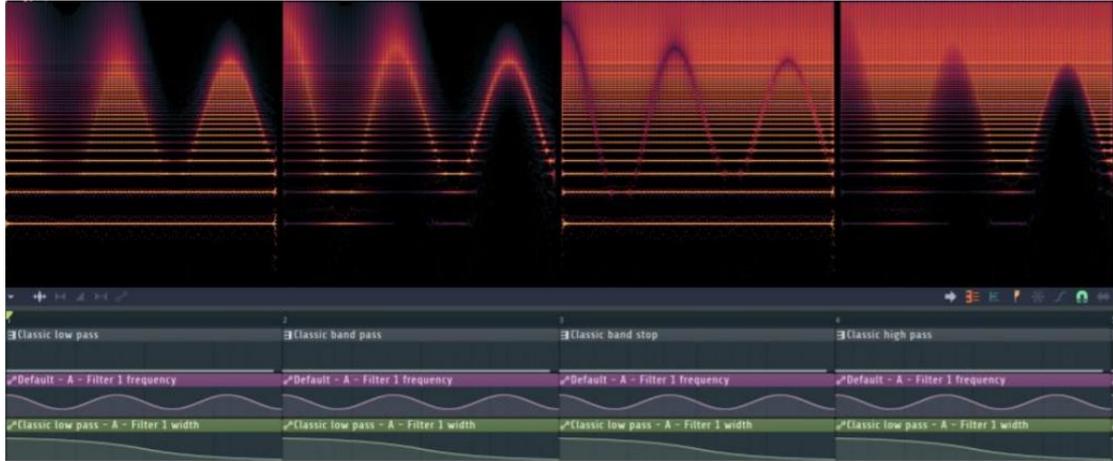
让我们先来看看图 38 中的第三小节。在第三小节中，我们把“key to filter 1 frequency”旋钮设置成了 0%。当是 0% 的时候，滤波器 1 的频率会在不影响基频的情况下影响截止频率（译者注：所有音中的最高频率都相同）。在这里我想使用一个比喻来解释 kb track 旋钮的工作原理。filter 1 frequency 影响的是一个旋转杠杆的高度。‘key to filter 1 frequency’和基频则影响杠杆的角度。我原本想弄明白其背后的数学原理，但是现在表示放弃。

## 过滤器宽度 (Filter width)

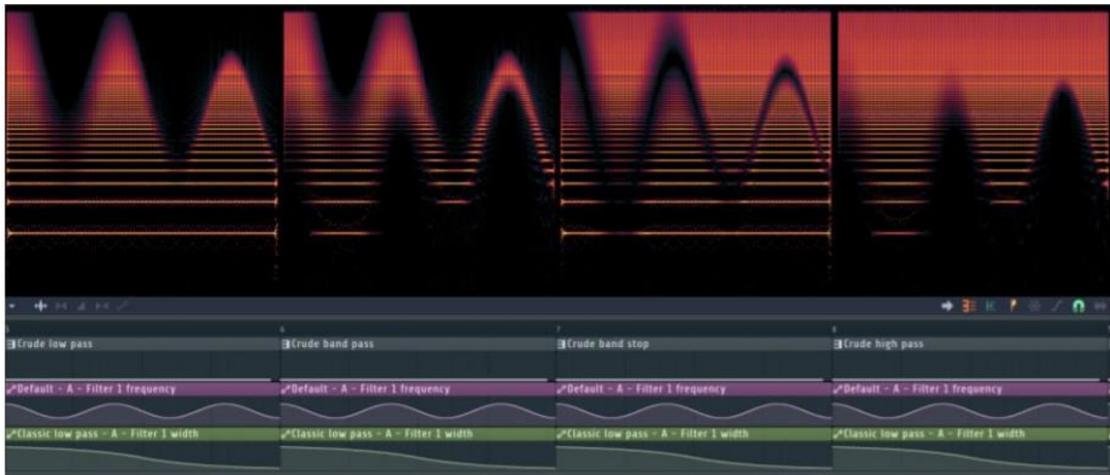


现在我们来讨论讨论过滤器宽度。幸运的是，这个用 WaveCandy 的频谱非常容易展示。在下面的图片中，我把宽度的大小从“非常宽”慢慢地变小。演示工程是：

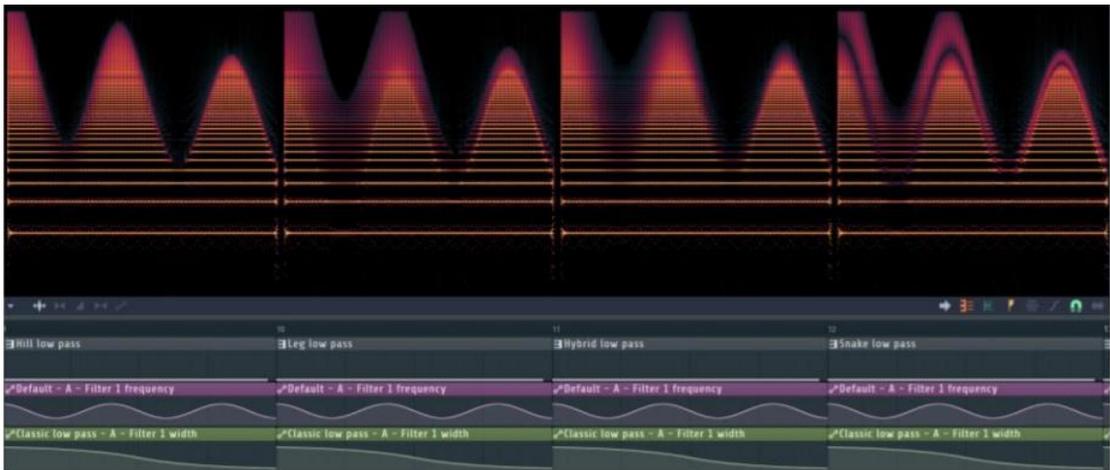
<https://www.malmgren.nl/Harmor/Harmor-12.flp>

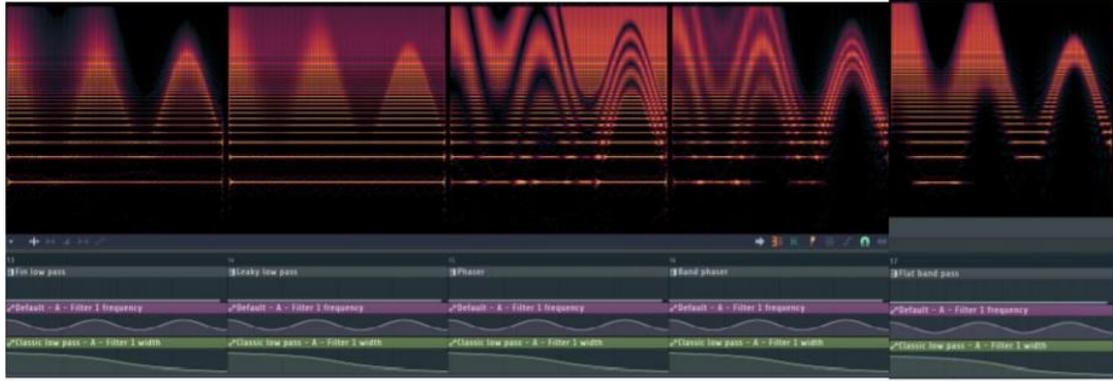


classic low pass (经典低通) 开始很模糊, 渐渐变得越来越清晰明朗



crude band pass (粗糙带通) 以及 band stop (带停) 都是越来越窄

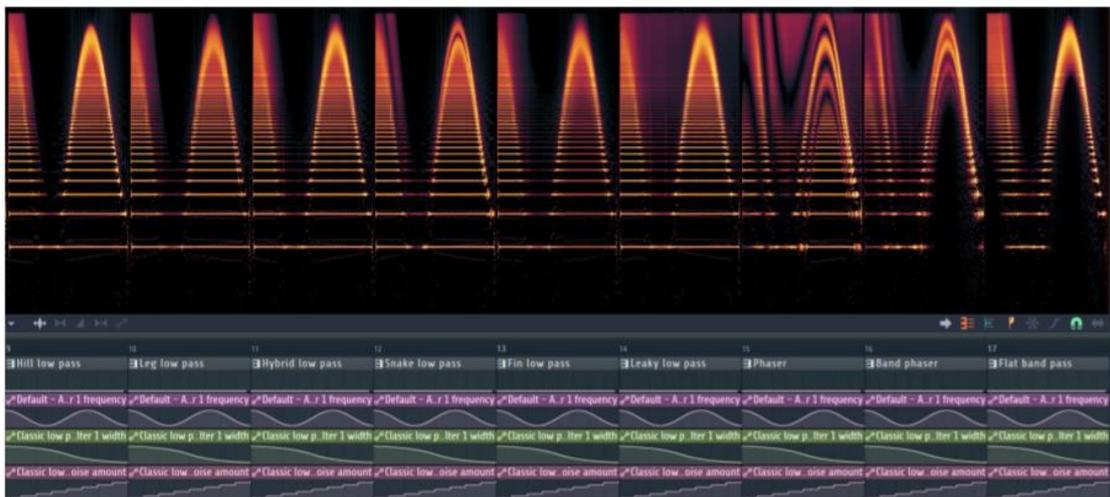
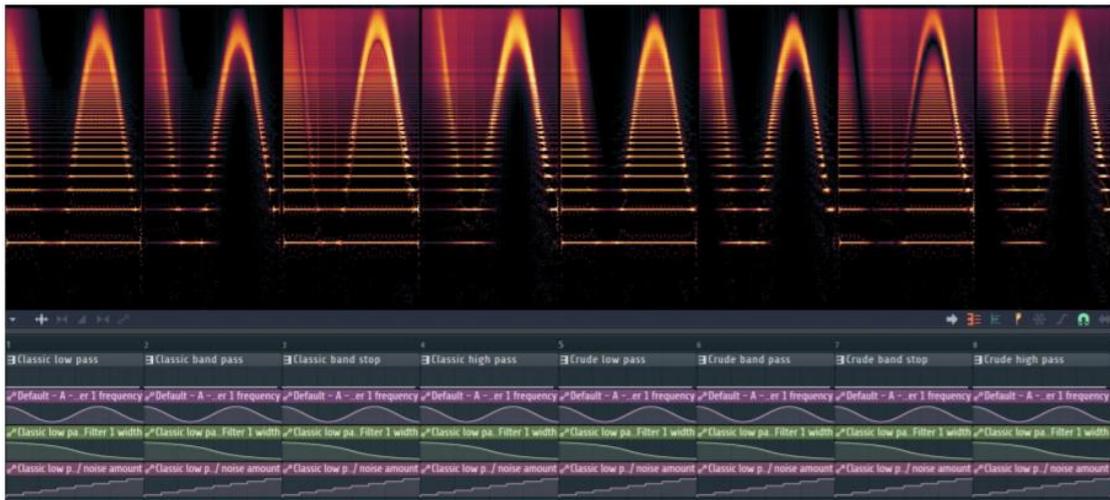




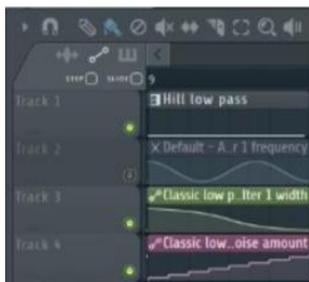
## 过滤器共振量 (Filter resonance amount)

接下来我们将会看一看 RES 旋钮的作用是什么。如果你想看一看这个试验的话请打开示范工程：<https://www.malmgren.nl/Harmor/Harmor-13.flp>。正常来讲在传统过滤器中，当共振被打开时，在截止频率的地方会有一个隆起。下面是在经典低通图像上的演示。

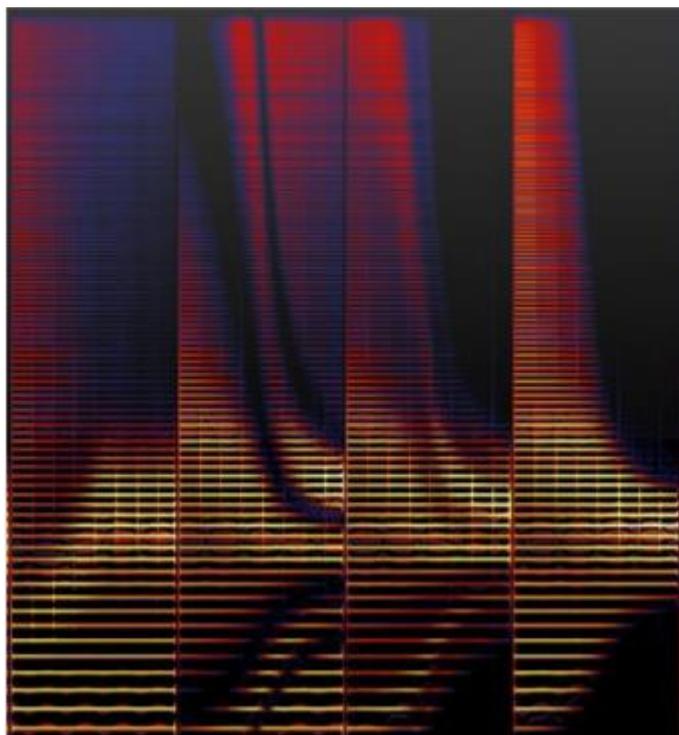
这里我决定使用阶梯自动图像来逐步增加 RES 的大小。特别是在高通中，你会发现随着 RES 旋钮的增大，噪声会逐渐减小。



如果你下载了示范工程的话，你可以做出更多的调整。比如你可以自己调整自动化包络线的开启或者关闭。



如果你点击了第二轨的单选按钮，并且打开混音台，激活第一轨的 WaveCandy，这样我们能发现其他的一些形状！（译者注：打开 WaveCandy 中的 visible 开关）

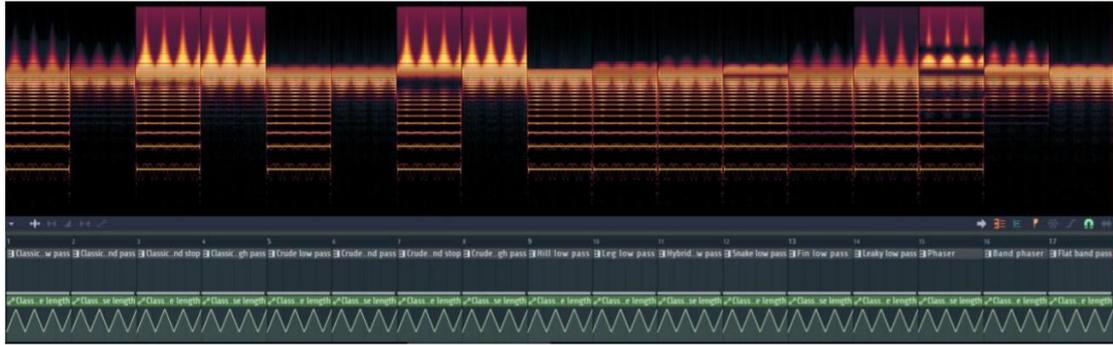


leaky low pass (漏低通)，phaser (移相器)，band phaser (带移相器) 以及 flat band pass (平带通)，特别是 phaser，听起来有点像人声。

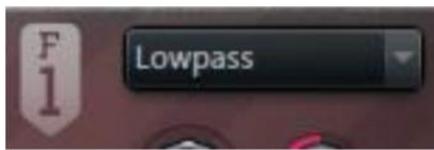
## 过滤器共振宽度 (Filter resonance width)



在 Harmor 中，共振也有其本身的宽度。当我们改变共振宽度的时候，我们可以制造更多人声。在示范工程 <https://www.malmgren.nl/Harmor/Harmor-14.flp> 中，你可以看到不同的过滤器，以及被自动化的过滤器共振宽度 (filter resonance width)。



## 过滤器共振类型 (Filter resonance type)

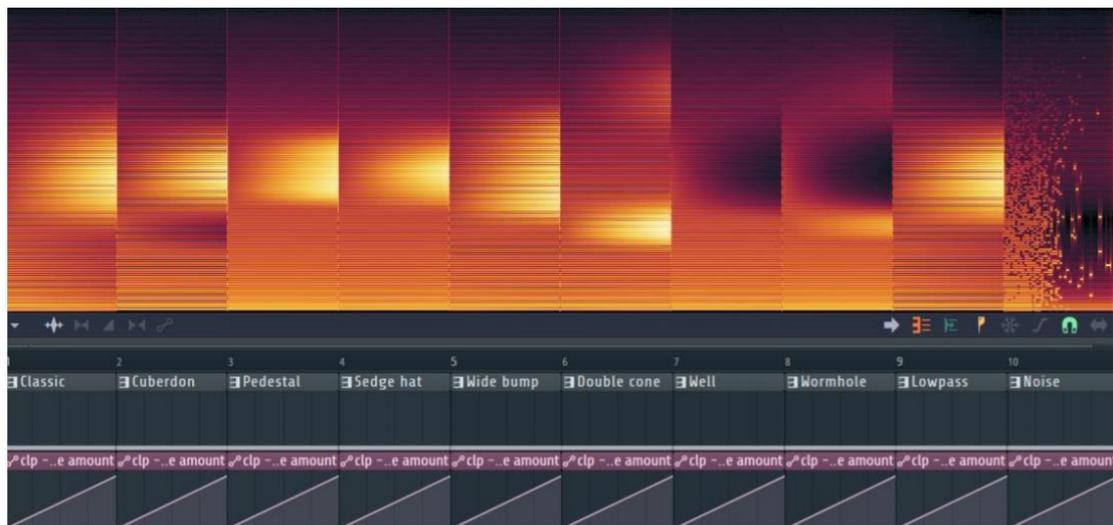


共振类型有一个下拉菜单。有 9 个已经预制好的共振类型，以及两个自定义的类型。我们会在共振形状 (resonance shape) 部分中讨论自定义共振类型。

为了研究共振类型 1，我创建了一系列的工程文件。在这些文件中我只使用了 crude lowpass 过滤器，以及不同的共振类型和不同的变量。

在我们的第一个试验中，只有共振 (resonance) / 噪音量 (noise amount) 旋钮被自动化了。截止频率 (cutoff frequency) 被设成 80%。

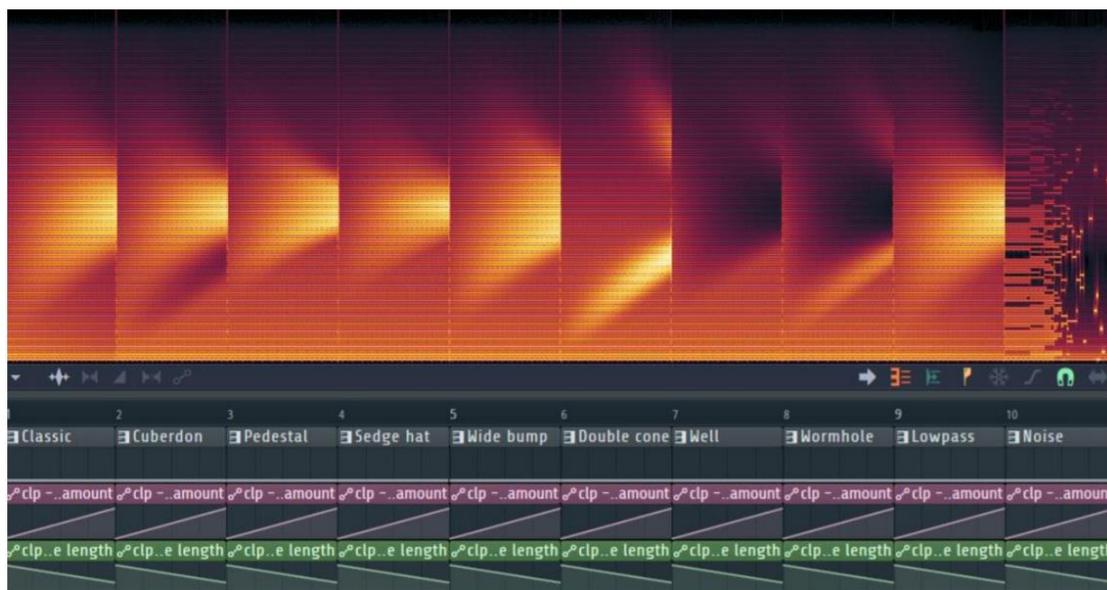
<https://www.malmgren.nl/Harmor/Harmor-15.flp>



在这个图像中，我关掉了 Edison 中的自然权重 (natural weight)。低通是默认的共振类型。它的设计是为了降低高于截止频率的小振荡器的振幅。作为一个默认共振类型其非常有用，因为这与传统上的共振类型的工作机制几乎完全相同。

在我们的下一个试验中，我们将共振宽度自动化了，同时也与不同的共振类型结合起来。<https://www.malmgren.nl/Harmor/Harmor-16.flp>。这个试验很好的展现了控制宽度的旋

钮的作用。

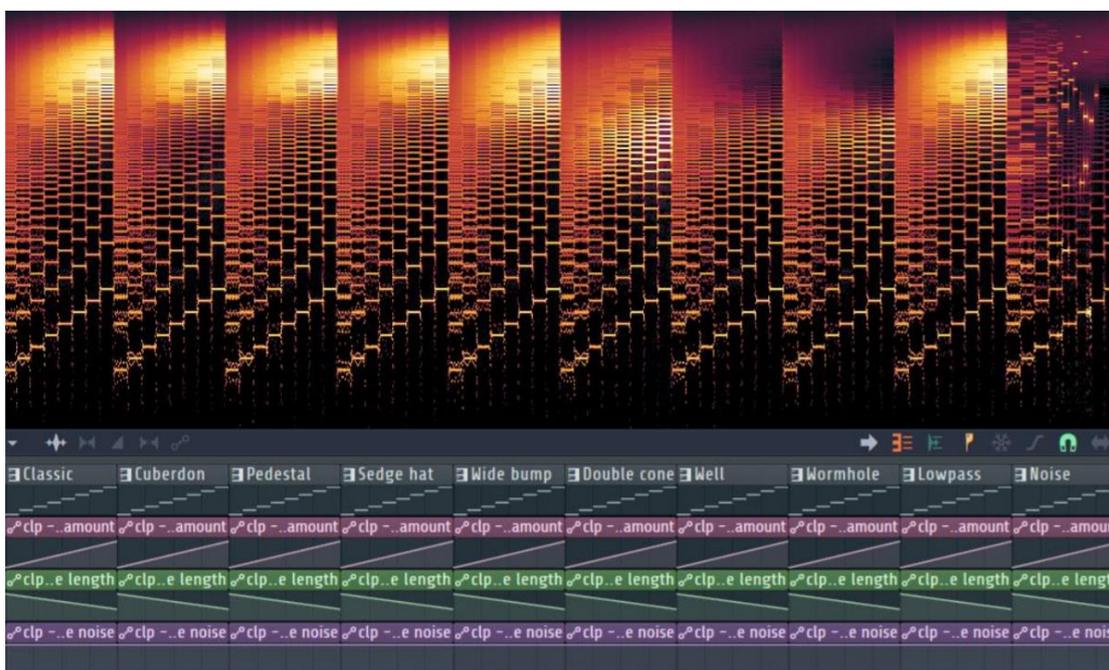
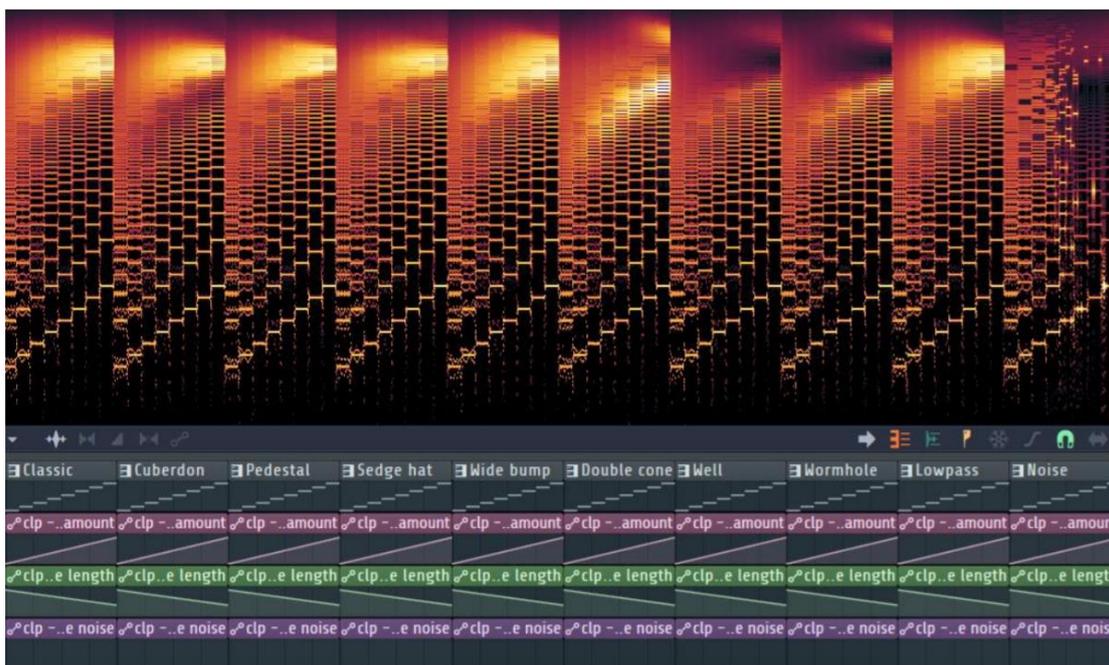


## 过滤器共振适应宽度/自由噪音 (Filter resonance adaptive width/free noise)



宽度旋钮，RES，的旁边有一个 adapt 按钮。这个按钮和 RES 旋钮之间被一条黑色的线所连接，这似乎在暗示我们这个按钮可以改变共振宽度的行为。宽度 (width) 基于过滤器带宽的变化量与基频之间有一个函数关系。为了验证这一点，我创建了一个样式，在这个样式中我添加了一些音调不断变高的音字符串，然后我在 adapt 按钮打开以及关闭的情况下分别播放这个模式，来比较两者。示范工程：<https://www.malmgren.nl/Harmor/Harmor-17.flp>。

第一张图是 adapt 没有被激活，第二张图是 adapt 被激活。



你可以观察到，当 adapt 被激活的时候，共振并没有随着音调的升高而升高（译者注：这里最好下载文件自己试一试，个人感觉听一听其中的变化更有助于理解）。这是只有加法合成器才能有的功能，因为在经典的减法合成器中，合成器没有基频的信息，但是在加法合成器中，过滤器拥有这种信息，而且可以加以使用。

## 过滤器共振偏移 (Filter resonance offset)

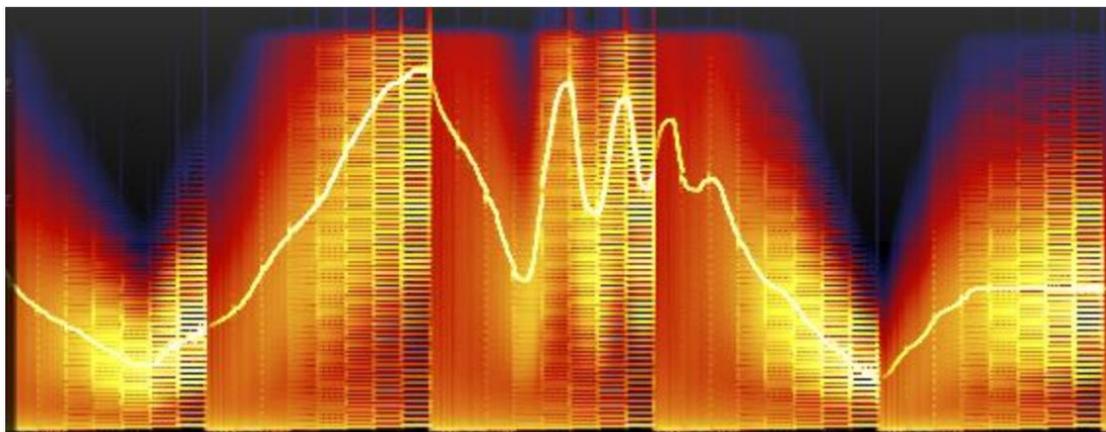


Ofs 旋钮可以让我们在截止频率和共振之间插入一个偏移，范围在+2400 和-2400 分之间。之前我们讨论过滤器效果的时候，共振总是与截止频率挂钩。但是现在有了 ofs 我们可以将共振移动到截止频率的下方或者上方。

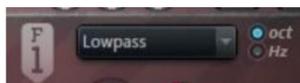
## 过滤器共振自振荡 (Filter resonance self-oscillation)



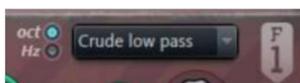
这是共振自振动效果器。Harmor 的官方文档说这个旋钮可以制造一个“不同色调的声音” (tonal sound)，这说的一点也没错。在视觉反馈视图中你可以看到共振的中心点，也就是中间的那条白线。



## 过滤器 oct/Hz 开关 (Filter oct/Hz switches)



在过滤器共振类型 (filter resonance type) 的下拉菜单旁边你可以找到这个 oct/Hz 开关。当 Hz 的灯亮着的时候，共振是在频谱中分散开的，但是在 oct 模式下，共振更加集中



在过滤器形状 (filter shape) 的旁边你可以找到过滤器比例的开关。当你在尝试使用这个开关的时候，你并不能立刻感受到声音上有什么明显的变化。好像这个比例在 Hz 模式下被压缩了。

## 第六章 . 模糊 (Blur)



模糊是把小振荡器横向抹开。如果你在锯齿波上试验这个效果，你几乎注意不到什么变化，因为这个波形已经占用了所有的小振荡器，所以没有多余横向抹开的空间了。当我们把波形换成方波或者正弦波的时候，你就发现其中的不同了。当我们将这些没有使用太多小振荡器的波形与锯齿波的小振荡器混合起来之后，我们就可以加入模糊效果了。

### 谐波模糊量 (Harmonic blur amount)

模糊 mix 旋钮的默认值是 100%，谐波模糊量 harm 的默认值 0%。结果就是在默认模式下，模糊效果被关闭。当我们慢慢加大 harm 的时候，效果也慢慢越来越明显。

## 顶部和底部张力 (Top and bottom tension)

我们可以将高频以及低频的效果平衡一下。这个效果有两个旋钮:

- 谐音模糊顶部张力 (harmonic blur top tension), 0 到 100%
- 谐音模糊底部张力 (harmonic blur bottom tension), 0 到 100%

张力按钮的默认设置是 50%，这也是他们的平衡点所在之处。

你可能会想，他们的名字几乎完全相同，他们也被放在了一起，因此他们的工作原理是相同的。但是实际情况却不是这样。他们的工作原理恰恰相反！

当顶部张力按钮是 100% 的时候，声音中的大部分的频段被压制，当是 0% 的时候，声音中大部分的频段则被放大。

当底部张力按钮是 0% 的时候，大部分的频段压制，当是 100% 的时候，大部分的频段则被放大。在我看来，底部张力好像控制的是所有的频段，而顶部张力只控制顶部频段。

## 模糊冲击以及衰减 (Blur attack and delay)

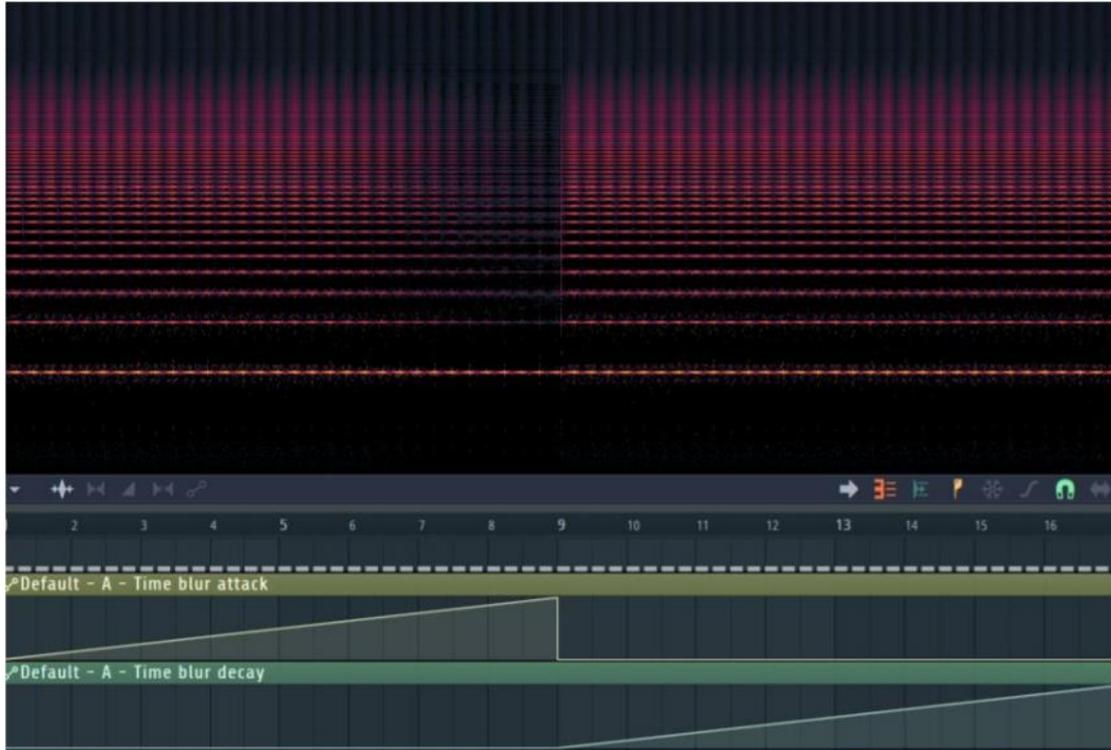
在左边有两个旋钮:

- 时间模糊冲击 (Time blur attack), 0 到 100%
- 时间模糊衰减 (Time blur decay), 0 到 100%

冲击 (attack) 旋钮影响的是该效果到底有多快会开始。衰减 (decay) 影响的是该效果有多快会消失。我至今还没有发现任何时间模糊衰减的影响。我已经用音量包络线以及自动化包络线试验过了，但是还是没能发现这个旋钮到底有什么效果。

现在一个重要的问题是，这个 100% 是什么的 100%？我尝试用一个 32 小节的长音来试验。整整 32 小节啊！当我加上了自动化包络线，并逐步增大时间模糊冲击的时候，终于可以听见，也可以看见其效果了。

在这个示范工程 <https://www.malmgren.nl/Harmor/Harmor-18.flp> 中，你可以看到我是怎么对这些旋钮进行测试的。



我用的波是一个正弦波。我把混合 (mix) 设置成了 72%，你可以很清楚的看到最下面的基音，以及模糊造成的其他声音。谐波模糊量被设成了 100%。顶部张力被设成了 100%，底部张力 50%。自动化包络线分别包含了两个时间旋钮。在第六小节的时候你会开始发现时间模糊冲击开始使一些频率减弱。这时自动化包络线的数值是大概 65%。就像我说的一样，增加时间模糊衰减在这种情况下不会造成任何影响。但这并不是说这个旋钮没有用。如果跟另一个效果结合在一起的话，就会产生效果，比如 *第七章弹拨和移相器 (Pluck and Phaser)* 中的弹拨效果。

## 第七章 . 弹拨效果和移相器 (Pluck and Phaser)

在这一章中，我们会讨论过滤器的下半部分。让我们先从弹拨效果开始。

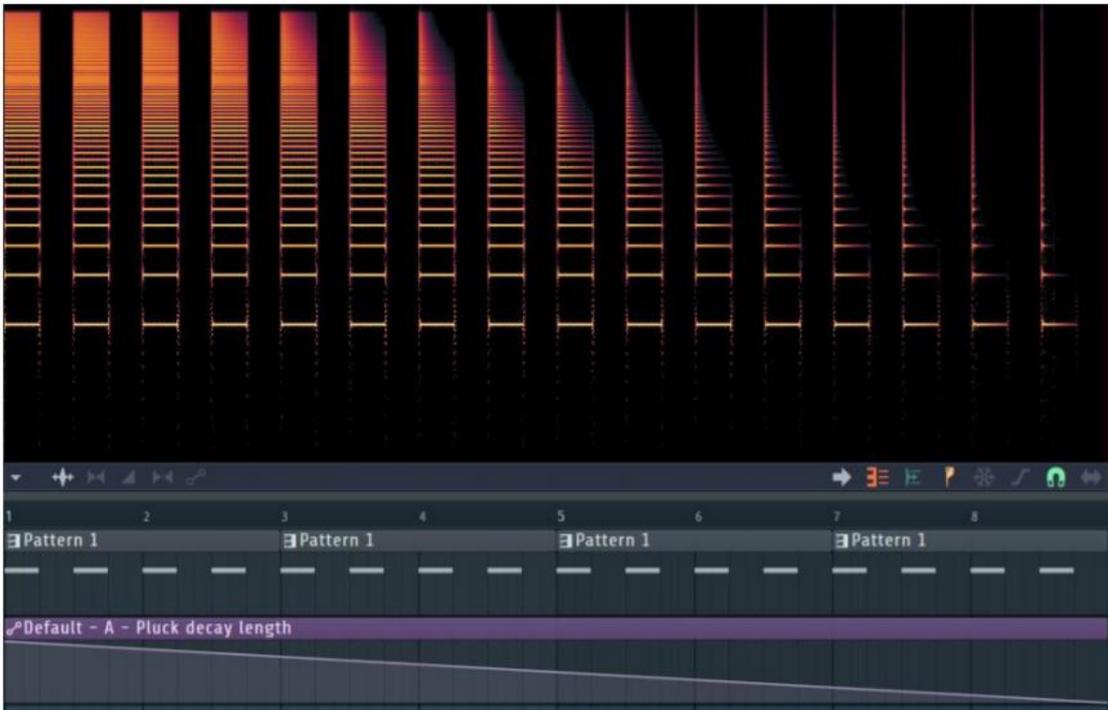
### 弹拨效果 (Pluck)

弹拨效果会创造一种在弦类乐器上弹拨时产生的音色。背后的原理是在一开始的时候激活所有需要的小振荡器，然后开始从高频衰减到低频。



## 弹拨效果量 (Pluck amount)

声音的衰减速度由弹拨衰减时长 (Pluck decay length), TIME, 也叫作弹拨效果量 (pluck amount), 控制。默认设定下是 100%, 当衰减速度最快时是 0%。由于基频在整个效果中是一直被激活的, 所以音调一直是稳定的。值得注意的是, 当这个效果逐渐移除所有除了基频之外的小振荡器时, 不论我们开始时的波形是什么, 最终我们都会只剩下正弦波, 除非声音被扭曲, 比如说电吉他。最终会变为一个正弦波的这个结果也是为什么弹拨效果会这么明显。我们会在弹拨形状 (Pluck shape) 相关的章节中讨论更多关于弹拨效果的相关内容, 那时我们将会看到, 我们还可以改变弹拨效果的包络线形状。在示范工程 <https://www.malmgren.nl/Harmor/Harmor-19.flp> 中, 我将弹拨效果附加在了锯齿波上, 并且逐渐减少弹拨效果的衰减时长。



直到 45% 的时候, 弹拨效果才开始变得明显起来。

## 弹拨效果代替模糊衰减 (Pluck replaces blur decay)



在 TIME 旋钮旁边有一个模糊 (blur) 按钮。当你打开这个按钮的时候, 我们就把正常的弹拨形状公式 (pluck shape formula) 换成了模糊衰减公式 (blur decay formula)。

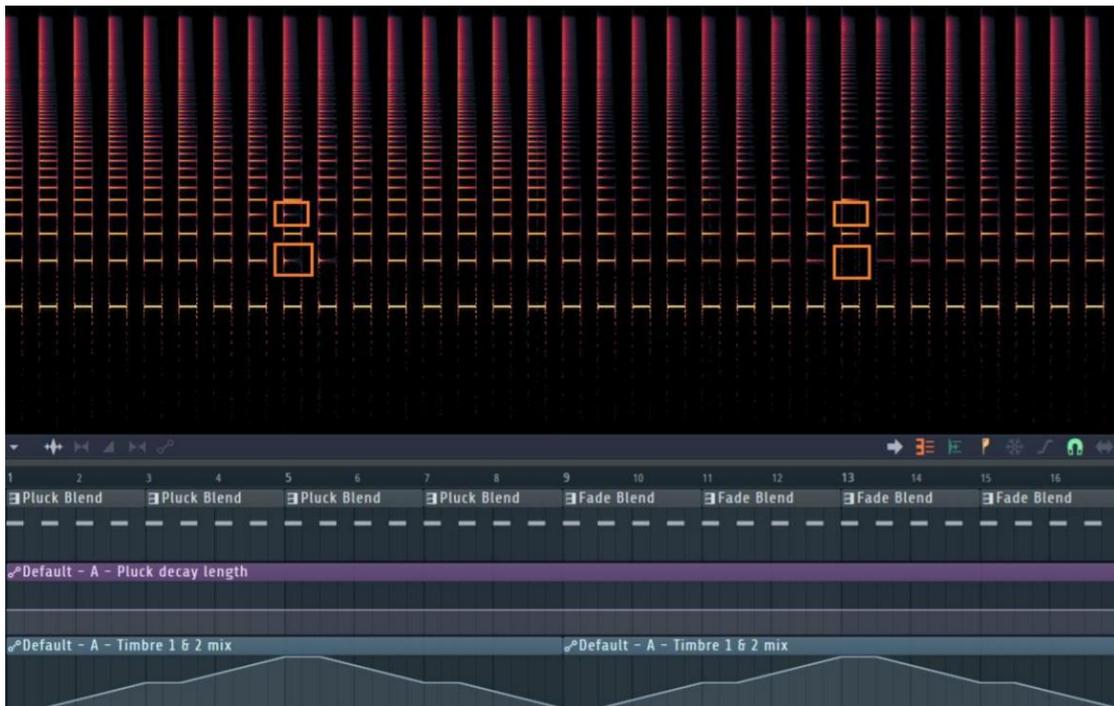


我们可以看到时间模糊衰减周围的绿色光环变的暗淡了, 而且正常的弹拨形状算法也被关闭了。为了研究时间模糊衰减旋钮, 我做了许多不同的自动化包络线, 但是我听不到, 也看不到任何差别。

## 音色混合模式：弹拨效果（Timbre blending mode: Pluck）

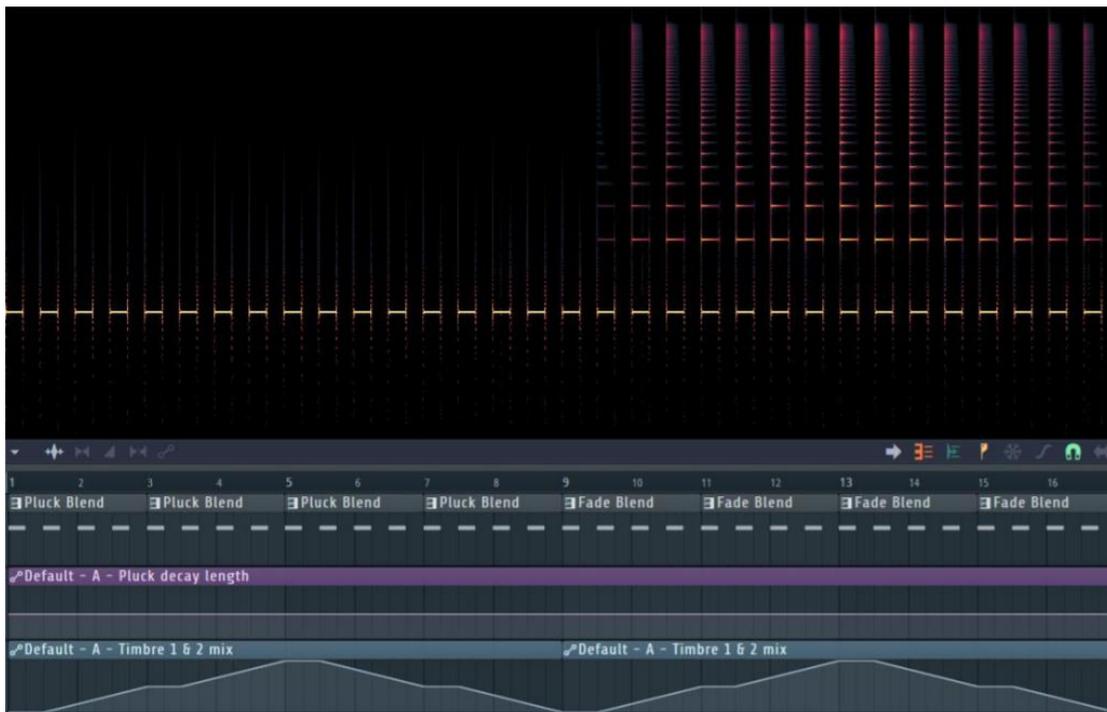
我们在音色混合模式（Timbre blending modes）的部分讨论了弹拨效果。在那个部分我们留下了一个问题，音色混合模式的弹拨效果到底是干什么的？

在示范工程 <https://www.malmgren.nl/Harmor/Harmor-20.flp> 中，你将会看到我制作了两个模式。弹拨混合（Pluck Blend），以及渐逝混合（Fade Blend）。



在“Timbre1 & 2 mix”的自动化包络线中，我逐渐在两个音色之间进行切换。两者都是默认的锯齿波以及方波，总的来说我们发现渐逝 (Fade) 模式中的小振荡器没有那么活跃。在 100% 的音色 2 中我们可以才能看到正在发生的事情：弹拨混合模式使得小振荡器在一开始的时候十分活跃。这也是为什么只有在音符最开始的时候两个小振荡器窗口才能够展示两个锯齿波。

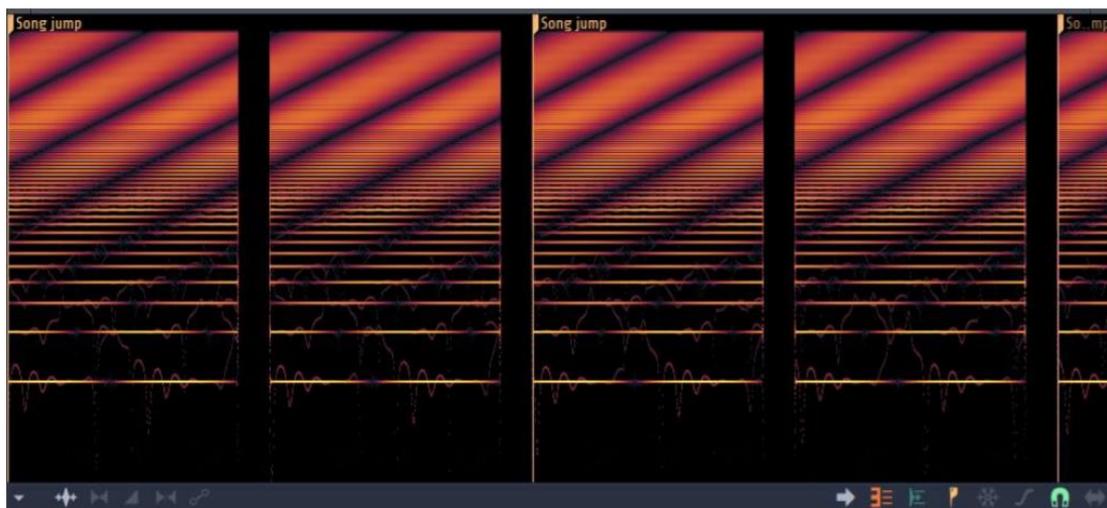
如果没有除基音以外的小振荡器的话，弹拨混合模式将不会有任何效果。为了测试这一点，我创建了示范工程：<https://www.malmgren.nl/Harmor/Harmor-21.flp>。



我这里有一个弹拨模式中的小技巧！你可以尝试右键音色窗口，然后选择随机。这会给你随机生成小振荡器的各种组合，而且会使你的弹拨音色听起来非常独特。

## 移相器（Phaser）

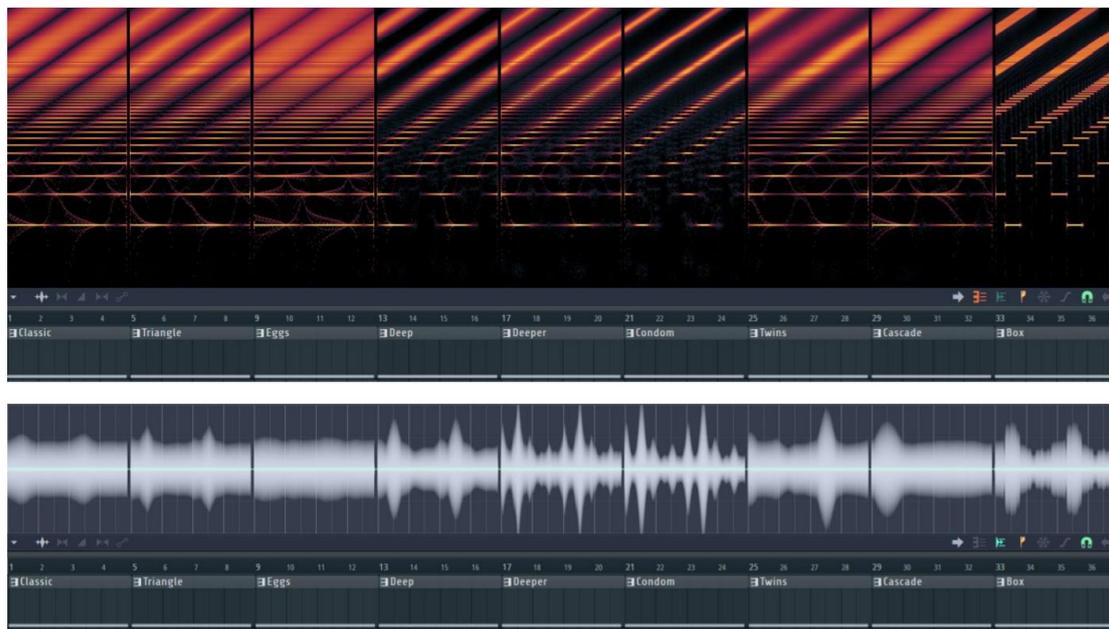
在讲过滤器的那一章，我们讨论了一个叫做移相器的过滤器的形状。这里的移相器效果是一个我们在过滤器那里见到的移相器的一个变体。过滤器中的移相器并不包含时间这个变量，其效果不会随着时间改变。然而这里的移相器效果却将其过滤器效果与时间相联系。我们可以通过试验来看一看。



我们可以看到这个效果可以在特定频率上产生一个抵消的效果。整个模式是不断运动着的，而且其抵消效果是与电脑的时钟相关的。我们可以看到，即使我们没有弹奏音符，这个运动效果依然是持续的

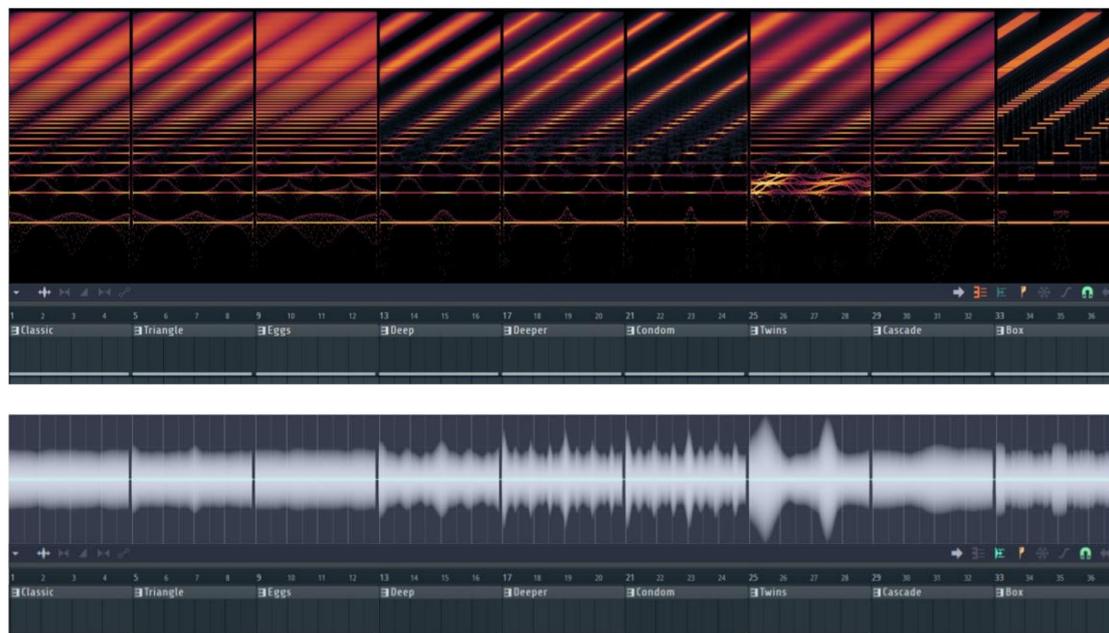
## 移相器种类 (Phaser type)

在示范工程 <https://www.malmgren.nl/Harmor/Harmor-23.flp> 中，我们可以看到除 freq 以及 custom 之外所有的移相器类型。我们会在移相器形状 (phaser shape) 的相关章节中讨论这些自定义的移相器种类。



Harmor 的官方文档把前三个移相器种类称为传统种类。当我们转换到 Edison 来展示波形的时候，我们可以清楚地看到在整个移相器的样式 (pattern) 中振幅是怎样的改变的。特别是在当基频被压抑的时候，音量变低了。你也许还能回忆起在第四章我们讲到，我们是可以保护基频的。

在示范工程 <https://www.malmgren.nl/Harmor/Harmor-24.flp> 中，你可尝试尝试保护基频。



看起来在 twins 模式中，保护基音的操作效果不是很好。但是剩下的都在稳定振幅以及基频方面都做得不错。

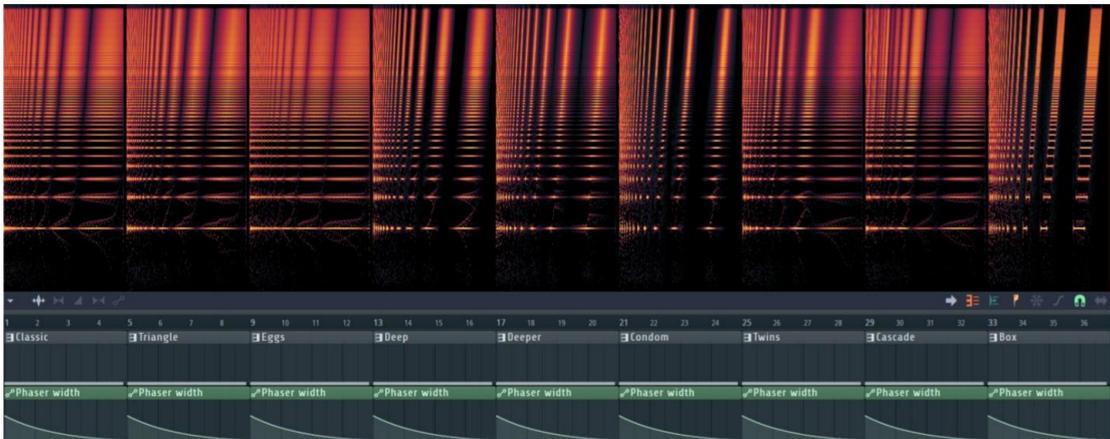
## 移相器混合 (Phaser mix)



MIX 旋钮，也就是移相器混合旋钮，可以使我们能够调整移相器效果的强弱。

## 移相器宽度 (Phaser width)

移相器宽度的工作原理与过滤器宽度 (filter width) 类似



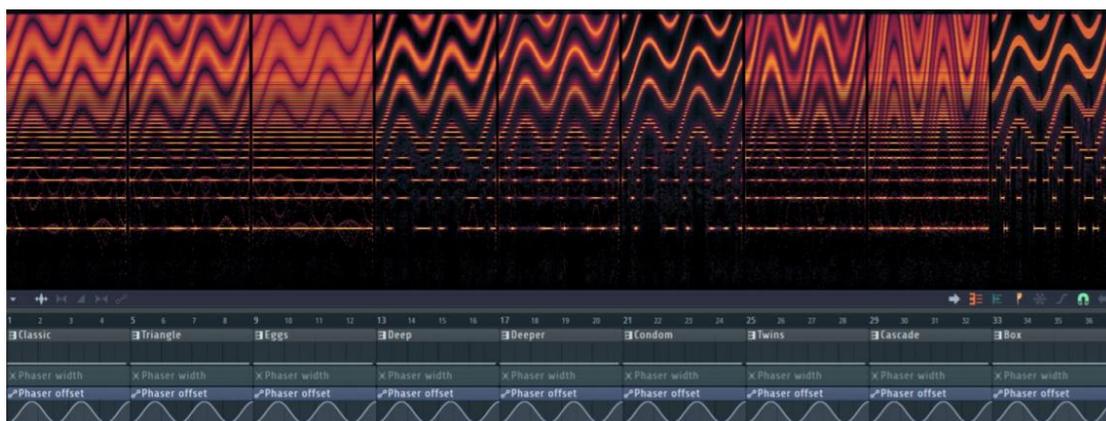
我创建了一个自动化包络线，从 62% 一路降到 0%。62% 以上的值所对应的音在图中的分布十分紧凑，所以基本上在截屏中根本看不见，但是你可以打开 <https://www.malmgren.nl/Harmor/Harmor-25.flp> 来自己看一看。你也可以在这个示范工程中调整各种参数，把玩一下。

请看一下这种像瀑布一样的形状。在这种结构下，你可以看到每个条状的移相器效果都在逐步的被放大。

## 移相器偏移 (Phaser offset)



正常来讲，移相器效果的起始点是基于电脑时钟来设置的。当然我们也可以相对于电脑时钟来对这个效果进行偏移。这可以通过移相器偏移旋钮 ofs 来操作。

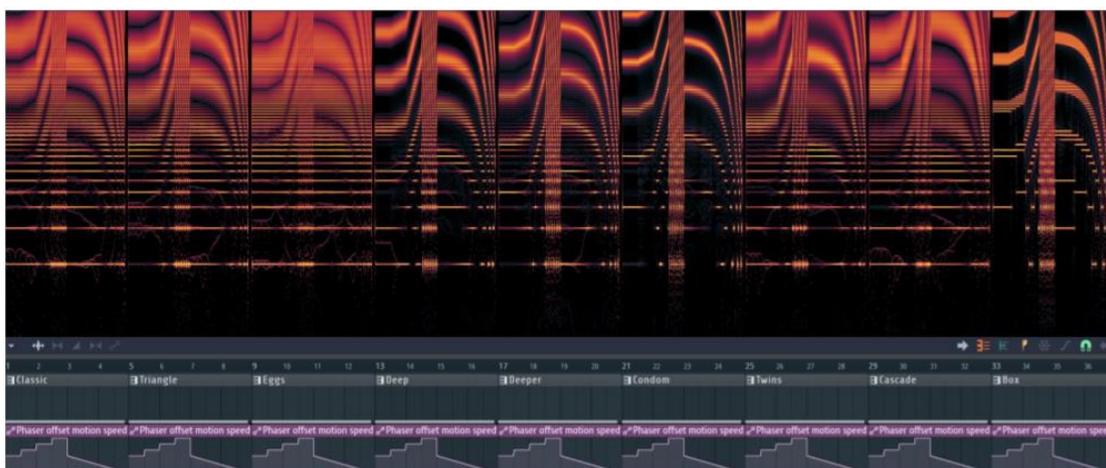


在示范工程 <https://www.malmgren.nl/Harmor/Harmor-26.flp> 中，我们对移相器偏移这个效果进行了研究。

## 移相器偏移速度 (Phaser offset motion speed)

Speed 旋钮是用来设定移相器效果在频谱中移动的速度的。默认值是 32:00。当我们将旋钮拧到右侧的时候，最小值是 1:00，拧到最顶端的时候，其值显示的是“still”。在显示这个“still”之前，该旋钮的值以指数形式增加。在另外一侧这些值是一样的。最小值是你把旋钮拧到最左侧的时候，也就是 1:00。

为了做关于偏移速度的试验，我决定将该旋钮自动化，然后尝试不同的值。你可以下载示范工程：<https://www.malmgren.nl/Harmor/Harmor-27.flp>。

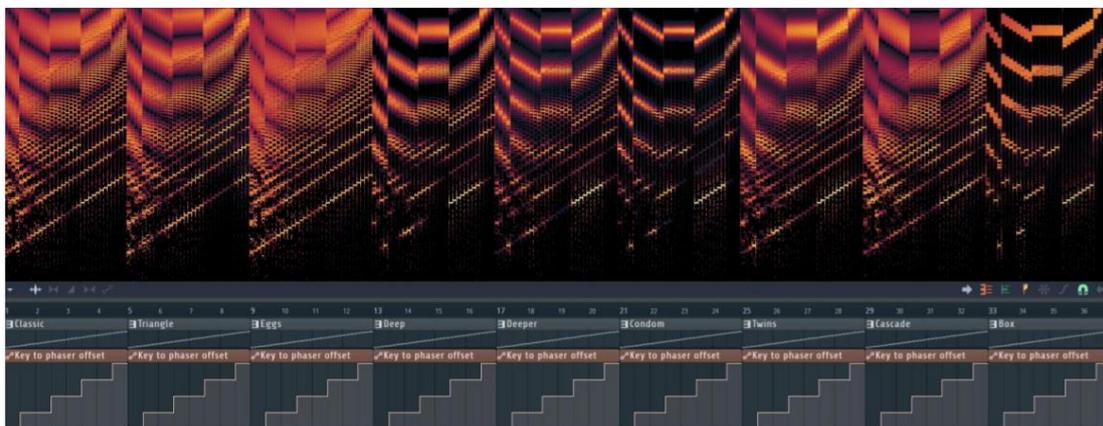


如你所见，在移相器偏移速度的曲线上，自动化包络线从 50% 开始。然后在三步之后我将旋钮拧到了右侧的最小值。然后我又回到了 50%，然后慢慢减小，但是这次是以一个斜线下降的形式。

我们在这里自动化的是相位偏移的速度。所以当我们的自动化包络线的值靠近 still 的时候，相位偏移的速度十分缓慢。当其值靠近最小值的时候，相位偏移的速度则变得十分迅速。当旋钮在右侧的时候，相位模式是向上走，在左侧的时候，相位模式是向下走的。

## 从琴键到相位器偏移 (Key to phaser offset)

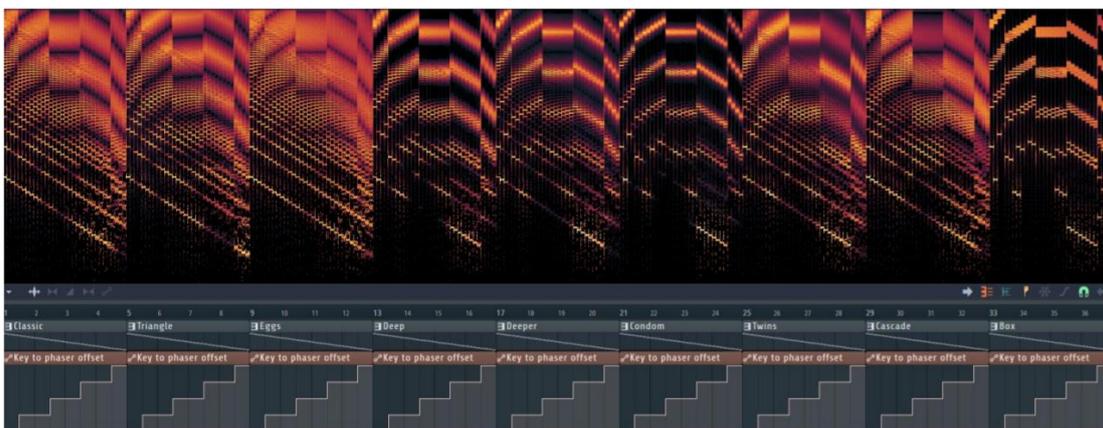
我们当然也可以将移相器时间偏移 (phaser time offset) 与基频联系起来! 相位偏移的键盘追踪 (keyboard tracking) 系统与我们之前在过滤器频率 (filter frequency) 部分讨论过的键盘追踪系统非常相似。当我们在对这个进行试验的时候, 我决定关闭速度, 把剩下所有的设置保持不变。这样的话基频就会成为相位偏移的唯一变量了。



在示范工程 <https://www.malmgren.nl/Harmor/Harmor-28.flp> 中, 里面的模式包含所有从 B2 一直到 F#5 的音。然后我为相位偏移旋钮创建了自动化包络线。自动化包络线从 0% 开始, 这里的 0% 对应的是琴键追踪旋钮的 -200。下一步是 25%, 对应着 -100, 然后是 50%, 对应着 0, 然后是 75%, 对应着 100, 最后是 100%, 对应着 200。

图像的斜率看起来像 key to filter frequency 图像的斜率。当键盘追踪旋钮的值是 0 的时候, 我们就得到了一个水平线。如果我们将速度这个变量也加进去——当然我们现在并没有——那么我们会看到速度的影响。在 100% 的时候, 图像的斜线十分完美地切合了音的斜率。在 200 的时候, 图像的斜率更加陡峭。当琴键追踪系统是负值的时候, 情况与是正值的时候完全相反。

如果这个理论是正确的话, 我们反过来从 F#5 开始弹奏, 一直到 B2, 我们会得到完全相反的节奏图像! 幸运的是, 我是正确的。详见 <https://www.malmgren.nl/Harmor/Harmor-29.flp>。



## 移相器比例 (Phaser scale)

Oct 模式使得移相器模式以对数的形式铺展开。在 Hz 模式中，移相器模式则以线性的形式展开。谐音模式 (Harmonic mode) (译者注：在 Harmor 的用户界面中的缩写是 harm) 在某些方面与 Hz 模式相似，例如谐音模式制造出来的样式 (pattern) 是线性的，但是有些地方既不像 oct 模式，也不像 Hz 模式，如果你想让谐音模式制造出类似于 oct 模式，以及 Hz 模式的模式的话，那就请把其宽度 (width) 设置成 2%到 12%。

## 第八章棱镜 (Prism)

棱镜 (prism) 会将小振荡器的位置从原来的位置进行偏移，其偏移基于 Harmor 中棱镜部分的设置。棱镜效果不会造成基频的偏移。因此当你的音色是一个正弦波的时候，加入棱镜不会有任何效果。

### 棱镜效果大小 (Prism amount)

这里有一个示范工程：<https://www.malmgren.nl/Harmor/Harmor-30.flp>。在这个工程中我假如了一个锯齿波和方波的混合音色。在其中我也对两者混合量创建了自动化包络线。我弹奏了下面的几个音：C3, F3, C4, F4, C5, G5, C6, E6。

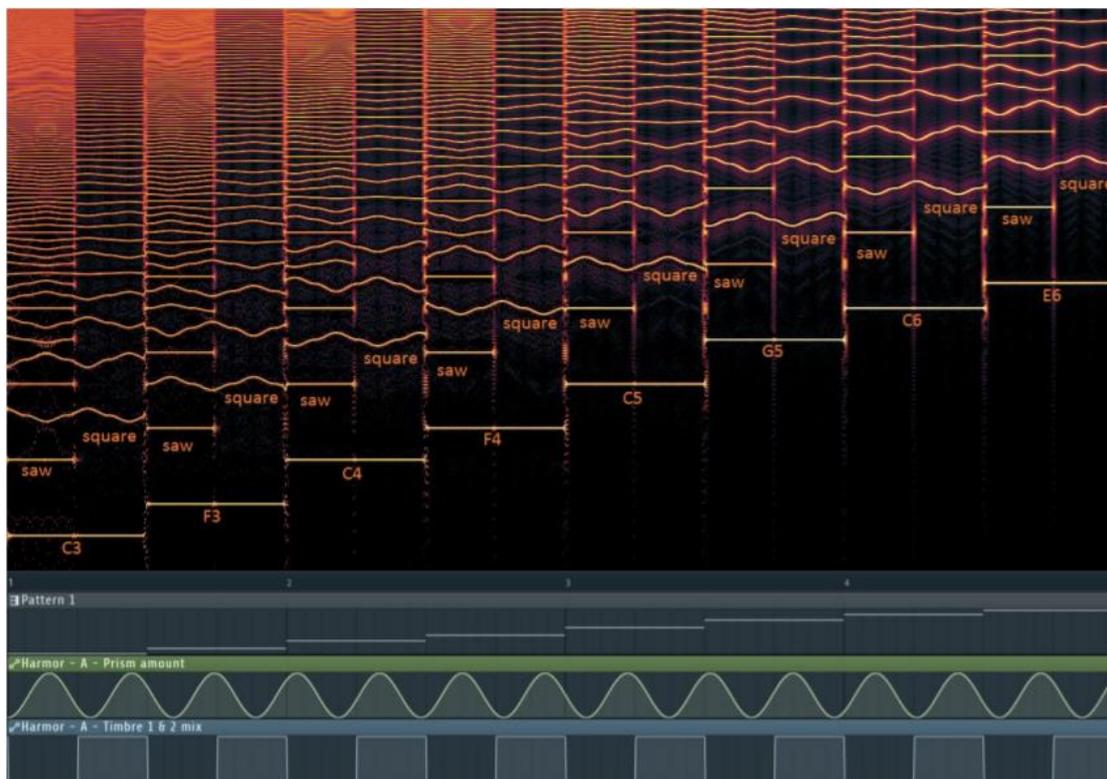


图 39. 自然音阶中锯齿波和方波的棱镜效果。

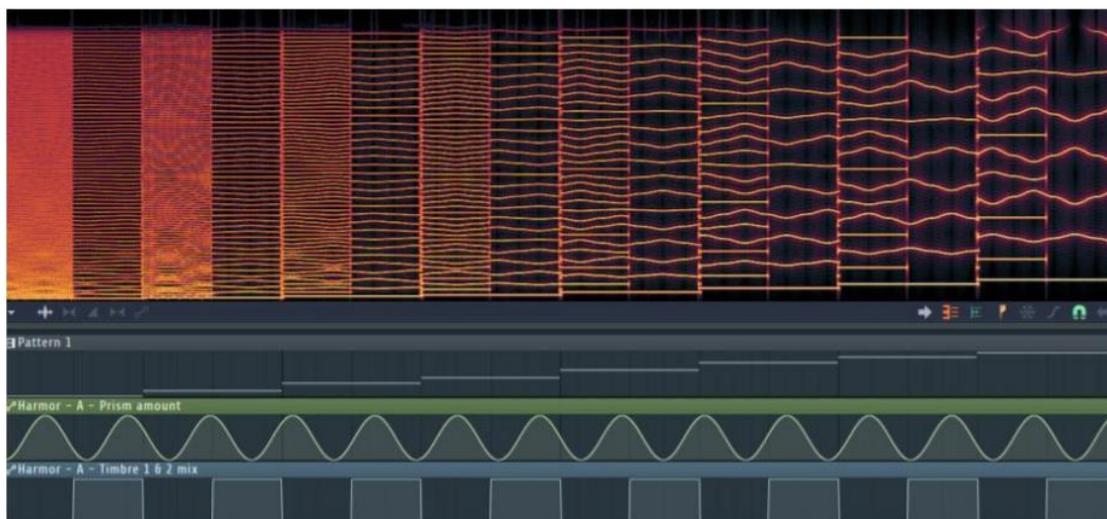
棱镜绝不会改变基频。当一个波形是由所有 516 个小振荡器共同振荡组成的时，我们也

会发现并不是所有的小振荡器都被偏移了。其偏移量取决于棱镜效果大小旋钮。只有频率处在八度音之间的小振荡器才会被偏移。当我们播放一个方波的时候，并不是所有的小振荡器被激活。基本原理依然相同，只不过方波只有基频上的音，而不像锯齿波那样有一整串八度音。

在这个试验中，我自动化了棱镜效果大小。这意味着当自动化包络线的曲线通过 0 的时候，就不会产生任何偏移。

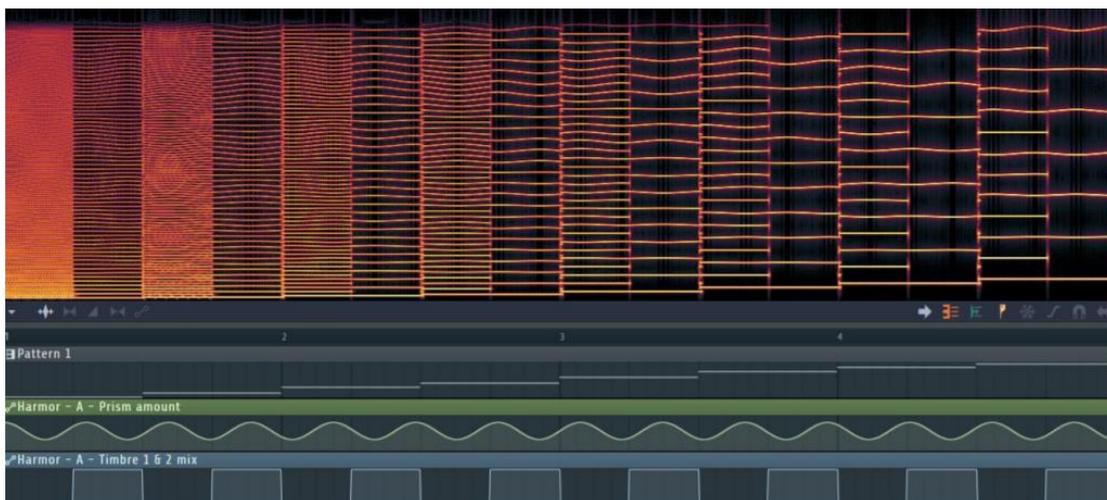
## 棱镜相加/相乘模式 (Prism add/multiply mode)

在默认设置下，棱镜处于相加模式 (add mode)，这也是我们目前为止一直在试验的模式。在相加模式下，基频附近的偏移量是最大的，并且随着频率的增加，偏移量将会越来越小。在另外一方面，即使从图 39 看来是这样的，但是由于整个频谱是以对数形式排列的，因此在频率很高的地方，即使看起来偏移量很小，但其实影响却更大。在 Edison 中我们可以关闭对数坐标轴，当我们这么做之后，我们就可以看到在对数坐标下看不到的细节了。

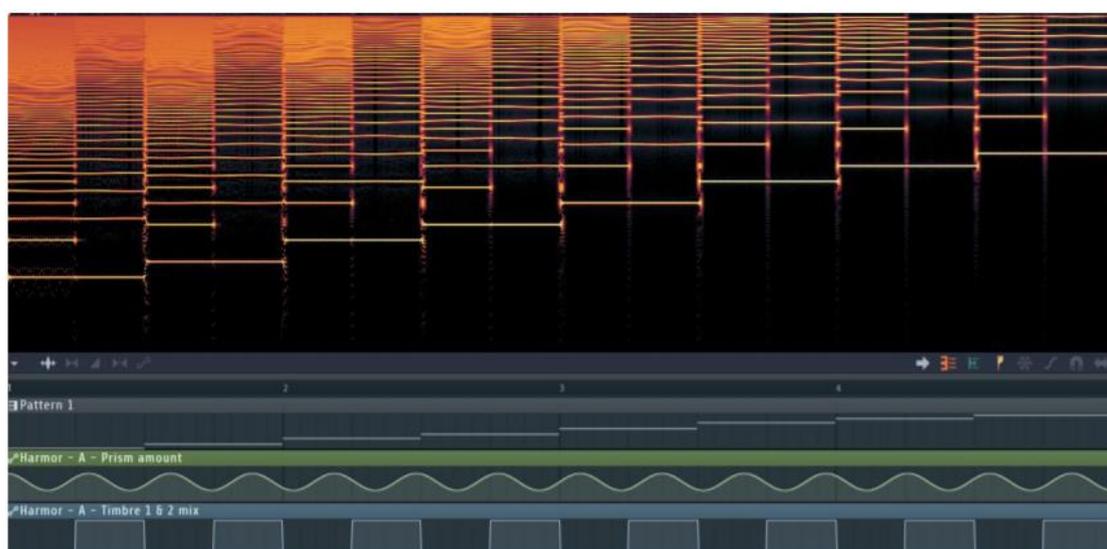


在关闭对数坐标轴之后，我们可以看到相加模式其实是对所有的小振荡器加成了同样多的振幅。现在是时候换成相乘模式 (multiply mode) 了，在这种模式下我们发现基频附近的小振荡器没有偏移。在这个试验中 <https://www.malmgren.nl/Harmor/Harmor-31.flp>，我们大量地减少了正弦波自动化包络线中的棱镜效果。我们之所以这么做是为了防止棱镜效果太大，使得小振荡器的频率跳到图像之外。

随着频率增大，我们可以看到越来越多的偏移效果。



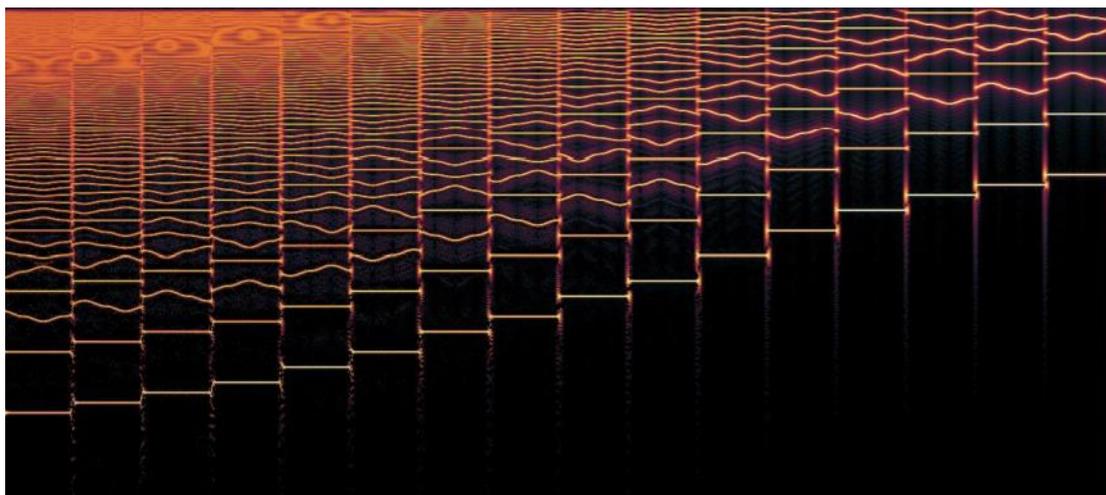
如果我们使用对数坐标系的话，我们基本上看不到高频区域内的变化。



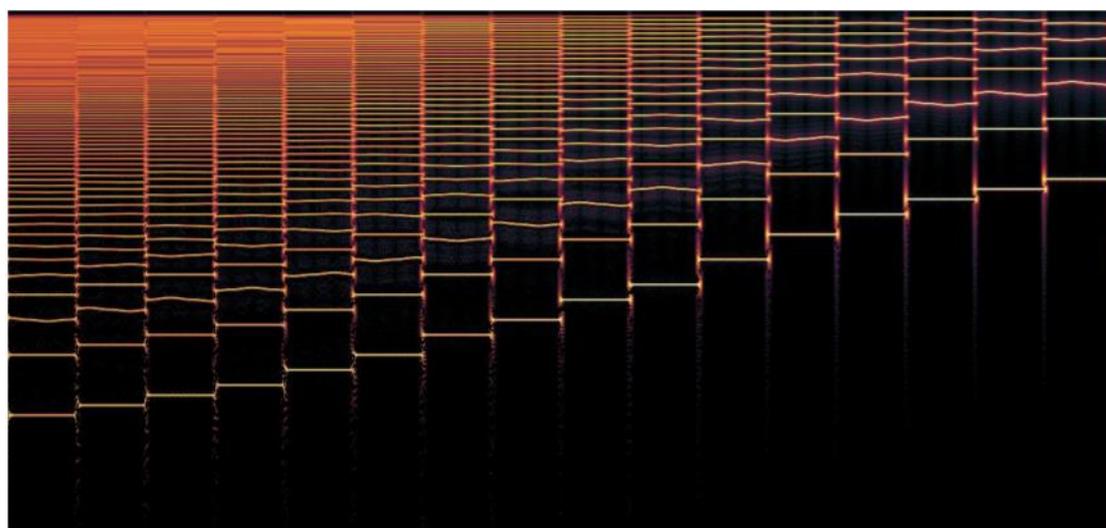
## 谐音水平到棱镜效果大小 (Harmonic level to prism amount)

下面是一张当谐音水平-棱镜效果大小开启的图。在这里我只试验了锯齿波。另外在这个试验里，我分别多播放了一个非常低的音，和一个非常高的音。

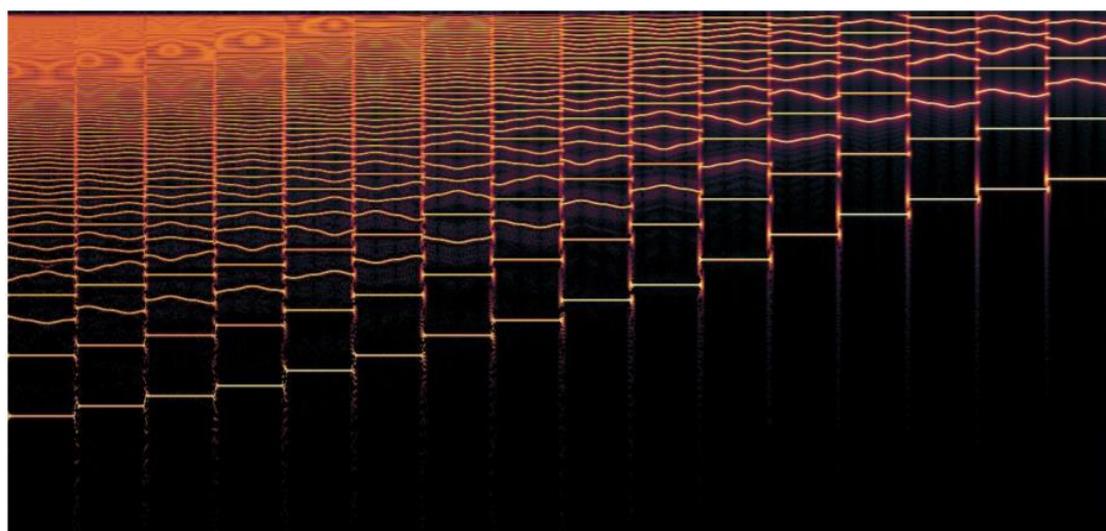
<https://www.malmgren.nl/Harmor/Harmor-32.flp>。



这里我设置的是谐音水平-棱镜效果 (Harmonic level to prism amount: on) 大小开启



每个小振荡器距离基频的距离决定了棱镜偏移在其身上的作用大小。我们发现所有不同音的第三个小振荡器的偏移量是相同的。



在这里我们的设置是谐音水平-棱镜效果大小反向 (harmonic level to prism amount: inverted)。在这个设定下，小振荡器距离基频越远，其受到的棱镜偏移量就越大。

(译者注：这几个设置位于 Prism 区域中 from vol 的下面，如果你在 Harmor 中找不到某个旋钮或者设置的话，可以把光标移动到旋钮上，然后左上角的工具提示框中就会出现这个旋钮的名字)

## 第九章 . 谐音器 (Harmonizer)

谐音器是用来增强某些小振荡器，并抑制其他小振荡器的一个效果器。首先基频附近的谐音会被复制，接着这些被复制的谐音会根据控制台上的设置，被算法粘贴到其他的频率位置。



### 谐音器混合 (Harmonizer mix)

谐音器混合，AMT，控制的是有多少谐音器效果会附加在小振荡器上。

### 谐音器宽度 (Harmonizer width)

谐音器宽度，WIDTH，控制的是从基音的频率开始，一直到多高会被谐音器所影响。谐音器强度 (Harmonizer strength)，str，控制的是谐音器效果的大小。AMT 和 str 看起来好像是重复的，但是当 str 是 0，并且调整 AMT 的时候，依然有效果；然而当 AMT 是 0 时，不论怎样调整 str，都不会有任何效果。

最后，我们来看看谐音器模式控制器。这是一个 2 乘 2，只有一个数位的转轮。转轮之间有一个运算符号。运算符号可以是加号 '+'，也可以是乘号 'x'。当转动其中一个黄色的转轮，ofs (offset)，时你可以将它设置成任何一个 0 到 9 的数字。当滚动白色的转轮，step，时，那么可以设置为 1 到 8 之间任意的一个数。

探索这个转轮的作用的最好方式是将 AMT，WIDTH 以及 str 设置到其最大值。然后将两个 shift 和 gap 都设置成 0x1。这样的话，谐音器会以最大功率工作。

接下来我们就可以将转轮的算法分离开了。我们可以演奏一个音，比如锯齿波，然后从视觉反馈窗口去观察。你试验的时候可以用我们在第二章讲过的方法，将视觉反馈窗口变宽。另一个有用的小技巧是将视觉反馈窗口改成 Hz 模式。如果我们想数小振荡器的个数的话，是十分方便的。

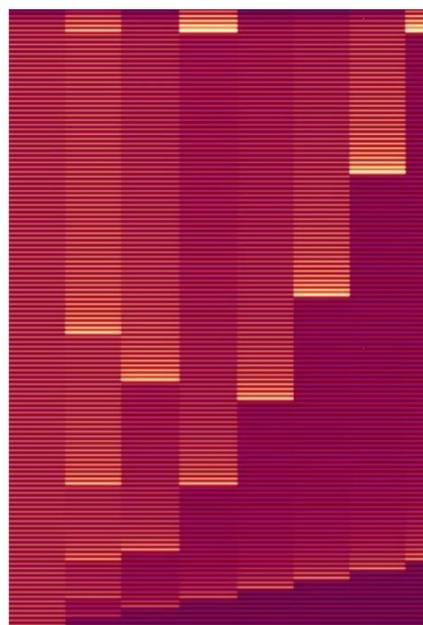


图 40. 在乘法模式中，当 step 是 1，2，3...8 时的情况

在谐音器复制小振荡器之前，你得先保证在谐音器中有足够的小振荡器。正弦波的小振荡器太少，因此在正弦波上使用谐音器不会有任何的作用。

这一部分的所有截屏皆来自这个示范工程：<https://www.malmgren.nl/Harmor/Harmor-33.flp>

## 谐音器的 shift 和 step 参数 (Harmonizer Shift Step)

当 gap 被设置为 0x1 时，请看图 40，我们会在 shift 选项上进行试验。结果就是在 'x' 的模式下，偏移效果将从对编号是  $ofs + 1 + (step-1) \times 2$  的小振荡器进行增强开始。

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.

然后，下一个会被增强的小振荡器的编号是前一个小振荡器的编号，减去 1，乘以 step 参数。然后以此类推。

加法模式 (plus mode)，请看图 41，会更容易理解一些。不像 0x1 那样，谐音器效果会在当 step 是 1 的时候被关闭，在这里当 0+1 时，其效果在 step 等于 1 的时候也清晰可见。这个效果的基础小振荡器是 step 参数的值。也就是说，在当 step 等于 5 的时候，基础小振荡器是第五个小振荡器，然后以此类推。这会被重复将近 7 次。

## 谐音器的 gap 和 step 参数 (Harmonizer Gap Step)

谐音器的 gap 和 step 参数模式算法十分有趣。这种算法不仅仅会加强某些特定的小振荡器，同时也会降低一些小振荡器的振幅。

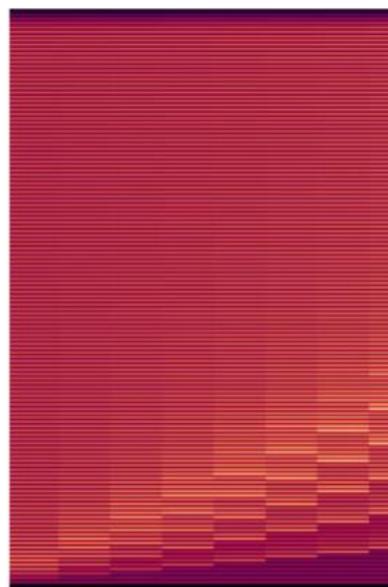


图 41. 谐音器在 '+' 模式下的偏移模式。

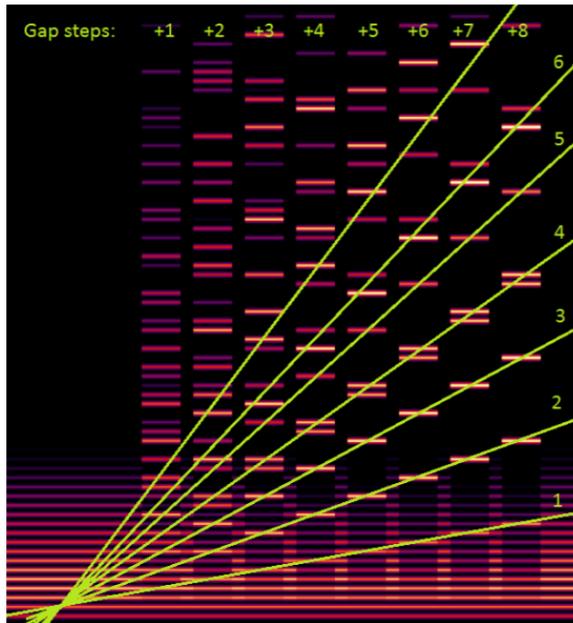


图 42. 这是谐波器中 gap-step 的 '+' 算法。这个图片是从 Harmor 的视觉反馈窗口中截取的，其坐标模式是 Hz 模式。

如果我们觉得 shift-step 参数组合有一个重复出现的模式的话，那么 gap-step 参数组合的模式就算是更有意思。从图 42 中，我们可以发现，最靠近底部的三根绿线划过的区域只有一个小振荡器被增强了。第四根线划过的区域则有两个小振荡器被增强。第五根线又变为了单独一个小振荡器。然后从第六根线开始，看起来大部分的线经过的区域都是由两个小振荡器被激活，并且当频率增高时，他们被激活的小振荡器之间的距离也更宽了。

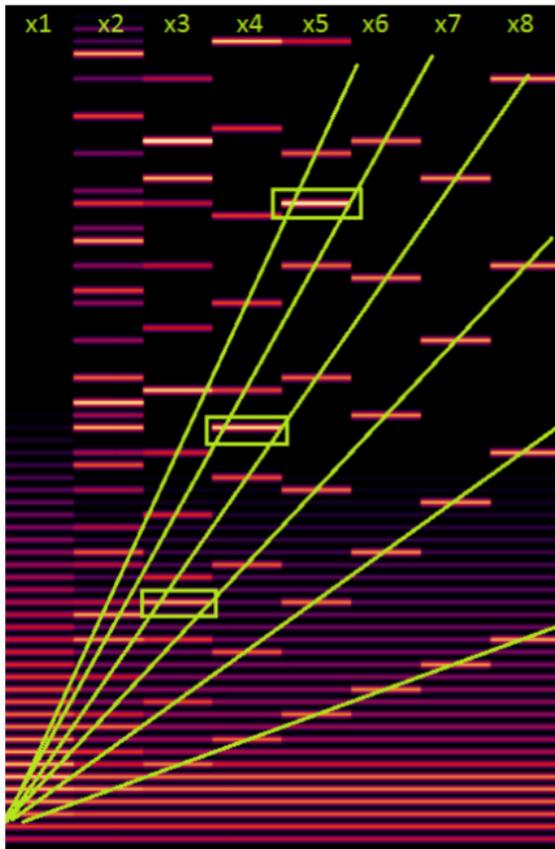


图 43. 'x'模式中 gap-step 的参数组合结果图。

'x'模式的 gap-step 参数组合效果在第  $\text{ofs} + 3 + (\text{step}-1) \times 2$  个小振荡器开始，这里的 step 必须是一个大于等于 2 的数。然后下一个小振荡器就是第  $1 + (\text{step}-2) \times 2$  个小振荡器，这里的 step 必须是个大于等于 3 的数。当 step 被设置成 2 的时候，其模式每 3 步被重复一次。请注意图中绿色长方形中的小振荡器，这些小振荡器是被插入在第 step 个小振荡器的上方。例如，在当 step 等于 4 的时候，在第 4 个被增强的小振荡器的上方就会多出来一个被增强的小振荡器。当 step 等于 5 的时候，在第 5 个被增强的小振荡器的上方就会多出来一个被增强的小振荡器，以此类推。

## 第十章 . 颤音 (Tremolo)

Tremolo 是一个意大利单词，颤音的意思。许多音乐上的专业词语都是源于意大利语，因为意大利在早期的音乐商业中十分重要。Harmor 中的 Tremolo 部分是用来模拟电子管乐上的扩音器。在 1937 年下半年，唐纳德莱斯利发明了这些扩音器。如果你想知道更多故事的话，请自行百度/wiki。

颤音音效背后的原理是改变左右声道的振幅以及相位。因为颤音效果是一种立体声，通过 WaveCandy 来将声音视觉反馈可以很好地帮助我们理解颤音音效。我创建了一个工程，你们可以用这个对颤音音效进行试验：<https://www.malmgren.nl/Harmor/Harmor-34.flp>。(译者注：Tremolo 在 Harmor 中的缩写是 TREM)

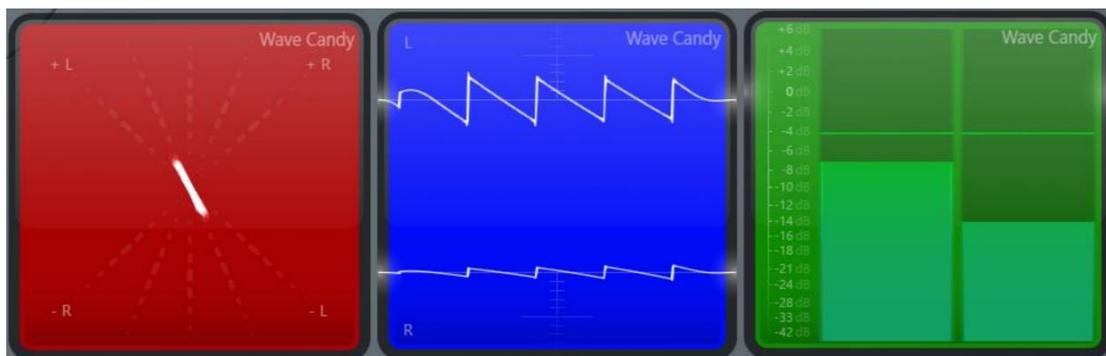


图 44. 这里我们播放一个锯齿波，然后用三个不同的 WaveCandy 视觉反馈窗口来展示。红色=矢量显示器，蓝色=示波器，绿色=音量表。

在图 44 中，当我们播放一个锯齿波的时候，我们可以看到一条线。如果我们播放一个方波的话，我们会看到两个点。白线的重点就是图像的重点。如果我们播放一个单声道的锯齿波的话，我们只会看到一条垂直的线。

## 颤音速度 (Tremolo speed)

颤音的速度，speed，控制的是终点循环 (rotation of the end points) 的速度。最小值是 16 ms，最大值是 16 s。如果你将改参数调节至 16 ms，矢量显示器就会将屏幕上的点移动的飞快，以至于你能够看到这些声信号是如何被加工的，特别是当你播放一个方波的时候，因为方波的信号在终点呆的时间最长。

## 颤音间隔 (Tremolo gap)

颤音的间隔 (gap) 参数设置的是声音信号会以怎样的方式运动。当间隔在 0 到 180 之间时，矢量显示器的上半部分就会沿着顺时针的方向移动。

当间隔在 180 到 360 之间时，上半部分顺着顺时针的方向移动。当间隔是 0 的时候，不会有任何的颤音效果，此时声信号是单声道。



图 45 当颤音间隔参数靠近 0 的时候

当间隔参数靠近 0 的时候，矢量显示器中的形状是一个水滴的形状。当间隔靠近 180 的时候，则是一个微笑的形状。

## 颤音深度 (Tremolo depth)

颤音的深度旋钮控制的是有多少颤音效果会被附加到原声上。当该参数是 0% 的时候，声信号会停留在图像中的最高以及最低点。当深度的大小逐渐变大的时候，其形状也会逐渐变大。

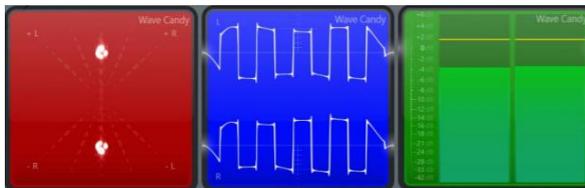
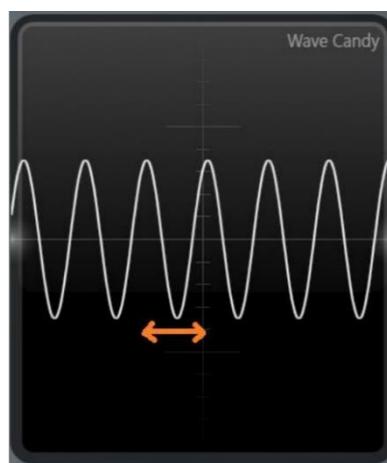


图 47. 当颤音的深度参数是 17% 的时候。

## 第十一章 . 颤动 (Vibrato)

Vibrato 是一个意大利词语，有颤动的意思。

Vibrato 效果通过向前向后移动起始相位来制造颤动的感觉。示范工程 <https://www.malmgren.nl/Harmor/Harmor-35.flp> 展示了一个 WaveCandy 的示波器图像，你可以在上面看到波形在前后移动。



### Vibrato 音调深度 (Pitch vibrato depth)

当 vibrato 的音调深度，depth，被调至左侧的时候，音效是关闭着的。右侧是 depth 的最大值，100%。Depth 旋钮控制的是在 vibrato 效果强弱，波形前后移动的幅度大小。

### Vibrato 音调速度 (Pitch vibrato speed)

Vibrato 的音调速度旋钮，speed，控制的是前后移动的速度有多快。当旋钮被拧到最左侧的时候，是最快的，每个前后移动的周期只有 125ms。当被拧到最右侧的时候，周期最长，250ms。

## 第十二章 Legato 和 Portamento

Legato 是一个意大利的音乐术语，意思是“捆绑到一起”。在合成器中的意思是，从一个音滑到另一个音，这两个音并不需要同时被演奏。

Portamento 也是一个从意大利语中来的词，意思是“持续”。在合成器中的意思是也是从一个音滑到另一个音，然而与 Legato 不同，这两个音不需要同时被演奏。

这两个 Harmor 的功能，legato 以及 Portamento 在 Harmor 的两个不同部分中工作：12a 以及 12b。在这两个部分中，有很多东西容易混淆，因此我决定在开始这一章的时候，仅仅只讨论 Legato，然后再回过头来讲一讲 Portamento。



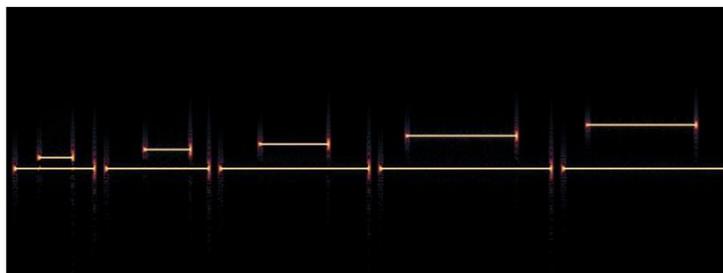
直到现在为止，我们一直讨论的是 Harmor 上半部分，棕色部分，中的东西。从这一章开始，我们讨论的将会是，在小振荡器被加和在一起，成为声信号之后发生的事情，也就是 Harmor 的下半部分，黑色的部分。

## Legato

为了便于我解释 Legato，我制作了一个示范工程，在示范工程中，我用正弦波在 C 大调中写了一个小小的旋律：  
<https://www.malmgren.nl/Harmor/Harmor-36.flp>



在 Edison 中，看起来是这样的：



## Legato 模式开关



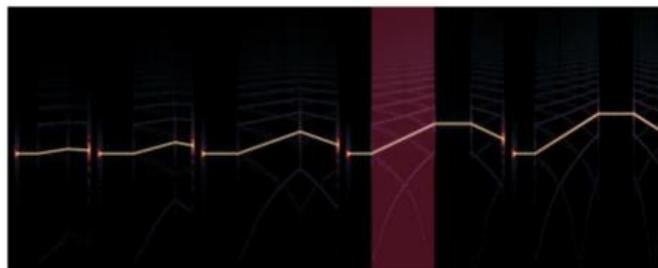
在 Legato 模式开关开启，并且有两个音同时被弹奏的时候，就会出现滑音的效果。当第二个音停止演奏的时候，就会重新滑回去。



## Legato 时间



Portamento 以及 legato 时间, time, 控制的是滑音的音长。旋钮左侧的单位是毫秒。最长的滑音是当旋钮被旋转 to 最左侧 1500ms 的时候。最短的滑音是当旋钮在最上边的时候。图片当中的滑音时长是 643ms。我们可以将 Edison 设置成按照毫秒播放。红色的区域是 643ms。

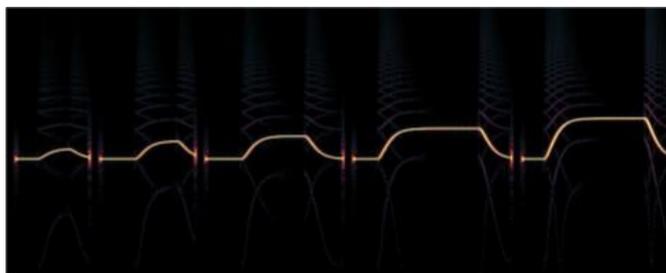


旋钮的右侧是根据整个工程的时长来控制滑音的时长的。

## 线性/对数滑音 (Linear/Logarithmic legato)



到目前为止，上面所有提到的例子都是线性 legato。通过点击 LEGATO 下面的按钮，我们可以在线性滑音以及对数滑音之间进行切换。

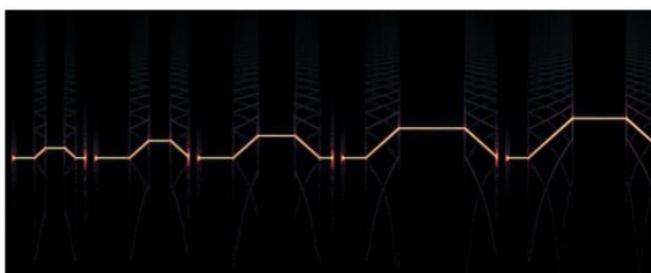


线性滑音指的是音与音之间的过度是线性的，而对数滑音中，音与音之间的过度则是一条曲线。

## 固定/可变滑音时长 (Fixed/variable legato time)



我们可以让滑音过度的时长变为音程 (distance between the notes) 的函数，这被称为可变滑音时长 (variable legato time)。当最右边的图标被点亮时候，Harmor 就会开启可变滑音时长模式。



这时，当音程增加，过度时间也会增加。

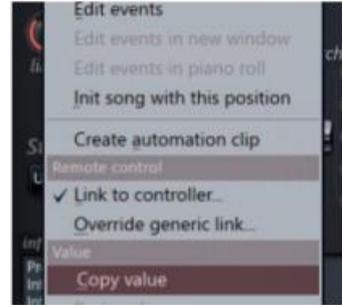
使用线性滑音模式更容易展示这种效果，因为过度过程的结束更加明显。当左侧图标被点亮的时候，Harmor 就会让过度时长保持一致，不管音与音之间的距离到底有多远。

## 滑音限制 (Legato limit)

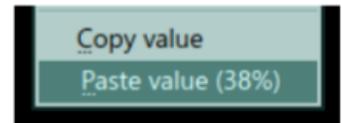


滑音限制，limit，设置的是能使用该音效的最大音程。

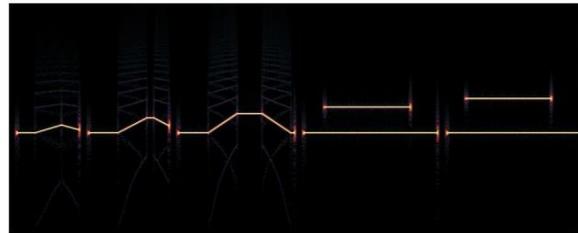
当旋钮在处在最左侧的时候，滑音限制被关闭。该设置的最大值是当旋钮在最右边的时候，此时的设置数值是 2400 分 (cent)。



现在是时候来分享一个小技巧了，告诉大家如何将一个旋钮的值复制并粘贴到一个自动化包络线的点上。假设你想在自动化包络线上设置一个 900 分的点，那么就先将滑音限制设置成 900 分，然后右键旋钮，选择复制 (Copy value)。然后找到相对应的自动化包络线，以及你想设置的点的位置，右键然后选择粘贴 (Paste value)：这样你就不需要自己计算，然后再将 900 分转化为一个百分比。按照这种方法的话，Harmor 会自动帮你计算。

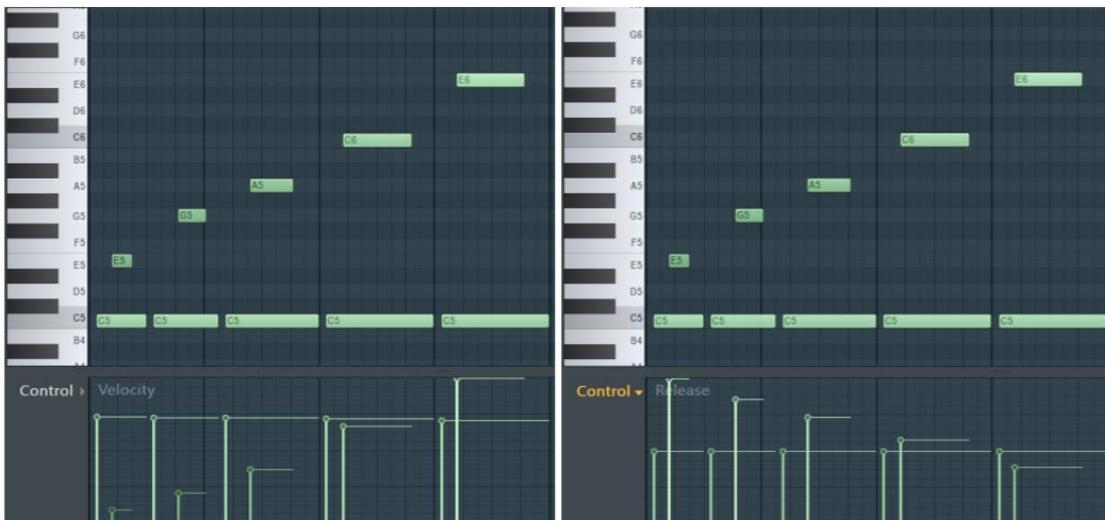


这里你就能看到滑音的音调限制被设置成了 900 分。第 4 和第 5 个音上没有滑音音效：



## 滑音音效的释放时间/滑音速度 (Velocity / Release to legato time)

在这个样式中，我将一组钢琴键排列起来，音调越高滑音速度就越快。并且我还对 legato 的释放时间进行了设定，音调越高释放时间就越短。



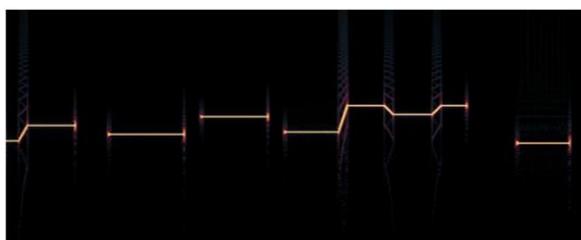


现在，我们打开滑音音效的速度开关，vel，我们可以看到这种设置对于音符的影响。



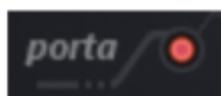
## 延音 (Portamento)

延音效果不但对相邻的音有效，而且还会对重叠的音产生效果。为了对这个效果器进行研究，我创建了项目工程：<https://www.malmgren.nl/Harmor/Harmor-37.flp>

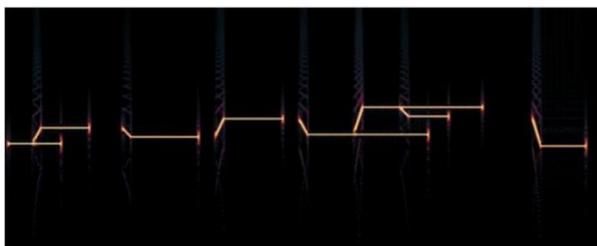


在滑音模式下演奏的音符效果就像上面右边的那张图那样：

## 延音模式开关 (Portamento mode switch)



这是在延音模式下演奏的音符，过度时间设置的是100ms。我们可以看到，当多个音重叠起来的时候，我们可以同时听到两个音。并且，即使音与音之间有停顿，也会出现过渡的效果。当一个音被演奏完之后，延音效果不会再使这个音过渡回去。



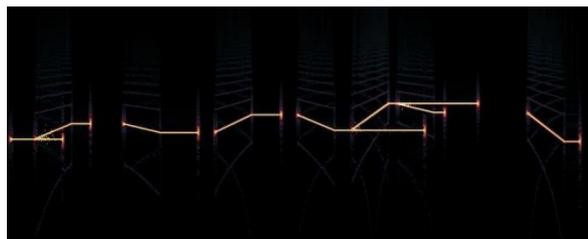
## 延音时长 (Portamento time)



这里我们将延音时长设置到401ms，然后再演奏相同的样式。我们发现过度的时间变长了：



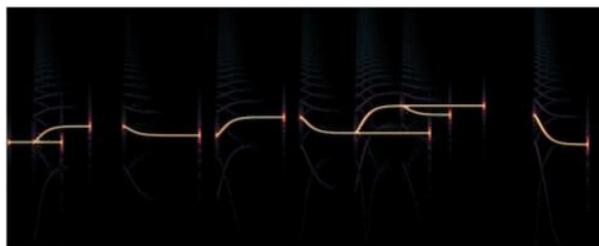
右侧图中的整个工程的时长是 4 分钟



## 线性/对数延音 (Linear/Logarithmic portamento)



在这里我们开启对数开关来演奏相同的样式：

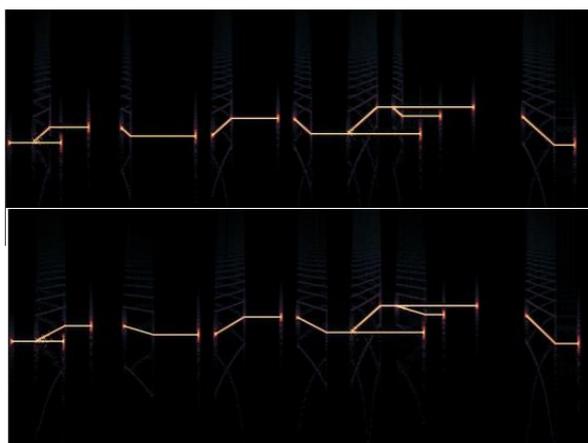


## 固定/可变滑音时长 (Fixed/Variable legato time)



过度的距离可以用来确定过度的时长。当右侧图标被点亮的时候，Harmor 就会开启可变时长：

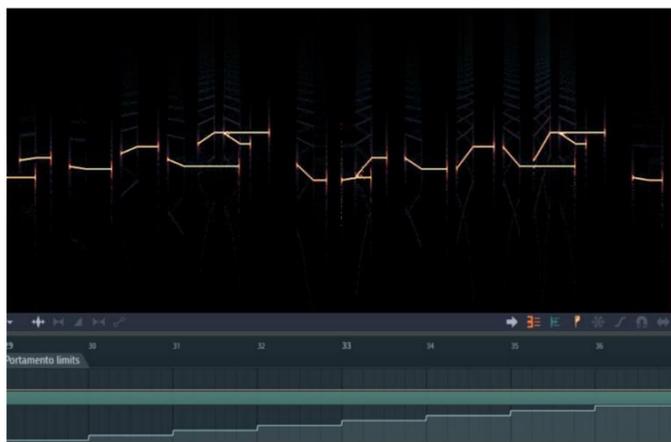
这是在固定滑音时长 (fixed portamento time) 模式下的样子：



## 延音音调限制 (Portamento pitch limit)

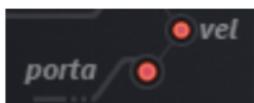
延音音调限制与滑音限制 (Legato limit) 完全不同。滑音限制决定的是滑音音效会在哪里失效。延音限制则限制的是最大过度时间。右边的图片展示的是当延音限制被提高八步 (steps) 时的演奏效果比较：

延音音调限制的值越小，则过度就越轻柔；当延音音调限制的值被增大，也就是第二次演奏的情



况中，我们可以看到演奏的效果更像没有音调限制（unlimited portamento effect）的情况。我觉得，由于这个音效是作用于小振荡器这个层级上的，所以音调限制的按钮被放置在了合成器使用界面的上半部分。

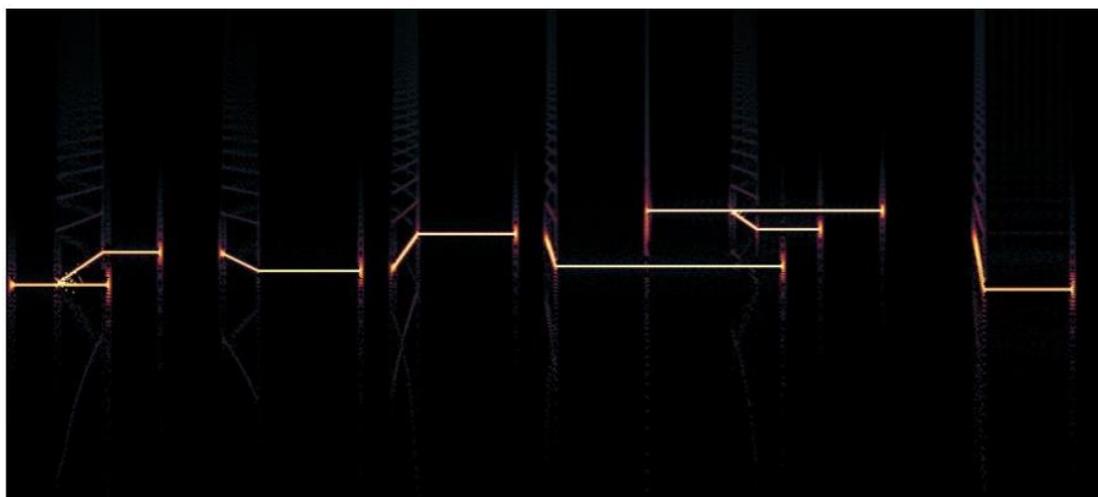
## 延音的起始速度/释放速度（Velocity / release to portamento time）



当我们打开 vel 开关之后，midi 键盘中的滑音速度以及释放时间就会影响延音时长。



下面就是把 vel 开关打开之后再演奏上方样式的结果：



# 第十三章 弹拨 (Strum)

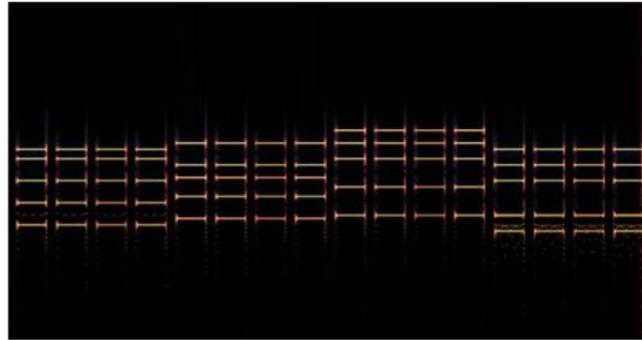


弹拨音效可以让我们将一个和弦中的音在不同的时间演奏出来。我创建了这个工程：<https://www.malmgren.nl/Harmor/Harmor-38.flp> 来研究弹拨音效。该工程中的样式是一段简单的旋律，音色我选择的是正弦波。这样的话我们可以更清楚的看清楚每一个音，以及当我们改变弹拨音效设置之后声音的变化。



在我们开始之前，请注意我改变了这些音的速度。当速度开关开启的时候，一个音的音量越高，则其演奏速度就越快。

当我们播放这段旋律，并且用 Edison 将其录制下来之后，我们发现其频谱与钢琴键上的样式十分相似。



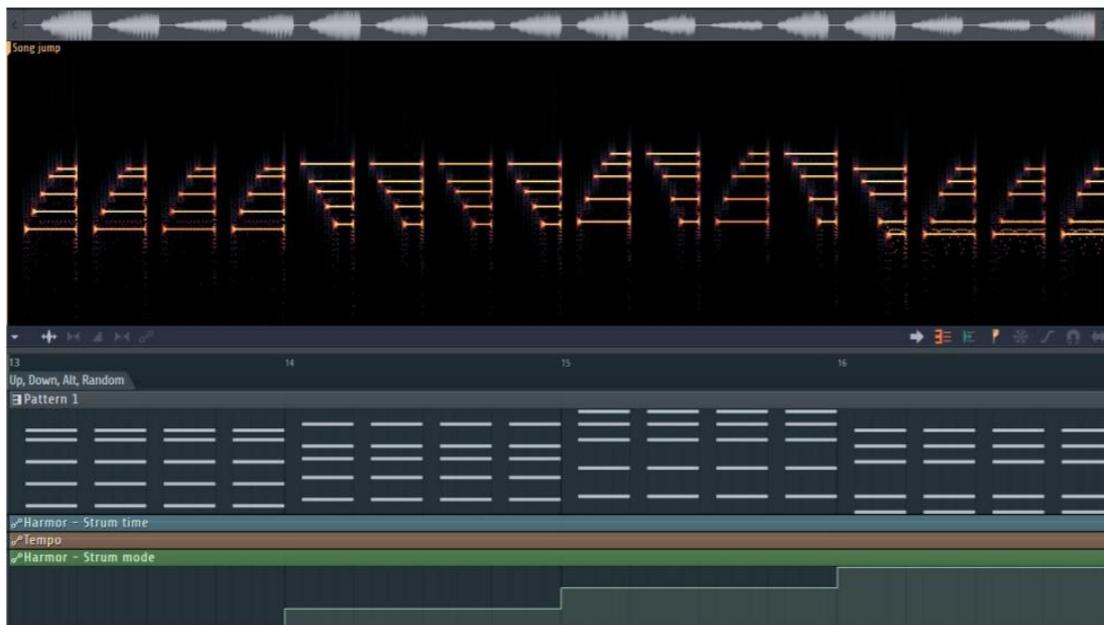
## 弹拨时间 (Strum time)

弹拨时间指的是每个音要等待多长时间才会被弹奏。最小的值是 0%，最大是 100%。但是，是什么的 0%，什么的 100%呢？我尝试了很多可能的设置，比如 BPM，然而依然不是很确定。假设我们将这个时间设置成相对于某一个值，那么 100%就会消耗更长的时间来完成整个和弦的弹拨过程。当弹拨时间被设置成更低的值时，那么弹拨过程将会完成的更快。

## 弹拨模式 (Strum mode)

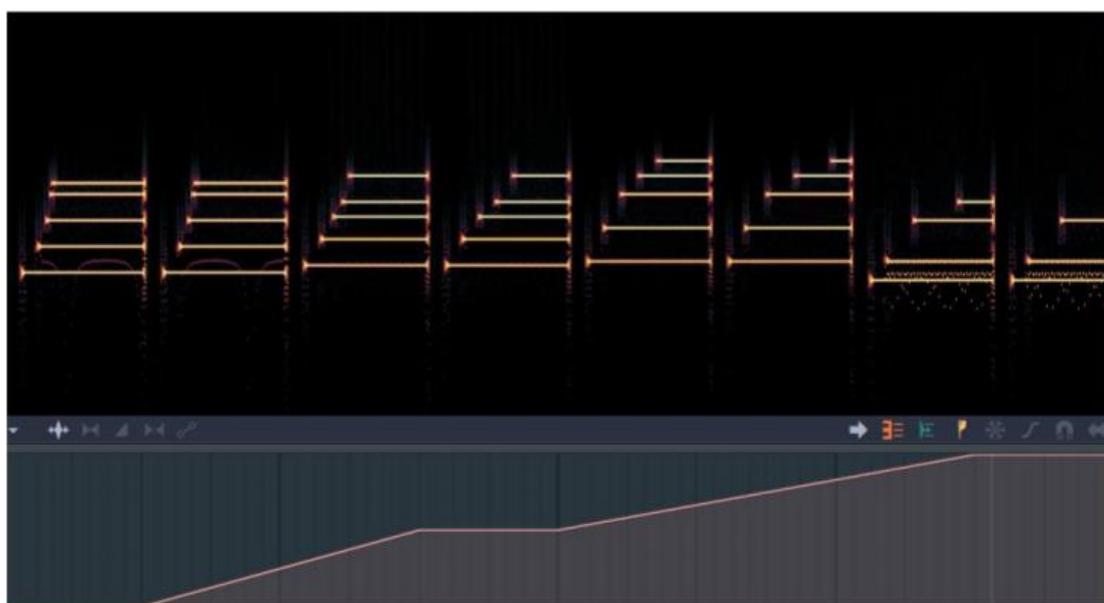
这里一共有 4 个弹拨模式：

- up (向上)。从最低的音开始，然后逐步向上弹奏更高的音。
- Down (向下)。从最高的音开始，然后逐步向下弹奏更低的音。
- Alt (交替)。在前两个模式下交替。
- Random (随机)。从前两个模式中随机选择一个。

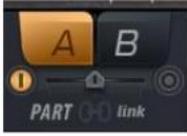


## 弹拨张力 (Strum tension)

弹拨张力旋钮，tns，控制的是和弦中的各个音是如何被演奏的。下面展示的是当弹拨模式是 up 的情况。弹拨张力一开始是 -100%，然后在 0% 停顿，最后向上上升到 100%。当张力是负值的时候，第二个音需要更长的时间才会被演奏，然而接下来的音却更快一些。这个效果在当弹拨张力为 0% 的时候几乎消失，这时，音与音之间的事件间隔基本上是相同的。当张力是正值的时候，第二个音会被快速的演奏出来，然后接下来的音却渐渐的需要越来越长的时间才会被演奏出来。



# 第 14 章 . A/B 部分



在 Harmor 中，有两个音色窗口，两个滤波器区域，两个合成器：A 和 B

## A/B 选择 (Part selector)

如果你想在合成器之间切换，点击 A/B 按钮。在某一个合成器中设置的改变都只会适用于这个合成器，不会影响到另一个合成器。

## A/B 部分激活开关 P (Part enable switches)

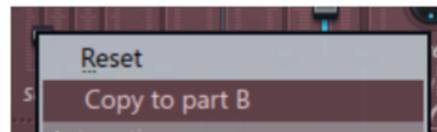
在 A 的标签下面有一个黄色的按钮。这个意思是合成器 A 已被激活。B 的下面有一个圈，这个意思是指 B 没有被激活。如果你关闭两个合成器，那么你也其实就关闭了 Harmor。

## A/B 部分混合

在两个激活开关中间你可以看到一个滑杆，这个滑杆可以用来进行两个合成器的混合。

## 瞬间复制用户的设定更改 (Instant replication of the user's changes)

在混合滑杆的下面你可以找到复制开关， GUI-link switch。当打开这个开关之后，你在任意一个合成器中做出的更改都会被自动复制到另外一个合成器中。你可以这么理解，每次当你做出更改之后，Harmor 就会自动将更改内容复制到另外一个合成器中。



我觉得 link 在这里的设计不是很好。Link 这个词暗含了事物之间的某种逻辑关系，但是这里并没有，而只是简简单单的将用户的指令复制过去了而已。如果你想要真正的 linking 的话，那么就右键，然后选择“连接到控制器” (link to controller)。

## 第十五章 失真 (Distortion)



这是我们讲解 Harmor 下半部分的第一章。在我们讨论之前，快看看这 FX 标签的蓝色光芒多么漂亮!!!



Harmor 的上半部分都是关于如何控制以及处理小振荡器的。在下半部分的许多区域中，我们依然会继续处理小振荡。然而还有另一些部分却是关于处理由小振荡器整合形成的声音的。效果器面板是用来处理声信号的。



在 Harmor 的主界面你可以找到两个滑杆。第一个滑杆 (pre-slider) 控制的是输入到效果器中的音量, 第二个滑杆 (post-slider) 控制的是从效果器中输出的音量。

现在我要开始讲失真了。我决定通过在 Harmor 外部使用 WaveShaper 反向构造失真效果来解释失真这种效果。

### Waveshaping 理论知识

在我的试验过程中，我发现三角波是最适合研究失真效果的波形。我用 Sytrus 创建了我的三角波，然后我将他输出成只有一个周期的波形，最后我把这个文件加载到进 Harmor。

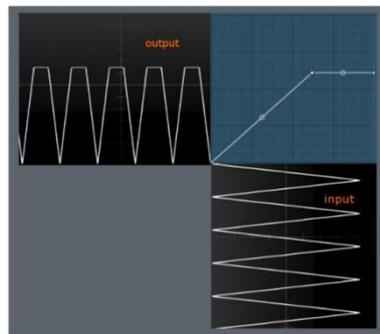


图 48. Waveshaping

我们可以一个二维的图来描述波形的变化。横轴代表输入 (input)，纵轴代表输出 (output)。在图 48 中，蓝色区域现实的是一个波形变化的例子。对角线从左下角开始一直延长到右上角，这条线上输入不会被经过加工，会被直接输出。(译者注：建议下载下面的工程然后将各个旋钮拧一拧试一试，有助于理解)

我创建了一个工程文件 <https://www.malmgren.nl/Harmor/Harmor-39.flp>，通过 WaveShaper 的试验来研究失真效果。在这个工程文件中，左侧的 WaveCandy 振荡器显示的是输入，右侧的振荡器显示的是输出，中间是 WaveShaper 的用户界面。



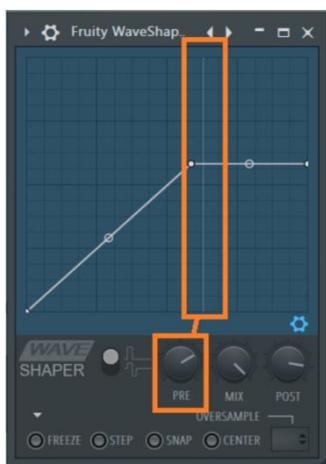
## Waveshaper 中的单极以及双极模式



WaveShaper 中的功能要比之前讲的理论知识要更加多种多样。这里有一个按钮来进行单极 (unipolar) 以及双极 (bipolar) 之间的切换。在单极模式下, 整个变化图谱所展示的变化都在零线 (zero-crossing line) 之上的变化。所谓零线也就是波形的中心线。然后相同的变化也会在零线之下发生。

在双极模式中, 你可以对零线上面和下面的部分进行不同的变化。

## WaveShaper 中的 Pre amp



在窗口中有一个 pre amp 旋钮, PRE, 这个管理的是输入的水平。要想让 WaveShaper 来对波形进行变形的话, 输入音量必须充足。这条垂直的细小白线的作用是展示峰值电度表的。我们可以调节输入音量, 以使得峰值电度表的那条竖线抵达变形图的合适位置。

## WaveShaper 中的混音水平 (Mix level)

混音水平, mix, 使我们可以将失真信号混进原信号中。

## WaveShaper 中的 post gain

Post 旋钮控制的是输出音量, 我们可以用这个旋钮来放大失真后的声信号。

现在让我们来讨论一下 Harmor 中的失真控制吧。在 Harmor 中没有 WaveShaper 的图

## 失真类型 (Distortion type)



这里有一个下拉菜单, 菜单中有 11 种失真类型, 以及 Off (关闭) 选项, 在默认设置下, 失真是被关闭的。我相信你可以通过观察 WaveShaper 的图像来理解这些不同的失真类型的。我们过一会再来看这些不同失真类型的图像到底是什么样的。

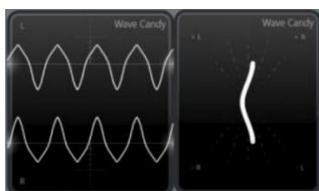
## 失真度 (Distortion amount)

amt 滑杆可以从 0%推到 100%。默认值是 50%。当在 0%的时候，失真效果则没有被激活。在 100%的时候，许多时候失真效果会把声信号转化为一个方波，或者近似方波。

## 失真不对称 (Distortion asymmetry/extra)



对于 asym 滑杆来说，当我们将这个滑杆向上推动的时候，就会产生一个单声道的调节。在示波器中，我们可以看到，波形的上半部分稍微变得平整了，而下半部分失真的程度则降低了。请看左边的图。



当我们把滑杆向下推动的时候，就会出现立体声的失真。在矢量显示器中，看起来形状好像弯曲了。在示波器中你可以看到波形就像刚才向上推的时候，也变得更平整了，但是整个声信号都发生了 180 度的相变，而且还倒过来了。当我们分析像这样的信号时，示波器不再显示细线了，所以也许我们可以尝试着将图像分解开来看看是不是会提供更多信息。

## 失真湿声音量 (Distortion wet volume)

Wet 滑杆控制的是失真效果的放大倍数。这个跟 post 旋钮的功能很像，不管是 wet 滑杆还是 post 旋钮都能将失真信号混到原声信号中。

## 失真混合 (Distortion mix)



当 mix 滑杆被推到+100%的时候，失真效果被关闭。在-100%的时候，声信号就会被倒过来。在左边的图像中，我们看到的是在+100%的时候的锯齿波。右边的图像中显示的是在-100%的时候的



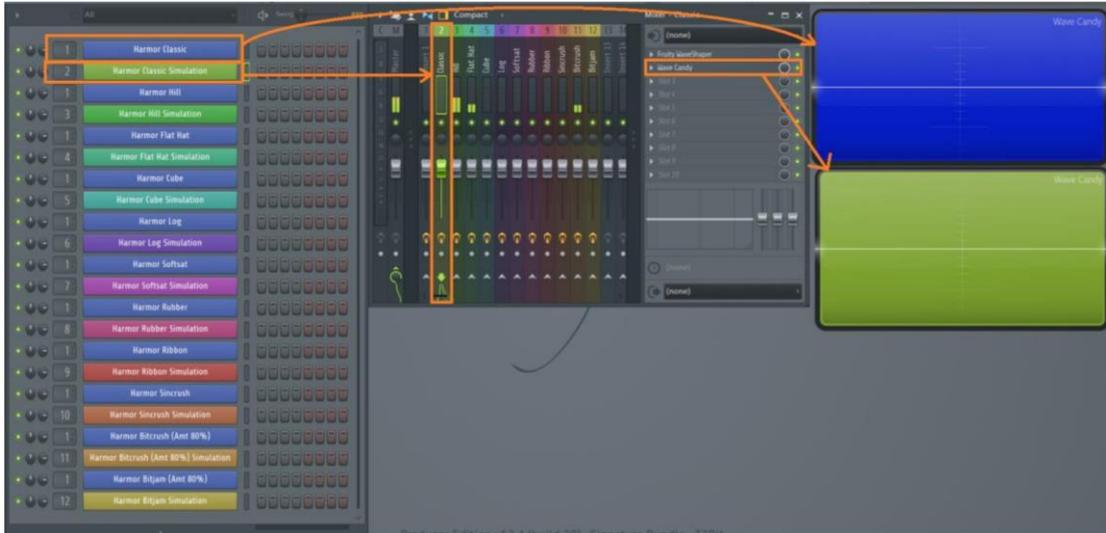
锯齿波。当 mix 滑杆在 0%的时候，失真信号与原信号相混合。在 50%的位置上时，只有失真后的声信号会被输出。在-50%的时候，同样也只有失真后的声信号会被输出，但是这个时候，波形则是倒着的。

## 失真高切 (Distortion high cut)

这其实是低通 filter 的另一个名字，默认中仅仅只被开启了一点点（译者注：在 Harmor 中该滑杆的标签是 filter，但是左上角的工具提示信息写的是 Distortion high cut）

## 失真效果示范工程

请打开 <https://www.malmgren.nl/Harmor/Harmor-40.flp> 这个工程，然后尝试不同的 Harmor 失真类型。



在这个工程中，我在通道台中插入了 22 个 Harmor。每个失真类型都有两个 Harmor，一个是原始的，一个是使用 WaveShaper 模拟的。

这里有 11 个蓝色的 Harmor，他们都通向混音台的轨道 1，轨道 1 中还有一个蓝色的 WaveCandy 振荡器。你可以选择，比如说 Harmor Classic 然后弹奏 C5 这个音。如果你

点击 Harmor Classic 这个按钮，FLStudio 会选择这个通道，并且还会展示 Harmor 的交互界面。如果你点击了蓝色按钮右边的那个小长方形，那么通道就会被选中，但是不会弹出 Harmor 的用户界面。这很方便防止把界面搞得一团糟。

当我们选中 Harmor Classic 并且弹奏 C5 这个音的时候，蓝色的示波器就会显示一个 Harmor Classic 形式的失真三角波。如果我们点击 WaveCandy 上这个“冻结”按钮，那么我们就依然能能在不弹奏这个音的情况下看到这个音的波形。如果你想打开 WaveCandy 的界面的话，你可以要么在混音台上点击它，或者你可以右键 WaveCandy，然后选择查看设置 (view settings)。当图像被冻结的时候，我们可以松开 C5 键，然后将 WaveCandy 的设置界面隐藏起来，然后拿下一个 Harmor 来比较。

接下来选择，比如 Harmor Classic Simulation. 这个 Harmor 被连接到第二个混音器中，在这个混音器中也有一个 WaveShaper 以及一个绿色的 WaveCandy 示波器。你可以再次演奏 C5 这个音，然后在 WaveShaper 中观察这个三角波是如何被失真的。

也许在现实中你没有钢琴键盘？没关系！当这个 FLStudio 图标  被点亮的时候，你也可以用电脑键盘来弹奏。在工具提示界面你能看到你正在弹奏的是哪个音。你能在键盘最下面一行的右侧找到 C5 这个音。在我的键盘上，C5 是“<,”这个键



示波器没有办法放大到一定程度来展示波形所有的微小细节。为了解决这个问题，我把 C7 当做基音来用，这样的话结果就是 C3。这样低的一个音在示波器中将能更好的展现各种细节。(译者注：这里的意思是，现在">"这个键弹奏的是 C3，而不是像之前那样是 C5)



对于下面所有的试验来说，除非有被特殊的提及，对于失真效果我都将原始 Harmor 设置成了像下面这张图中所展示的一样。这样的话，我既没有添加高切滤波器，也没有将其混到原始信号中去。

## Harmor Classic Distortion



## Harmor Hill Distortion



## Harmor Flat Hat Distortion



## Harmor Cube Distortion



## Harmor Log Distortion



## Harmor Softsat Distortion

这个 filter 基本上啥作用也没有



## Harmor Rubber Distortion



总算是感觉有点区别了……

## Harmor Ribbon Distortion



## Harmor Sincrush Distortion



## Harmor Bitcrush Distortion



这个失真类型包含了许多波形上的变化，但是我们却看不到。当我们观察频谱图的时候，我们可以看到一个额外的“音” (tone)。当我们把 amt 滑杆向上推到 80% 的时候，示波器中的图像看的就更清晰了。我决定试着重新制作出这样的失真来。



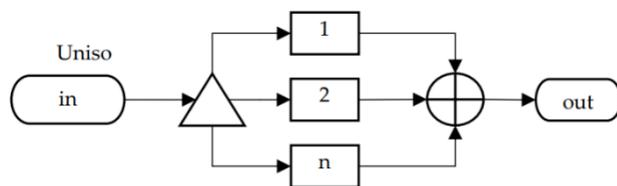
## Harmor Bitjam Distortion

同样在这里我将 amt 设置成了 80%

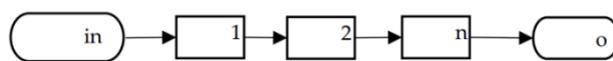


## 第 16 章 . 合唱 (Chorus)

合唱效果是 Unison 一个更加轻便的形式。在右边你可以看到一个图，图中展示的是两者之间的区别。Harmor 中的 Unison 是一个在小振荡器级别上的加和效果 (additive effect)。每一个声音的处理都是同时发生的。合唱则是在声信号这个级别上的一个效果。合唱对声信号进行处理，然后将声信号送到下一步去处理……。

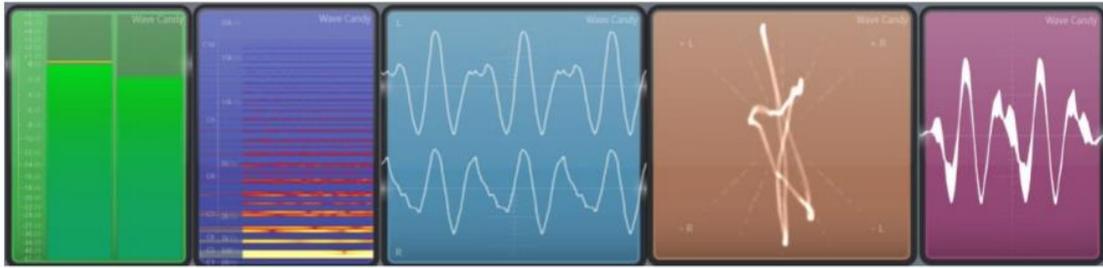


Chorus:



处理的步骤个数叫做阶数 (order)，步骤本身叫做一个声音 (a voice)。我为合唱效果制

作了一个示范工程：<https://www.malmgren.nl/Harmor/Harmor-41.flp>。在这里我用合唱效果演奏两个音，结果我们可以发现矢量显示器展示了一个不断在变化的 3D 五角星！（译者注：这里墙裂建议自己试一试，同时按下两个琴键，结果的图案非常美！）



Harmor 中的合唱效果就跟 Flangus 效果器插件一样。Flangus 就是通过叠加不同阶数的 Flanger 效果来工作的。

## Fruity Flanger

为了更好地理解 Harmor 中的合唱效果，我们可以参考一下 Flanger 以及 Flangus 是怎么工作的。在测试工程 <https://www.malmgren.nl/Harmor/Harmor-42.flp> 中，我用 Flanger 以及一个带有立体声分解和矢量显示器的示波器建立了一个工程文件。你可以通过演奏 C5 来把玩一下。最开始的时候你什么也听不见。



Flanger 有 12 个旋钮！前三个要尝试的是干声（dry），湿声（wet）以及延迟（delay）。干声旋钮控制原声信号的输出，湿声旋钮控制的是被加工过的声信号。你可以将这两个调到 100%。延迟旋钮是要在干声以及湿声之间创建一个延迟的效果。如果我们只有湿声旋钮被打开的话，延迟的效果就不会很明显。



假设我们将延迟（delay）设置到大概 1 毫秒。现在我们将 depth 设置到比如说 2 毫秒。



通过这样的操作我们引入了大概 3 毫秒的延迟效果。Flanger 自己会自动的在 delay 以及 delay+depth 之间扫动。在我们的这个情况中，也就是在 1 到 3 毫秒之间。现在我们

用 rate 按钮来设置扫动速度，比如 0.3Hz。



当你在这个测试工程中尝试这些的时候，你可以看到右边以及左边的声信号会从干声扫到湿声。现在当你将 phase 旋钮向右旋转 90 度之后，左右声信号的相位就会错开 90 度。



Damp 旋钮能对湿声提供一个高切滤波，但是不能对干声提供。在下面你可以看到一个波的头是圆的，另一个则是尖的。



Shape 旋钮改变的是在这个扫动模式下的波形。正弦波可以让扫动更加平滑。现在假设你不想要任何干声，你要怎么做呢？我们可以将湿声重新加入到效果器中，来代替干声。



因为当我们增加输入比例（feed percentage）的时候湿声已经包含了早期的输入，输出的音效就会立即包含许多原声及其复制。下面你可以看到当我们把湿声颠倒过来的时候是什么样的



在这里我们将反馈信号颠倒过来。

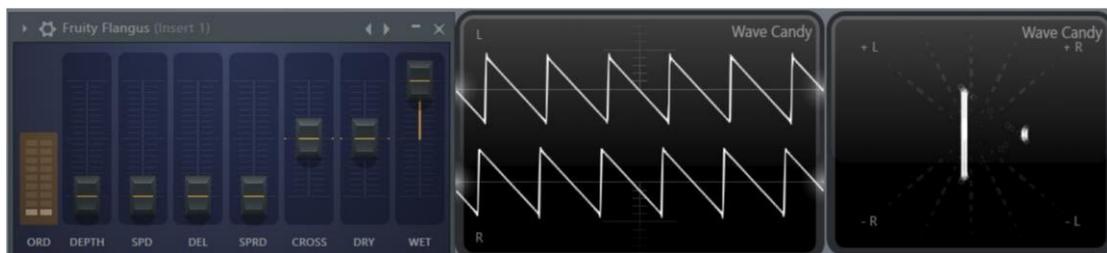


Cross 旋钮会将左侧的湿声立体声信号复制一部分然后粘贴到右侧的立体声信号中，反之亦然。



## Fruity Flangus

现在我们可以以同样的方式来分析 Fruity Flangus 了。我用 Fruity Flangus 以及一个示波器和一个矢量显示器制作了另一个工程，下载链接在这里 <https://www.malmgren.nl/Harmor/Harmor-43.flp>。下面的图显示的是 Flangus 在湿声设置是 100%的时候直接将声信号输出的结果。



请注意我们也可以输出 -100%的湿声。这个时候左右声道会有一丁点不同



我将湿声还原成+100%了。当我们调大干声旋钮的时候，我们会加入第二个声信号让他们叠加起来。我们发现叠加之后的声信号是稳定的。



当我们调高 depth，我们会在两个声信号之间加入一些差距



通过在阶数控制器（order control），CRD，中点亮一个长方形，我们就得到了 3 个声信号混合而成的输出信号。你可以试试点亮更多的长方形，然后看看会有多少个波峰会被加入到图像上。



当我们调高速度滑杆 spd 的时候，抖动开始出现。在 Flanger 中这个叫做 rate。

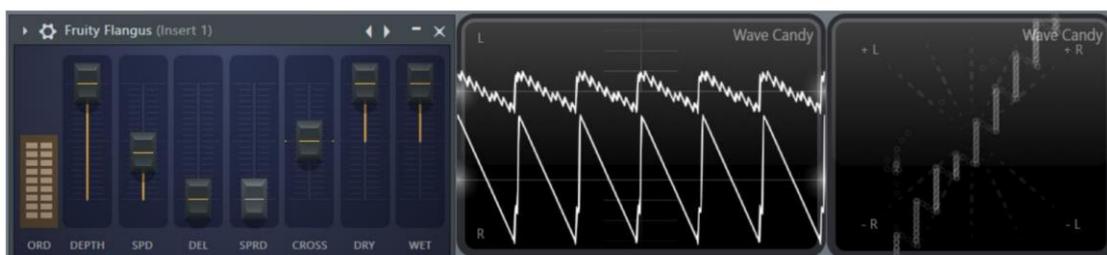


延迟，del，控制的是不同的声信号被添加到效果中所需要的时间。我必须提一句，如果你不知道这一点的话，这个效果看起来只有在第一个声信号以及其他声信号之间的这一个延迟。也就是说首先我们有一个声信号，然后延迟，然后几乎所有剩下的声信号几乎同时被加了进来。当 sprd 旋钮被调整到 0% 的时候尤其明显。当 sprd 旋钮被设置到 100% 的时候，声音之间的延迟变得更加均匀分散开了。当我们用 Edison 将声音录下来之后能看到这个现象，就是在弹奏过一个音之后，将这个音录下来，在 Edison 中打开，将波形放大，然后就能看到这种现象了。Harmor 的官方手册说每一个声音都被分配了“一个不同的什么附加物”。那到底是什么东西？

请回忆在 Flanger 中我们有一个 phase 旋钮。假设当 sprd 被设置成 100% 的时候，Flangus 中的每一个声音的 phase 旋钮都会在 360 度之间均匀分散开。



但是，我能看到的是 sprd 旋钮使得 phase 的使用更加“有效”了，特别是当我们有许多声音 (voices) 的时候。在上面的这张图中，sprd 被调成了 100%，然后声信号就会晃动形变，但是整体来讲立体声的图像控制的还是很好的。



这里的 sprd 是 0%。我们会发现出现了有趣的几何图形，但同时也能看到一个更加狂野的立体声图像。

Cross 旋钮有两种模式，正向以及负向。正向的设定与 Fruity flanger 中的 cross 作用一样，他都是将左声道中的湿声复制一部分，然后再粘贴到右声道中，反之亦然。

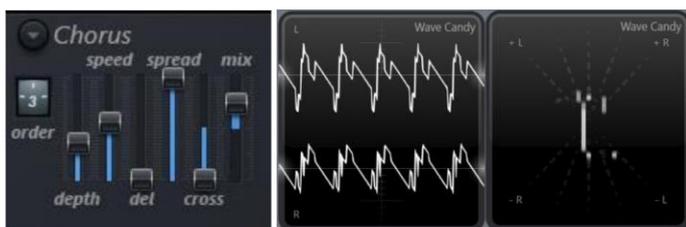


当我们将 cross 调成一个负值的时候，被复制的声信号被倒了过来。



## Harmor Chorus

现在我们终于可以得出一个结论，也就是 Harmor 的合唱效果制造原理与上面的这些完全一样！



当 Harmor 中的 mix 被设置成 0% 的时候，我们只会得到干声。在 +100% 的时候，我们只会得到湿声。在 Harmor 中是 50% 的时候，就像是在 Flangus 以及 Flanger 中，干，湿声被调成 100% 一样。当 mix 是负值的时候，你会得到相同的波形，但却是倒过来的。请看 <https://www.malmgren.nl/Harmor/Harmor-44.flp>。

## 合唱预设 (Chorus presets)

合唱预设的下拉菜单囊括了一些非常不错的预设。下面是各个预设中不同参数的默认值：

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

## 第 17 章 . 延迟 (Delay)



在你成为一名音色工程师之前你会把这个叫做回声。从今天开始，你就可以将这种效果称之为延迟了。但其实这就是回声。在真实的生活里，你一定体验过回声，比如在落基山脉中，或者是在挪威的峡湾之中。在这些场景中，你面对的山距离你并不是特别远。现在假设你站在峡湾的一侧，然后大声的喊一句话，那么你的声音就会从对面的山反弹回来，这之间就出现了时间差，而且声音的响度也降低了。

如果你很幸运的话，声音会来来回回反弹个几遍。我们就可以用今天要讲的效果器来制作出这种音效，但是我们并不将这种效果器称为回声，我们把它叫做延迟。

这种音效背后的原理很简单，声音信号首先被输入到这个效果器中。在效果器会将声音信号先记录下来，然后再在特定的时间播放出来。

Image-Line 公司的线上 Harmor 说明中有一张像上面那样的一张截图，但是文稿竟然是

在讨论其他的滑杆??!?

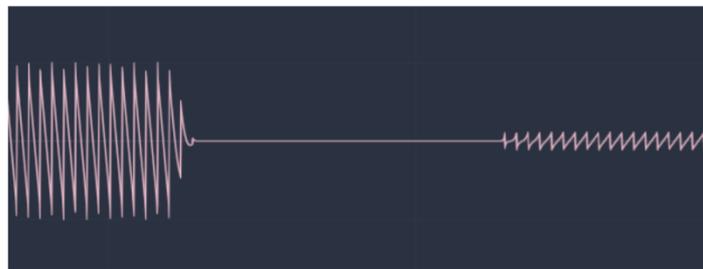
## 开启延迟效果 (Enable delay)

 这不是一只猫的绿眼睛。这是你开启了延迟效果器。

## 延迟输入音量以及延迟左右声道输入均衡

 我们不一定非要将所有的声信号都输入到效果器中。我们可以通过在 0%到 100%之间调整这个滑杆来达到这个目的。

 从另一方面来讲,你也许想将输入信号倒过来。这样的话,你可以通过在 0%到-100%之间调整这个滑杆来达到这个目的。在右侧的图像中你可以看到在一开始的输入声信号是一个正常的锯齿波,然后回声将这个信号倒了过来,因为这里设置的是一个负值。



如果你需要在输入的一端收紧,那么你可以通过调整最上面的左右声道 (panning) 旋钮。

## 延迟反馈模式: (Normal, Inverted) 以及 (Ping Pong)

 这里有三种反馈模式。Normal 是从输入发送到输出。Inverted 将左侧发送到右侧,反之亦然。Ping Pong 则是在左右两端来回切换。

如果你想使用 Ping Pong 模式,那么你就可以将输入端的左右声道输入均衡旋钮设置成正或负 100%,然后将其模式更改为 Ping Pong。这样的话,输出的声信号就会在左右之间反复横跳了。

## 延迟反馈水平 (Delay feedback level)

延迟反馈水平旋钮, fb, 控制的是在刚开始被录下来的原声会多久之后消失。在 100%的时候,这段录音永远不会消失。在 0%的时候只会有一次回声。



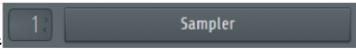
## 延迟时间 (Delay time)



这一个滑杆控制的是声音从山谷的一边然后再传回来所需要的时长。在这个滑杆中间的上半部分控制的是相对于整个工程而言的时间。下半部分则是以毫秒为单位来控制这个时间。

常常当我们提到时间的时候，我们的概念其实是非常混乱的。很明显理解时间的基本含义一点都不难，但是你真的理解我们在这里讨论的时间吗？这里有个小试验你可以做一下，以帮助你理解到底什么是工程时间。你可以录下你想要研究的一段声音，然后将这个录音放到钢琴条形窗口中，然后你就知道工程时间有多长了。我并没有制作出一个工程文件来展示这个试验，所以如果你想做的话，请使用以下步骤：

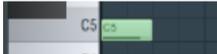
1. 创建一个新的工程文件

2. 将默认采样器  换成 Harmor

3. 在 Harmor 中激活 FX 标签，然后打开延迟效果器 

4. 将延迟时间调整到 5:00 然后将反馈调整到大概 80%，保持其他设定不变

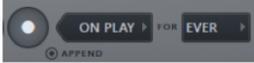


5. 在通道机架上右键 Harmor 然后选择钢琴窗。添加一个 C5 键。你能听到回音，但是回音和原音之间的时间差到底有多长呢？ 

6. 现在打开播放列表，然后将该模式放到播放列表当中。将模式拉的足够长 

7. 在混音台的第一个混音器中添加 Edison 

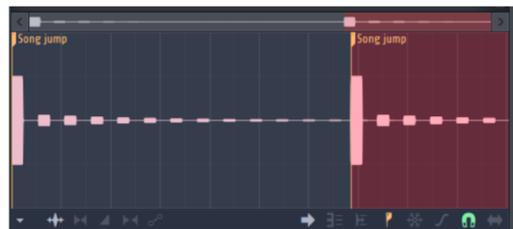
8. 将 when 这个下拉菜单设置成“播放中” (On Play) 然后将录制

时间设置成“永远” (Forever) 

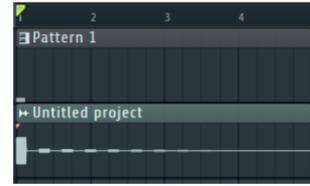
9. 按下那个圆形的录制按钮（不要按工具栏中的那个录制按钮）

10. 现在在工具栏中点击播放

11. 当开始播放的时候 Edison 就会开始录制。当一首歌的 song jump 出现的时候，你就可以停止播放了。回到 Edison 单后双击最后一个 song jump 标记。现在你可以按下键盘上的 Del/Delete 键来删除其余的声音了。

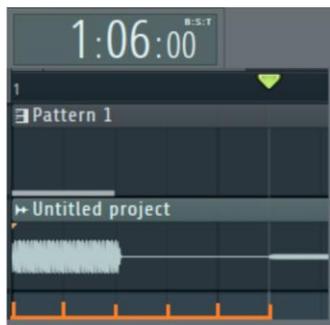


12. 用“拖拽按钮”将文件拖到播放列表上。在那里释放声音
13. 现在我们就把录制好的声音放在了播放列表上。当准备好的时候我们就可以开始研究他的图案了
14. 在工具栏的右侧点击时间窗口。这样的话我们就可以看到它的模式是 B:S:T。B:S:T 的意思是 Bar:Beat/Step:Tick, middle unit 是 Step.

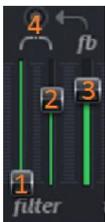


现在我们可以开始数了! 在右侧的图像中你可以看到那个绿色的光标在第一小节, 在第 6 步开始, 他用了 5 步才到那里!

有了这个方法, 你就可以在不同的时间类型之间进行切换, 而且再也不用担心混淆了!



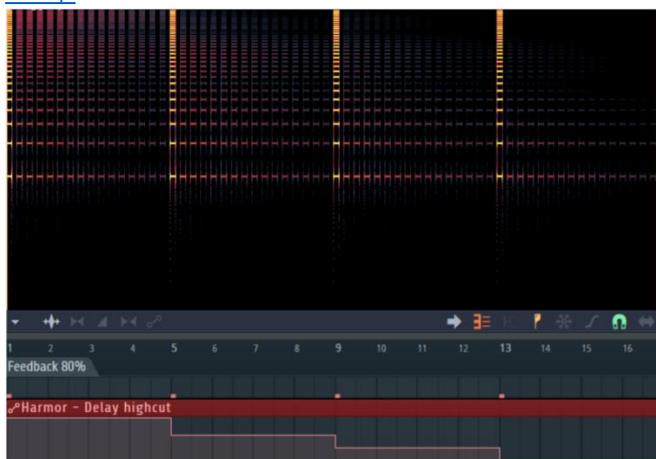
## 低切, 高切, 反馈以及反馈衰减



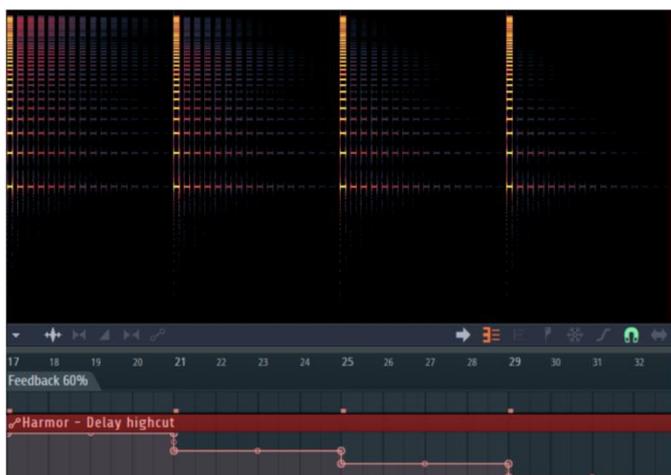
现在让我们讨论讨论左侧图片中的控制器吧!

- 延迟低切 (Delay low-cut)。这是一个高通滤波器
- 延迟高切 (Delay high-cut)。这是一个低通滤波器
- 延迟反馈 (Delay feedback)
- 延迟反馈衰减 (Delay feedback damping)

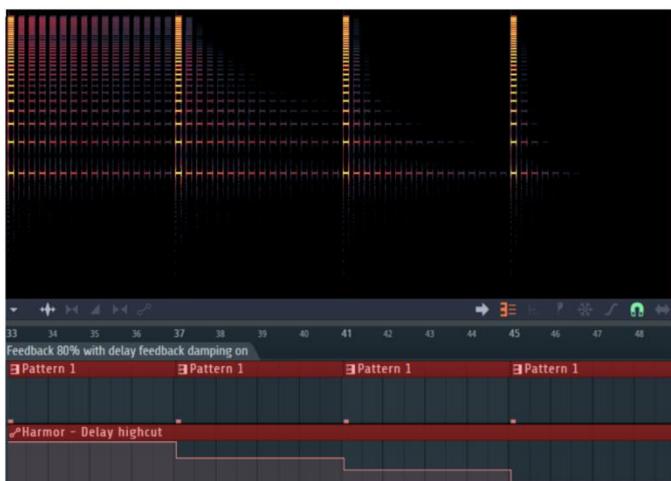
为了解释这一部分, 我建立了一个范例工程: <https://www.malmgren.nl/Harmor/Harmor-45.flp>



在这个工程的第一部分中, 在标签“Feedback 80%”中我演奏了一个 C6 音。这是一个锯齿波, 并且我也打开了延迟效果器。反馈被设置成了 80%。延迟反馈衰减 (Delay feedback damping) 被关闭了。现在我们可以演奏一个音符单后让它慢慢消散, 然后观察这整个过程。在这个工程中, 我将高切滑杆自动化, 以关闭 (off), D8, F6, G4 的顺序摆放。在左侧的图中你可以看到, 在音符被演奏之后, 回声效果就会出现, 然后不断重复演奏同一个音符, 并且音量慢慢减弱。

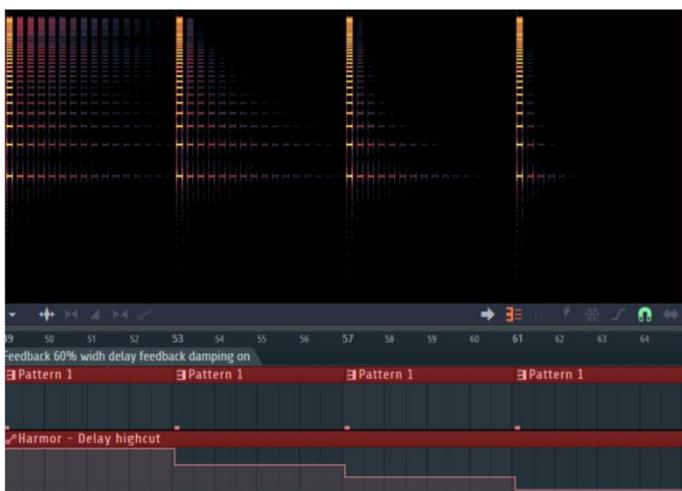


回声音量减弱的速率 (rate) 是通过反馈 (feedback) 滑杆来控制的。为了说明这一点，我们将反馈调成 60% 然后演奏。现在声音消失的更快了。我们可以发现有一个趋势，就是低音会持续的时间更长一些。



现在我们要再做一边相同的试验。但是这一次我们会把反馈衰减 (delay feedback damping) 这个开关打开。在左边的图中，feedback 被设置成了 80%。

这次回声消失的更快了，因为高切的参数调低了。在反馈衰减这个开关开着的时候，输出的声音会被重新添加到滤波器中，然后被切过的声音会被再切一次，所以声音会被切的很干净。

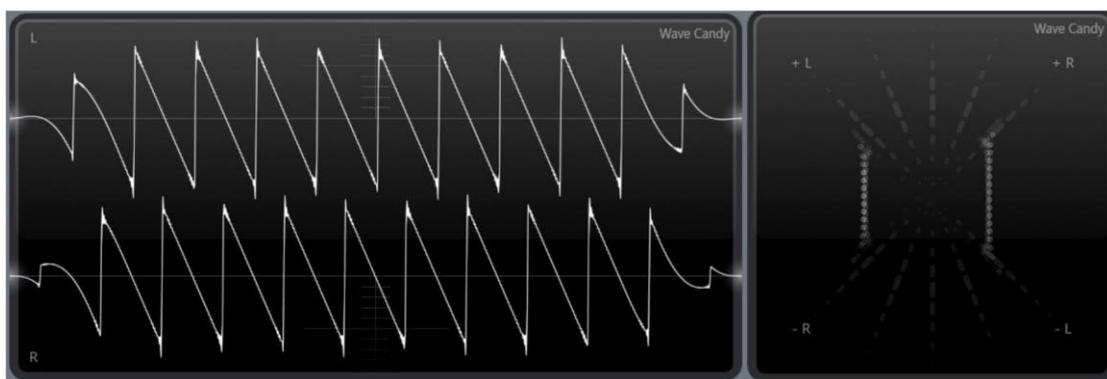


在左侧的图像中，反馈被设置成了 60%，然后反馈衰减被打开。声信号消失的更快了。

对于延迟高切滤波来说，结果是相同的。但不是很直观，所以我不添加图片了，以免你觉得无聊。

## 延迟时间立体声偏移 (Delay time stereo offset)

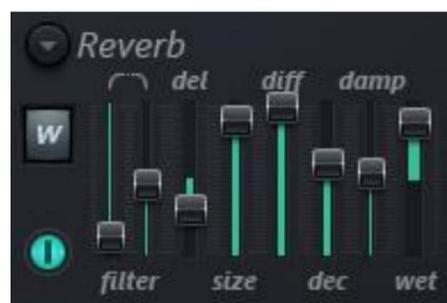
这个滑杆的标记是 s.ofd，这个滑杆控制的是左右声道的偏移。



这个效果是在左右声道之间增加相位差。在这个工程里 <https://www.malmgren.nl/Harmor/Harmor-46.flp> 我搞了一个试验，在这个试验里我将偏置设置自动化了，所以我们可以把玩不同其他的滑杆来看看会产生什么样的效果。

## 第 18 章 . 混响 (Reverb)

如果你刚接触音色设计，然后第一次打开混响，神奇的事情就会发生。有了混响，你在弹奏的声音就会突然产生一种空间感。混响与回声的原理是相同的。也许你曾经尝试过延迟时间很短的回响，你是不是发现这种效果就跟混响很像？这其实就是混响。在上一章，我们用山谷以及挪威的峡湾中来回反弹的声音作为例子。混响其实就是反弹距离很短的回声，比如在房间里的回声。在混响中有一些不同的要素来局盯着混响的音色，比如房间的大小，以及墙上的饰物等等。



也有可能是我错了，但是我觉得混响效果的本质是将一堆回响音效叠加起来。哦抱歉，我的意思是“延迟效果”。

Harmor 中的混响效果有可能是从 Sytrus 借鉴来的。如果你对 Sytrus 熟悉的话，那你一定会很高兴的，因为 Harmor 的混响效果器界面跟 Sytrus 的一模一样。

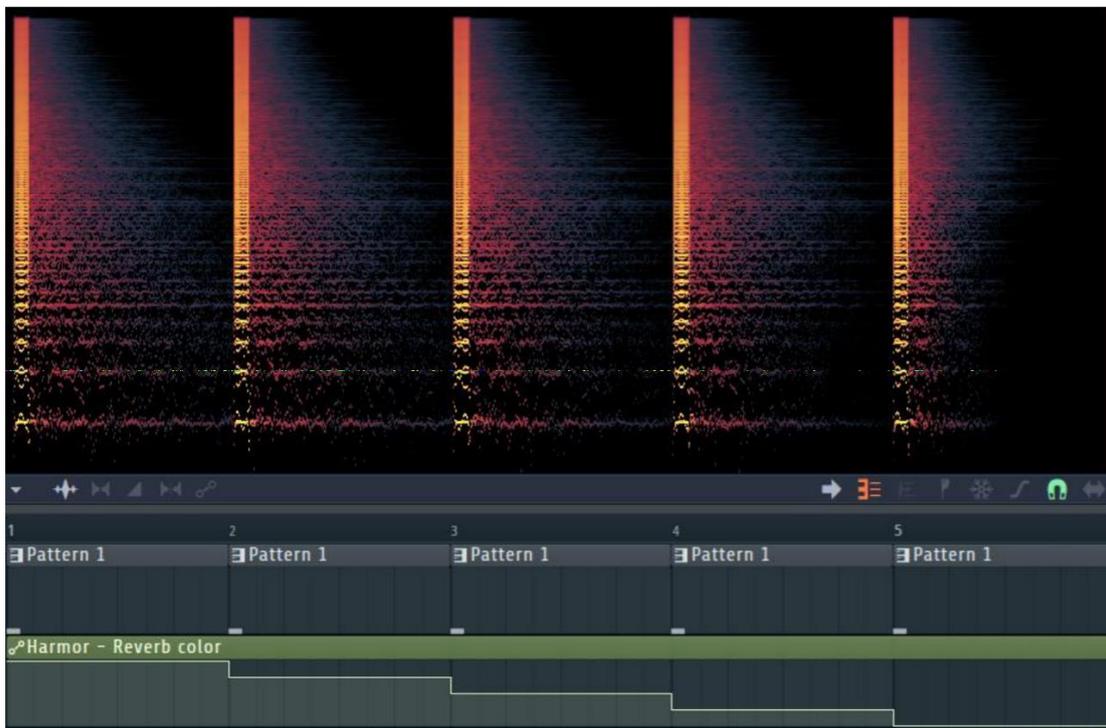
### 混响颜色 (Reverb color)



回响颜色选择器控制的是在混响效果中低频衰弱的方式。当混响效果中有许多低频的声音的时候，我们将这种空间效果称之为“温暖” (warm)，并且当低频声音很少的时候，我们称之为“明亮” (bright)。这里有五种混响颜色设置：

- Brighter (更加明亮)，更少的低频声音 (B+)
- Bright (明亮) (B)
- Flat (平整) (F)
- Warm (温暖) (W)
- Warmer (更加温暖)，更多低频声音 (W+)

我们得做些试验来研究研究这个效果！在这个工程中 <https://www.malmgren.nl/Harmor/Harmor47.flp> 我插入了一个默认设置下的 Harmor，然后开启混响效果，并对混响颜色添加自动化包络线。下面就是其看起来的样子：



第一个 bar 播放的是混响效果中最温暖的那个版本。我们可以看到低频部分十分的有活力。越是朝着“明亮”的方向发展，低频的声音消散的就会越快。在最亮的那个设置下，低频消散的就是最快的。

## 混响输入滤波器 (Reverb input filter)：高切以及低切



我们可以在声信号进入到混响效果器之前就将其进行过滤。在 filter 标签上有两个滑杆。左侧的是混响低切滤波器，右侧的是混响高切滤波器。如果我们知道他们的功能是什么，那么他们的名字是什么其实没有那么多所谓。低切滤波器通常又叫做高通滤波器，然后高切滤波器一般被叫做低通滤波器。对于一个新手来说，这样容易混淆的概念很容易就让他放弃了音色设计。但是让我们再稍稍坚持一下吧，因为我们正要进入 Harmor 中最令人激动的部分了！

如果我们可以在声信号的低频进入到混响之前就将其过滤，那我们为什么还需要混响的上色系统呢？这是因为这完全是两码事。滤波器只会作用在输入信号上，但它不会在声音衰减的过程中被激活，而混响的上色功能却可以做到这一点。现在假设你对低频声音进行了很大程度上的削弱，那么结果就是声音听起来非常明亮，难道这不正说明了滤波器的的确确在工作吗？其实并没有，因为混响只会处理滤波器过滤后的声音，所以如果你把所有的低频都消掉，那当然输出的声音会非常明亮。

## 混响预延迟 (Reverb pre-delay)

我们可以用混响预延迟, `del`, 对混响的开头进行延迟

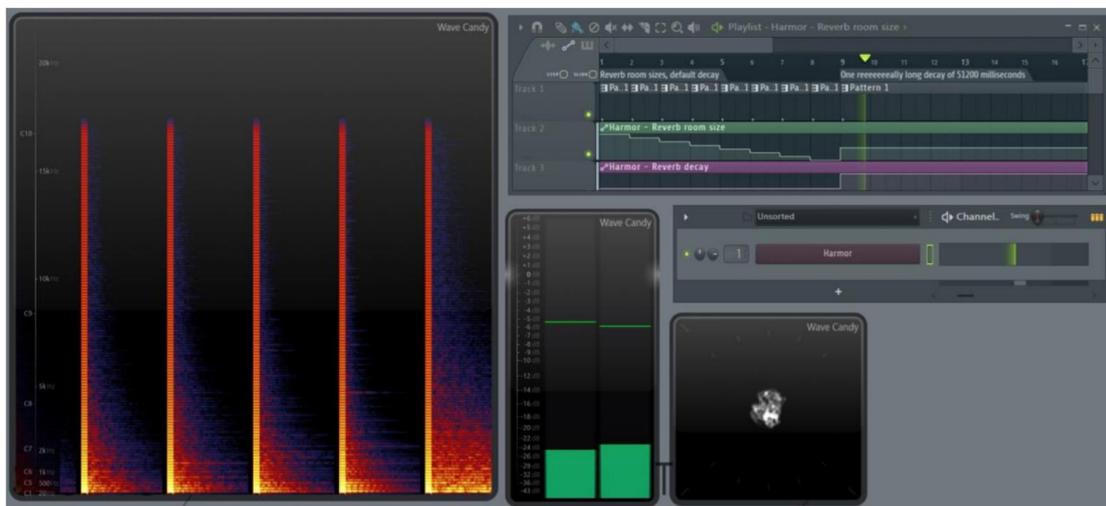
## 混响空间大小 (Reverb room size)

混响空间大小, `size`, 控制的是空间的大小。空间大小在这里很明显是一个抽象的词语因为音乐中没有空间。在这里的话, 指的是一个算法。有人可能会问, 这个声音效果听起来的确有种空间感, 但是为什么呢? 如果我们从分析声信号的角度去看的话, 空间大小指的其实就是声音的高频会用多长时间消散。如果是一个大空间, 那么高频持续的时间就会更长。低频的衰减时间不随空间大小的变化而变化。

## 混响衰减 (Reverb decay)

混响衰减, `dec`, 影响的是混响效果的衰减时间, 但是这个滑杆既可以控制高频也可以控制低频。

我们需要做试验来对空间大小以及混响衰减进行更加深入的研究。我创建了工程 <https://www.malmgren.nl/Harmor/Harmor-48.flp>。这个工程中的第一部分包含的是默认衰减时间, 但是空间大小却是慢慢减小的。第二部分则是保持空间大小不变, 然后改变衰减时间。第二部分第一个音的衰减时间被设置成了最大, 也就是是 51200 毫秒! 接下来的音的衰减时间就会逐渐减小。



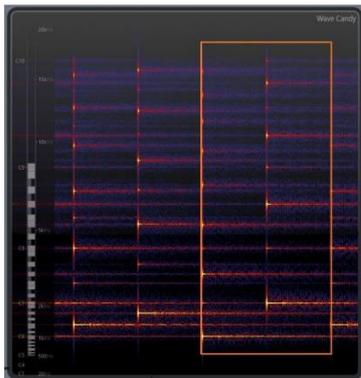
## 混响高衰减 (Reverb high damping)

在我们之前关于回音效果的章节中, 噢抱歉, 是“延迟”效果的章节中, 我们讨论了衰减的功能。如果我们打开了衰减开关的话, 重新被传递到每个回声周期的声音信号都会再次被滤波器过滤。混响也有一个些类似的系统, 但是只有高切效果。

用生活常识来理解的话，就是比较硬的表面会更加轻易的反射高频，而较柔软的表面则会吸收更多的高频声音，而不是将他们反弹回去。

## 混响扩散 (Reverb diffusion)

混响扩散，diff，控制的是声音的反射清晰程度。如果这个参数的值比较低，那么反射的声音就会非常清晰；如果值比较高，那么反射音中就会掺杂更多的杂音。低值的效果可以被理解为空间十分的空，而高值则说明空间中充满了不规则的表面。



这个工程 <https://www.malmgren.nl/Harmor/Harmor49.flp> 可以说明这个效果是怎么工作的。在这个工程中包含了一段十分简单的旋律。每一个小节中的混响扩散都会比上一小节低四分之一。在第四小节中，声音很明显是从墙上反射回来的，因为混响扩散的值非常低。如果你仔细的观察的话，你会发现在第四小节中声音稍微有点颤抖。

## 混响湿声音量 (Reverb wet volume)

混响湿声音量滑杆，wet，可以从-100%一直调到+100%。在 0%的时候，输出与输入是完全相同的。默认值是 50%，也就是说 50%的原声与音效的输出相混合。在+100%的时候，我们就只能听到混响音效了。

50%是一般的设置，其声音效果就像听者就站在合成器旁边一样。也许听者就正在用合成器演奏呢？在+100%的时候，差不多就像是听者站在距离合成器稍远一点的地方？

那么正值与负值之间的区别是什么呢？



上面左面的图像展示的是一个混响湿声音量在+50%的锯齿波, 右侧的是一个相同的锯齿波, 但是参数则是-50%。有可能是图像被翻转了? 并不是很确定。

但我们观察矢量显示器的时候, 我们遭遇到情况的也差不多。就是看起来像是图像从左边移到了右边, 但是他们的形状还是会有一丢丢的不同。

我建议你可以把玩一下混响中不同的设置, 然后体会一下不同设置下的音色听起来是怎么样, 并且在矢量显示器中看起来是怎么样, 祝你玩得开心!

你可以用这个工程来测试: <https://www.malmgren.nl/Harmor/Harmor-50.flp>

## 混响选项 (Reverb options)

混响选项的下拉菜单有 6 个混响预设。从这些不同的预设选项的参数中你能学到很多。这里是他们的参数:

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

在最后一行中, 我加入了默认设置值。

## 第 19 章 . 压缩 (Compression)

压缩器的原理很简单。假设我们有一个这样的仪器:



这个仪器有一个峰值表 (peak meter) 以及一个放大率旋钮 (amplification knob)。如果这个世界上有一种职业叫做压缩员的话, 那么他的工作就是: 当峰值表的绿长条超过橙色的线之后快速的降低放大率, 当绿长条低于橙色的线的话就将放大率调回原来的位

置。就这么简单。这肯定是一个十分无聊的工作!



压缩器要做的下一步就是将声信号的频率分成三个部分，低，中以及高。然后每个部分都有其自己的压缩器。现在事情开始变得有趣了。对于一个三部分的压缩器来说，他除了要保证对各部分超出某一阈值的音进行压缩以外，还要做更多。比如我们有可能加重低频的声音来突出高频。这也是压缩器所擅长做的。

## 压缩类型 (Compression type)

压缩类型菜单有 5 个选项：Off (关闭)，Limiting (限制)，Warming (加温)，Heating (加热)，Burning (燃烧)。我对这些的理解是这些名字都是混响中对室温的比拟，比如说当有更多低音的时候，空间就会“更热”。在这里，当压缩器给低频更多放大率的时候，我们称之为“变得更加温暖了”。

## 压缩量 (Compression amount)

当压缩类型被设置成 Limiting 的时候，压缩量这个滑杆就被关闭了。对于其他的压缩类型来说，这个滑杆控制的是这个效果有多么的强。

## 压缩带 (Compression band)：低，中，以及高

压缩量影响的是应用在声信号上的压缩效果有多强。但是如果这个功能不够用的话，我们就可以通过调整压缩带的滑杆来调整他们之间的“平衡”。

# 第 20 章 . 曲线编辑器 (Line editor)

Harmor 的官方文档将这部分称为“包络线或映射形状” (envelope or mapping shapes)。这也许是这部分的功能，但是我们现在要讲的是这些曲线的编辑问题。在下一章，我们才会讨论这些曲线所代表的不同功能。

你其实已经用过这个曲线编辑器很多次啦！曲线编辑器在 FLStudio 中无处不在：Parametric EQ，音色窗口 (Timbre windows)，Sytrus，自动包络线，WaveShaper 等等。实在是忍不住在你们自己已经理解了这个编辑器之后才开始讲这部分的知识。



图 49. 在这一章中，我们只讨论橙色区域内的编辑工具

## 点模式 (Point mode)

在这里你可以看见这些点模式的标志：

- step editing (步编辑) 。在背景中左键点击并拖动来创造刚才选定的点类型。右键然后拖动则是删除划过的点。这是删除许多点最方便的方法了。我建议你一定要记下这个小技巧！
- Freeze editing (冷冻编辑) 。冻结整张图来防止被意外编辑
- Snap to grid (对齐网格) 。将点对齐网格
- Slide succeeding points (滑动成功点) 。使用这个来保护位于正在被拖动的点的右侧的图。(译者注：如果开启这个功能，那么当拖动一个点的时候，这个点左侧的图也会跟着一起动)

## 曲线的基本操作 (Basic line operations)

下面列举的是在脚步编辑关闭  的情况下的线的基本操作：

- 如果想创造一条线段，就右键背景界面。如果想插入一个点就右键点击这条线段
- 如果想将一个点向左移动就点击并且拖拽这个点。如果想垂直移动就按住 shift 键，如果想水平移动就按住 alt 键。
- 右键一个点然后选择删除以删除这个点。你也可以按住 alt 然后左键这个点来删除
- 右键一个点然后选择复制这个点的垂直水平。右键另一个点然后选择粘贴。这样的话第二个点就会跟第一个点处于同一个水平位置了。
- 右键一个点然后选择线段类型
- 在一条线段的中间点左右你能找到张力把手。左键然后拖拽以改变张力。这个功能的实际作用是什么取决于线段的类型。
- 右键张力把手以重新设置该张力把手
- 按住 control 键然后左键并拖拽张力把手来微调该张力把手

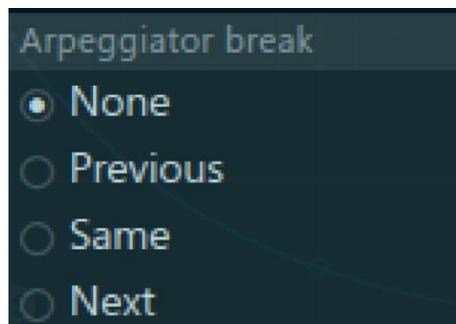
## 线段类型 (Line segment types)

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)	

取决于张力点的设定，波线可以有两种不同的形状

## 琵琶音点 (Arpeggiator points)

当一个图的横轴是时间的时候，这个图就可以用来描述一个随时间的变化。当我们在 midi 键盘上同时弹奏多个音的时候，琵琶音就会被激活。正常情况下如果你在钢琴窗中写了一个和弦，那么 Harmor 会同时奏响所有的音。当打开琵琶音阻断 (arpeggiator break) 的时候，Harmor 就会开始在该和弦的音与音之间来回横跳。所以我们可以通过琵琶音阻断来决定 Harmor 是如何在和弦之间跳跃的。

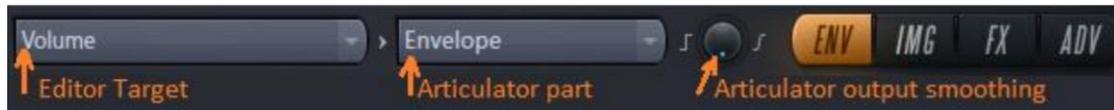


- None (没有)，没有跳跃
- Previous (上一个)，跳到音调更低的一个音
- Same (相同)，演奏同一个音
- Next (下一个)，跳到音调更高的一个音

## 第 21 章 . 目标的基本知识 (Target basics)

直到现在为止，我们已经讨论了 Harmor 中不同部分的默认功能。我们要么手动的调整不同的旋钮，要么通过自动化包络线的帮助。现在我们要学更多的方法来修改 Harmor 的工作成果。

在 ENV 标签中你可以找到两个下拉菜单。

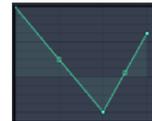


如果你点击 Editor target（编辑器目标）那个下拉菜单，就会出现 3 列从左到右排列好的 Harmor 的工作目标：

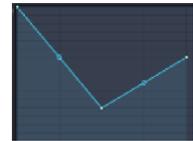
- Articulation（发声）。这些目标包含了大部分的功能：11 个选项，以及 articulator output smoothing（发生器输出平滑化）。Articulator part（发生器部分）的窗口有 LFO，Envelope 界面有 ADSR 控制器等等。
- Mapping（映射）。对于这些目标来说，Articulator part 的下拉菜单中有 9 个选项。不包括 LFO 以及包络线。发声具有输出平滑化的功能。
- Shaping（塑性）。这些目标没有任何 Articulator part 的下拉菜单内容，而且没有 Articulator output smoothing 功能。

我第一次打开 Editor target 的下拉菜单的时候，我以为工具提示栏坏掉了，因为它只显示三个不同的值！当你的鼠标划过 Editor target 的菜单，以及 articulation（发声）和 mapping（映射）的目标选项的时候，你会发现在工具提示框中有三个不同的文字。

1. 合成模式：双极加和（Combining mode: Bipolar）。这个图的 0% 位于中间。中轴线上方的正值，下方是负值。阴影是围绕着中轴线产生的。在下面我会用“BΣ”这个福海来表示这种合成模式。



2. 合成模式：单极乘积（Combining mode: Unipolar product）。这个图的 0% 位于底部，然后底部以上的所有值都是正值。阴影是由底部和曲线围成的。下面我会用“Ux”来表示这种合成模式。



3. 合成模式：双极乘积（Combining mode: Bipolar product）。阴影是围绕着中间产生的，与双极加和的模式一样。下面我会用“Bx”来表示这种模式。



（译者注：这里讲的是当你选择了 target 之后会看到的图像框架）

## Editor target 下拉菜单

这是 Editor target 的下拉菜单。点击这个表格中的任何一项可以将你传送至讲解这一项的部分。

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

这里是发生器部分的下拉菜单。这里的每一个部分在这章的后面都有其专门的讲解

- Envelope（包络线）（只能在发生器以及映射功能中使用）。这部分描述的是冲击，衰减时长，持续以及释放的设置，也就是 ADSR
- LFO（只能在发生器以及映射功能中使用）。这一部分是用控制低频振荡器的。
- Keyboard mapping（键盘映射）。这部分的功能就是将数值与正在被演奏的音符连接起来。
- velocity mapping（速度映射）。这一部分是将功能与速度映射连接起来。
- Modulation X, Y and mapping（调节 X, Y 以及 X 映射）。X, Y 以及 Z 映射将 MOD 控制器与 Harmor 的各个功能连接了起来
- Random mapping（随机映射）。将旋钮调整到一个随机的数值
- Unison index mapping（齐奏指数映射）
- Held index mapping（保持指数映射）
- Mod X speed mapping（Mod X 速度映射）



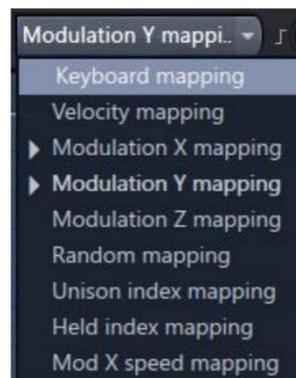
## 合成模式：单极乘积（Combining mode: Unipolar product）

单极乘积的 Articulator part 的图像有两个维度的图像。X 轴是固定的，而声音设计师则会设置 y 轴的值。最终的值就是所有 articulator 图像中 y 轴的最小值。现在事情会开始变得非常抽象，所以让我们先来看一个例子。



为了以通俗的方式来解释这个合成模式，我会使用一个虚构的概念，“smoot”。Smoot 与声音设计没有任何关系，所以特别适合用来解释单极乘积背后的数学原理。假设 Harmor 有一个旋钮，其值可以是 0 到 364.4 中的任何数。那么这就是一个单极乘积旋钮。

Smoot 旋钮可以在用户界面手动的调节，或者用自动化包络线来控制。这样就可以在发生器部分中设置好最大值。如果 Harmor 有一个像这样的旋钮，工具提示框就会显示 smoot 的具体数值，但是如果我们用自动化包络线来对其进行控制的话，自动化包络线上的点就会用 0 到 100% 的方式来表示其 0 到 364.4 的这个范围。



现在假设我们的 smoot 旋钮是一种“映射”。那么 articulation part 的菜单就会像右边图中这样。当这个菜单中的某一项的左侧出现一个小三角的时候，这就说明这一项参与到了改变音色的过程中去了，也就是被激活了。

下面展示的就是 Modulation X 以及 Modulation Y mapping:



当我们调节 articulator 图像的时候，我们很容易会忘记设置 smoot 旋钮，但是 smoot 旋钮却在一直输出其最大值。假设这里的 Oliver Smoot Position 旋钮被设置成了 50%，那么其相对应的最大值就是 188.2。

当我们将这两张图像半透明地叠加在一起的时候，我们很容易就会发现，这个例子中的所有点的最小值变成了一条直线！



## 合成模式：双极乘积 (Combination mode: Bipolar product)

假设我们在 Harmor 中有另一个 Smoot 按钮。由于这是双极的，所以中间有一个 0。也就是说+182.2 在正的一边，而-182.2 则在负的一边。我们已经讲过的每一个知识在这里依然成立。图像的中点也同样是旋钮的中点。

那么负的一边的情况呢？最小的值由旋钮设定，也就是图像中的中点。

## 合成模式：双极加和 (Combination mode: Bipolar sum)

这个模式中使用的是每个 articulator part 的绝对值，然后结果就是 y 值的加和

## 发生器部分 (Articulator part)：包络线 (Envelope)

当你第一次打开 Harmor 的时候，这就是你会看到的第一个包络线。这是音量包络线。这是 FLStudio 中很常见的一种控制器，所以有可能你之前在其他合成其中已经见过了。



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

Envelope 中的发生器部分可以编辑声音的冲击 (attack), 衰弱 (decay), 持续 (sustain), 以及释放 (release), 或者简写就是 ADSR。通过打开左下角的那个开关你就激活了这个包络线。默认的图像表示的就是不同 ADSR 部分的默认张力 (tension)。你可以按照你所想的调整其张力，或者直接使用默认值。图像下面的 ADSR 旋钮控制着整张默认图像。你当然也可以添加更多的点，只要是原本的点能够依然使得 ADSR 旋钮工作。如果你，比如说，移除了处在 S 那条线上的点，那么默认旋钮就不知道该做什么了，直到你再添加一个新的点，然后再右键，然后选择 sustain/loop end.

Tempo 开关可以将图像中的时间替换成整个工程的总时长

当 global 这个开关被关闭时，所有的音符都有了他自己的包络线。单独的音会根据他自己的包络线被演奏出来。当 global 这个开关被开启以后，每一个音都会以与其他音相同的包络线被演奏出来。

我们可以将一个音频片段拖拽到音量包络线窗口中去。这跟打开选项菜单然后选择 Analyze audio file 是等价的……



图 50.VOC\_138InProgress 被拖拽进 Envelope 窗口之后的图像

这里我将“VOC\_138InProgress”这个音频片段拖进了 Envelope 中，你可以在 Packs->Legacy->Vocals 中找到这个片段。当我们在分析音频的时候，如果有需要的话移除一些点是特别有帮助的。你可以使用选项菜单中的工具。这些工具都在下面被解释的很清楚：

## Envelope 窗口选项菜单：Make all loop

这个功能将第一个点设置成了 Loop start，最后一个点设置成了 Sustain/loop end.

## Envelope 窗口选项菜单：Insert points at markers

这个菜单中的各个项总是灰色的

## Envelope 窗口选项菜单：Flip vertically

这个功能可以将图像进行上下颠倒。

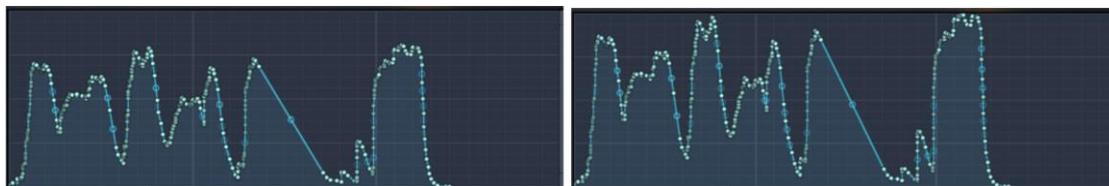
## Envelope 窗口选项菜单：Scale levels

这个功能可以打开一个比例窗口。Offset 滑杆可以将图像向上或者向下移动。当这些点走不动的时候，他们就会卡在顶部或者底模。滑杆 multiply 可以将图像拉长或者压缩。滑杆中间的下方是可以将图像进行垂直翻转-拉伸或者压缩的。Tension 旋钮以及 Center 滑杆可以一起工作，我一会儿将会列举一个他们的例子。假设你先用 multiply 滑杆将图像压缩，你可以用 offset 将图像向上移动至中间。然后当 Tension 被设置成 0%的时候，Center 就会将图像相对于 Offset 移动向上或者向下移动。如果你将 Tension 设置成其他的值的话，移动 Center 你会看到图像是如何被同时拉伸以及压缩的了。



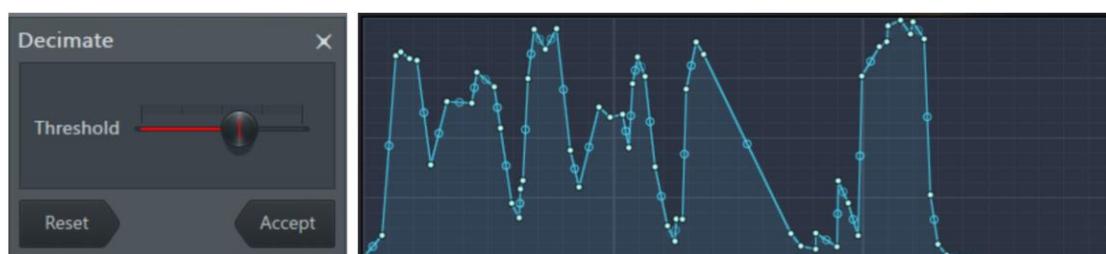
## Envelope 窗口选项菜单： Normalize levels

假设你的图像在垂直方向上没有占用所有的窗口，那么这个功能就是去将你的图像拉伸，以至于使得图像充满整个窗口



## Envelope 窗口选项菜单： Decimate points

如果你的图像中的点太多你可以使用这个功能来移除这些点



## Envelope 窗口选项菜单： Filter...

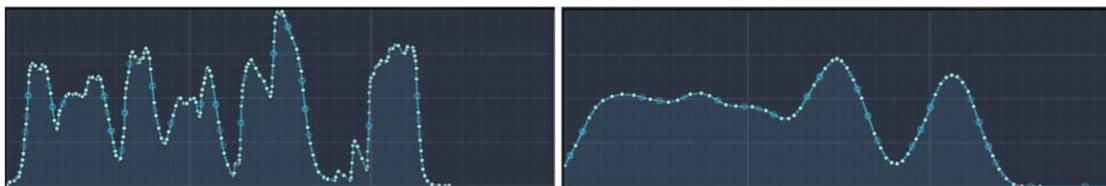
假设我们从图 50 的声音开始，然后激活滤波器功能。默认的滤波窗口请看下面这张图。



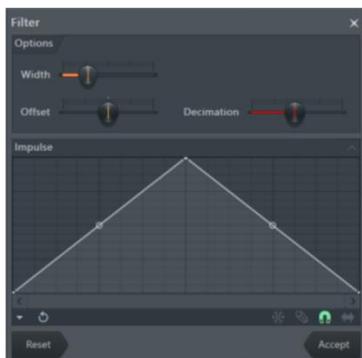
滤波的结果被直接展示了出来。当我们将滑杆 Decimation 拖向左侧时，我们保证了图像的形状，但是我们却加入了更多的点。但我们向右拖拽的时候，这些点就会减少。直到这张图看起来与之前十分不一样为止。



滑杆 Offset 可以将整个图像像左或者向右移动。滑杆 width 的作用是用来设置滤波包络线。现在有好多包络线啊！当我们将 width 滑杆拉向左侧的时候，原声就会出现，拉到最右侧的时候，整个图像就会被更多的过滤。

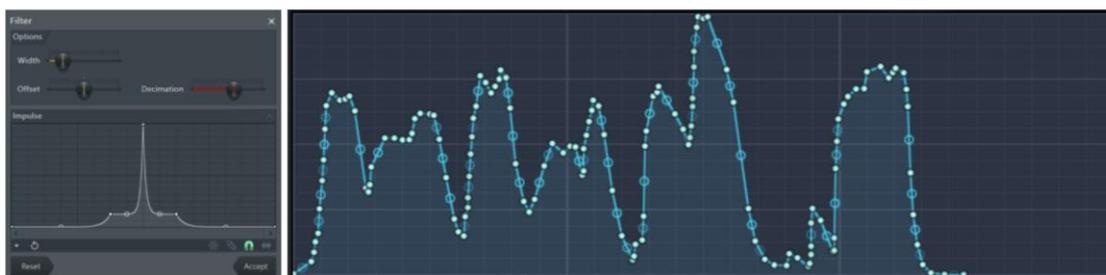


当你点击 impulse 后面的下拉箭头的时候……一个新的功能就会出现！



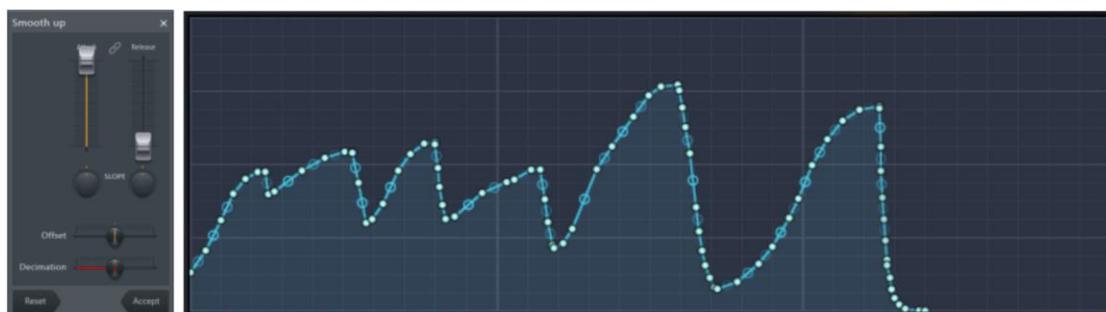
这是脉冲曲线，这条曲线可以用来过滤整个图像。我对这个功能的理解是，我们可以将脉冲的形状移动到原图上，然后落在脉冲曲线下方的点都会被用来计算一个均值。并不是说所有的点的权重都是相同的。每个点的权重都是由它距离脉冲图像底部的距离决定的。底部中间的点的权重是最高的，然后距离这个点越远，权重就越低。这个脉冲曲线在原图上一点一点地扫过，每移动一次，就计算一次均值。滑杆 width 控制的是脉冲曲线的宽度。一个比较窄的脉冲曲线会在计算的时候纳入较少数量的点，而比较宽的脉冲曲线则会将更多的点带入到每次计算之中——所以会产生更加圆滑的曲线。

你可以点击 Impulse 来打开已经预制好的脉冲曲线。你当然也可以对其进行再编辑。下面你可以看到我的试验。

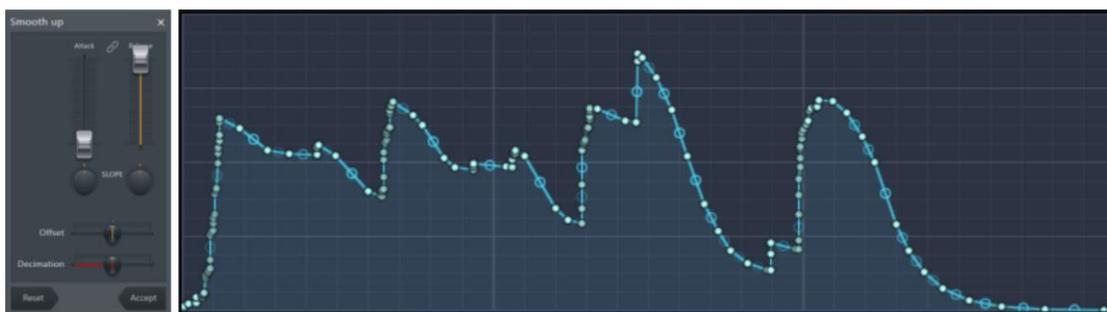


## Envelope 窗口选项菜单：Smooth up…

这也是一个滤波的一种。我们以图 50 为例，假设我们激活了 smooth up 功能，那个功能就是将图像变得更加平滑。同时根据你对冲击以及释放时间的要求，图像会有不同的变化。比如如果你将冲击 (attack) 设置到最大，释放 (release) 到最小，你会看到图像的增进 (build up) 会非常慢，但是释放 (release) 却会很短。



当我们把条件反转过来，你就会看到非常快速的冲击，以及更长的释放。



在 attach 以及 release 滑杆下方你可以看到 SLOPE 旋钮。当旋钮指向右侧的时候，他们就会放大他们被影响的部分。工具提示栏中没有显示其具体数值，这是我的猜测。当旋钮指向左边的时候，他们被影响的部分则会变得更加“安静”。

滑杆 Offset 可以将整个图像向右或者向左移动。

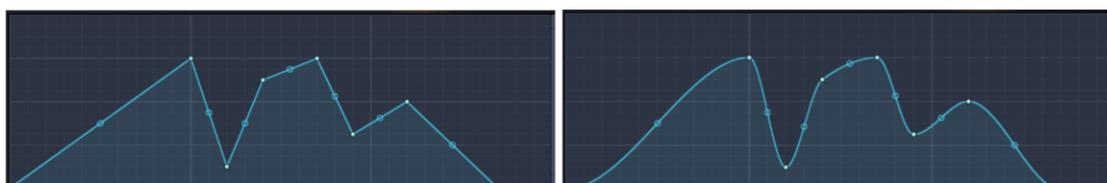
当向左侧拖拽滑杆 Decimation 的时候，更多的点会被加入，而向右侧的时候，图像上点则会被渐渐移除。

## Envelope 窗口选项菜单：Smooth up abrupt changes

这个功能可以将十分突兀的变化平滑化

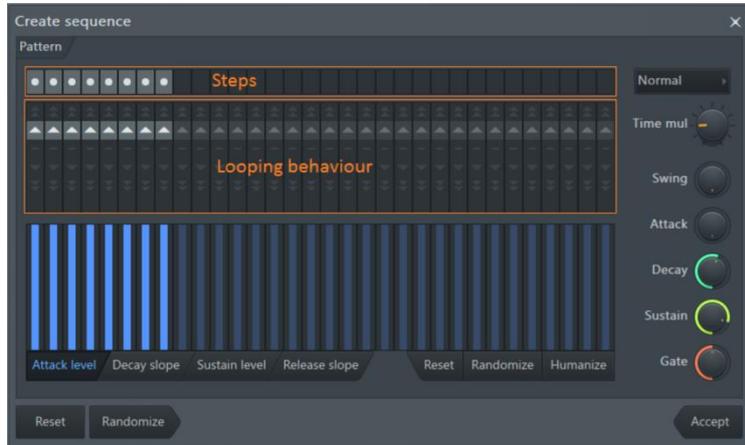
## Envelope 窗口选项菜单：Turn all points smooth

这个功能可以将所有的线段变得更加平滑



## Envelope 窗口选项菜单： Create sequence

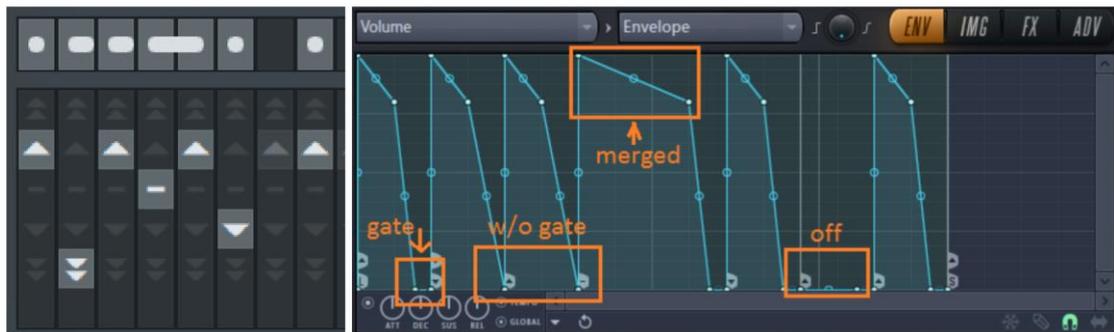
这个工具可以自动化编辑琶音音阻断（arpeggiator breaks）。如果你想试试这个怎么用的话，我建议你在 Harmor 的钢琴窗中建立一个比较长的和弦。然后弹奏这个和弦，然后打开 Harmor，然后开启 create sequence 功能。你用这个工具创造的新序列会使基于网格之间的空隙的，就像你在窗口中定义的那样。



在上面这张图中你可以看到创建新序列的工具窗口是长什么样的。顶部的那一行包含的是 steps，每一个 step 都有不同的状态：

- On (开启), step with gate
- Step without gate (无间隙)
- Step merged with next (合并)
- Off (关闭)

你可以通过重复点击其中的 step 图标来在其状态之间进行切换。右键来切换回上一状态。下面是一个例子，以及其根据其创造出来的图像。



该工具下一个区域的作用是控制循环播放的模式。你可以将其设置成下列中任何一个：

- 演奏和弦中下下个更高的音 ，也就是跳过一个音
- 演奏和弦中下个更高的音
- 演奏上一个音
- 演奏和弦中下个更低的音

- 演奏和弦中下下个更低的音 ，也就是跳过一个音
- 关闭，这个时候图标就会变成灰色。演奏和弦中所有的音  
你可以对每个 step 中的冲击（attack）水平进行设置。



你可以对每一步的衰减（decay）斜率进行设置。



你可以对每一步的持续（sustain）时间进行设置。



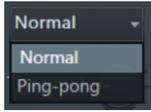
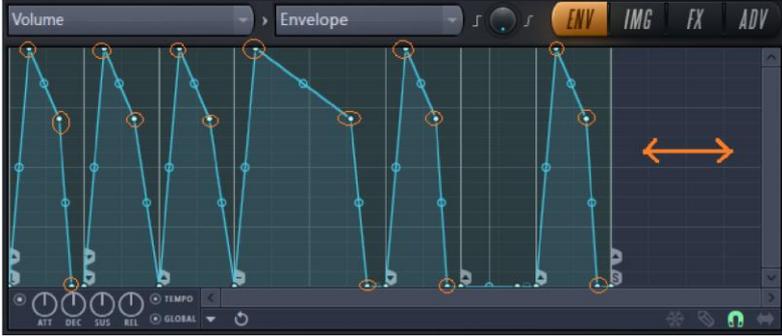
你可以对每一步的释放（release）斜率进行设置。

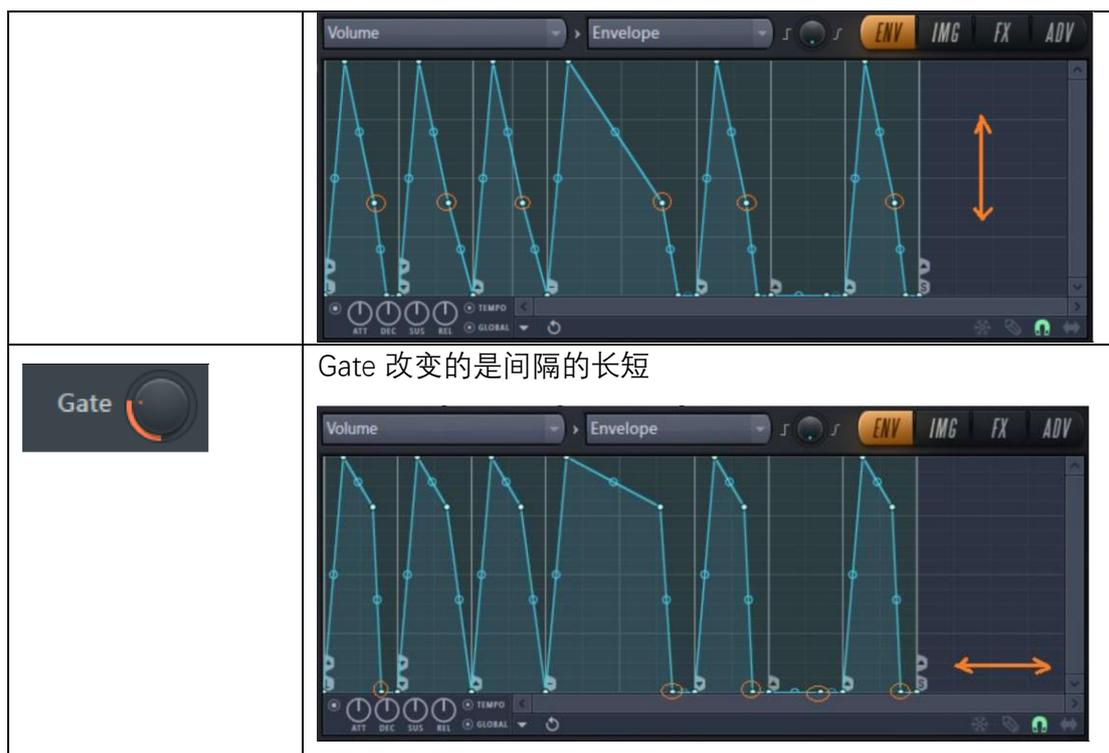


有一个恢复默认设置按钮，如果你希望恢复 release, sustain, decay 以及 attack 的默认设置的话，就可以用这个。随机按钮可以将设置随机化。



你可以通过在设置中添加一些微小的随机性将整个和弦变得更加自然

	<p>右上角有一个模式菜单，默认值是正常。你也可以选择 ping pong 模式，在这个模式中，整个模式会被复制一遍，然后再将该模式的第二个复制品反转过来</p>
	<p>swing 可以让每个不均匀的 step 变得更长，然后均匀的 step 变得更短</p> 
	<p>这个旋钮，也就是时间乘数，可以将时间的单位改变。在默认设置下，时间是被乘以 1 的，而这个旋钮则可以改变这个比例</p>
	<p>Attack 这个旋钮可以将每一步的冲击变得更长或者更短。但是这个旋钮也会将剩下的点之间的距离变得更短</p> 
	<p>Decay 控制的是每一步的衰减时间</p> 
	<p>Sustain 旋钮可以将图像中控制 sustain 的点上下移动</p>



Gate 改变的是间隔的长短

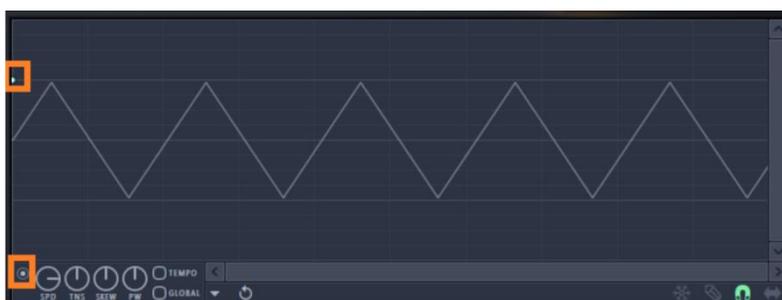
在窗口的最下方你可以找到下列的按钮：

- 重置 (reset)。将模式重置成默认模式
- 随机 (Random)。设置一个随机模式
- 接受 (Accept)。接受目前的设定，并关闭窗口。

如果你不想使用这个模式，那么你就用右上角的叉叉关掉这个窗口

## Articulator part: 低频振荡器 (LFO)

在第 10 章 *Tremolo* 以及第 11 章 *Vibrato* 中，我们讨论了两个创建低频振荡器的方法。LFO 是第三个方法。在默认情况中，LFO 的发生器是关闭着的。如果你改变了其中的任何旋钮，左下角的开关就会自动开开。当然你也可以点击那个图标来开启。



开关 tempo 以及 global 的功能与之前讨论过的一样。

在图像的左侧你可以看到一个点，这个点控制的是 LFO 图像的振幅。你可以将这个点上下移动来改变振幅。如果你把这个点移动到中轴线以下，整个图像就会产生一个 180 度的相位差。如果你从这个点出发画一条线，这条线将会决定整个图像随时间变化的振幅。

图像下面有 4 个旋钮：

- LFO 速度, SPD; 控制的是 LFO 图像的频率
- LFO 张力, TNS; 当 SKEW 是 0% 的时候, 0% 的 LFO 张力就会产生一个三角形的 LFO 图像
- LFO 倾斜度, SKEW; 可以将图像倾斜。当是 -100% 的时候, LFO 的图像就会变成“N”的形状, 也就是一个锯齿波。当是 +100% 的时候, 你就会得到一个 M 形的波。我是在斯拉夫字母表中找到这个 M 符号的!
- LFO 脉冲宽度, PW; 当是 0% 的时候, 波形的顶部以及底部的宽度是一样的。当 PW 是负值的时候, 底部的宽度就会大于顶部, 正值的时候则是顶部大于底部。



在 Harmor 的主要部分你可以找到这个 LFO 旋钮。你可以用这个旋钮同时改变 Harmor 中所有的 LFO 效果量。

## Articulator 部分: Keyboard mapping

Articulator 部分中的键盘映射可以让你根据正在弹奏的音来设置某个值。为什么这个不叫做音符映射呢? 我不晓得。你可以在你的鼠标光标划过键盘映射图的时候看一眼工具提示框, 因为这会方便于你知晓你的光标指的是哪个音符

## Articulator 部分: Velocity mapping

在钢琴窗中每个音符都有一个速度值。在 Harmor 中也可以对这些速度值进行编辑。下面的图中你可以看到钢琴框中不同的速度值。



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

X, Y 以及 Z 调整控制器可以提供一个标准化的方法来改变合成器中的各种参数。一开始这个控制器是二维的, 但是当你改变 Z 这个旋钮的时候, 整个网格都会朝你移动或者是远离。在以后的第 26 章预设 (preset), 中我们会提到, 信息窗口会对 X, Y 以及 Z 的功能做一个简短的介绍。当你看到这些信息之后, 你就会更加了解 Harmor 中不同的预设, 你也会知道你可以怎么对 X, Y 和 Z 进行编辑。



将旋钮与 X, Y 或者 Z 连接起来的最简单的办法就是右键旋钮, 然后选择“Quick amp to”, 然后在出现的菜单中选择你希望连接的旋钮。并不是所有的旋钮都有这个菜单选项

## Articulator 部分: Random mapping

这一部分中编辑图像的方法与我们在这一章的介绍中讨论的一样, 但是图像的 x 轴好像

没有什么用，只有图像的 y 轴有用。这一部分的功能就是根据合成模式的规则来随机一个 y 值。（译者注：你可以在其中添加几个点，然后看看他们的值，你会发现这些值都是随机的，与其处在 y 轴的哪个位置无关）

## Articulator 部分：Unison index mapping

我们在第三章合唱 (Chorus) 中详细的介绍了合唱。但编辑这一部分之前，你需要先设置好合唱的阶数。合唱效果中的阶数将决定这图像中 x 轴上的线的数量。X 轴上的每一条线都代表着合唱的设置。对于 y 轴来说，则会按照合成模式的规则工作。

## Articulator 部分：Held index mapping

这一部分可以使我们创造 held notes and index，然后让它可以与 index 映射起来。一个 held note（延音）是一个与另一个音互相重叠，但是并不是在同一时间被开始演奏出来的音。只要音与音之间有空隙，index 就会重新再来一遍。

如果你使用 midi 键盘来写一个和弦，那么 index 映射就不会被激活，因为所有的音都是同时被开始演奏的。如果你在 midi 中用弹拨音写一个和弦，而且他们都是在不同时间被弹奏出来，那么 index 就会被激活

X 轴上的每一条线都代表着一个 index step

## Articulator 部分：Mod X speed mapping

在这个 Articulator 部分中，我们可以将 mod x 的移动速度映射出来！为什么没有 Y 与 Z 的映射呢？我不知道。

## Articulator 输出平滑化 (Output smoothing)

发声器的右侧有一个 Articulator 输出平滑化的旋钮



这个滤波器与我们之前在音量包络线 (volume envelope) 中讲的滤波器差不多：Envelope 窗口选项菜单：Filter…。这个旋钮设置的是那个用来计算均值的脉冲图的宽度。较窄的宽度会保留更多原声信号的特点，而较宽的宽度则会使整个声信号的图像变得更加圆润。

## 第 22 章 . 图像合成 (Image synthesis) / 再合成 (Resynthesis)

当一个音被以更快或更慢的速度被演奏出来的时候，有很多传统算法来保护这个声信号的音调。声信号在分析框架中会被切成一片一片的，然后每个片段都会被重叠起来，最后再被缝补起来。在 Harmor 中，所有的这种传统不完美都会被移除，因为加和引擎可以将每个音在任何速度下都能控制的很好。这就是加和引擎的好处，我们可以利用这一点做很多很酷的事情！

IMG 标签提供了两种方式来加工声音：

- 再合成 Resynthesis。在这个模式下，你可以看到图像设置菜单，以及“plane(s) to work on”下拉菜单。



在这种模式下，sharp 按钮就被激活了。



在这种模式下，选项菜单的第三以及第四部分都变成了灰色。

- 图片合成 Image synthesis：这个模式与再合成模式很相似，但是在这种模式中，你可以看到另两个按钮，copy 和 paste。

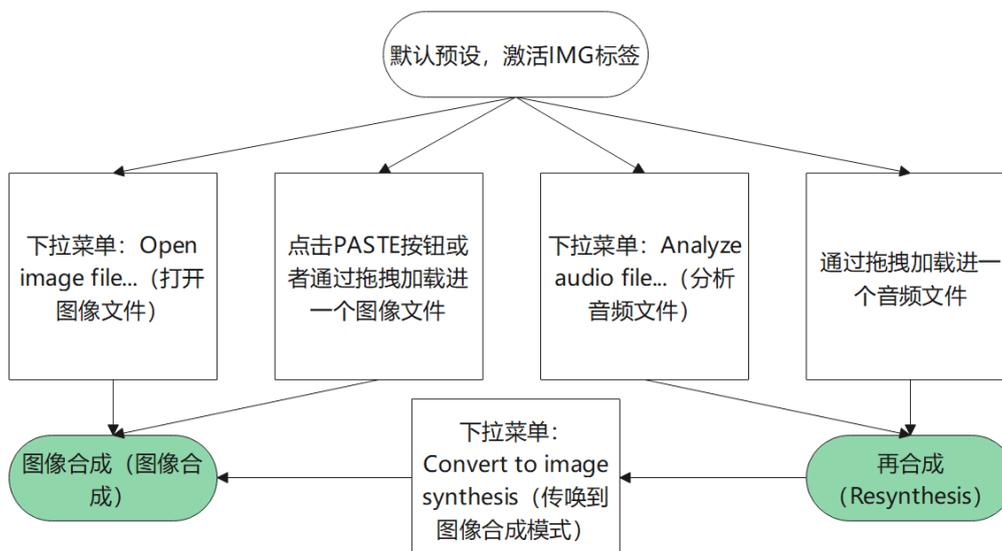


在这种模式中，sharp 按钮没有被激活。



在这种模式下，选项菜单中的 image 以及 selected plane 被激活了。

Harmor 中没有开关使得可以在两种模式下相互切换。原因是因为再合成模式将相位数据与“图像数据”记录在了一起。换句话说，再分析模式记录了声音内部的高分辨率数据。而图片合成模式则是一个更加简单的分析。我们可以从高分辨率走向低分辨率，但是却不能反过来从低分辨率走向高分辨率。下面的图展示了你如何激活两种模式：



当从再合成模式调整到图片合成的模式时，和声相位信息就被注入到和声相位窗口 (harmonic phase window) 中。

我们过一会儿会详细讲下拉菜单，但是现在我们会先讨论图片合成/再合成的其他部分。

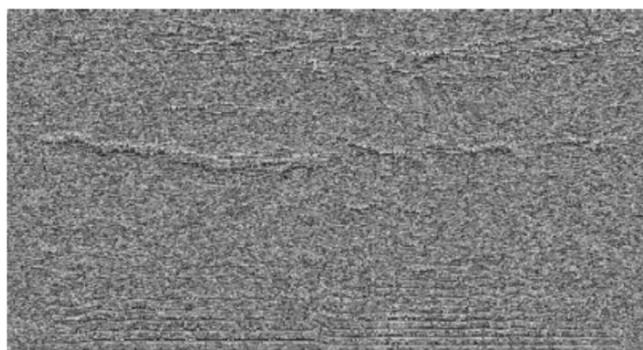
## 查看模式 (View mode): plane(s) to work on

在标签 PLANE 旁边我们可以找到一个下拉菜单，使用菜单中的选项我们可以设置 plane 查看模式。这里有三种查看模式：

- Gain，展示的是增益图像。如果你分析一个音频文件，那么通常会看到一个像频谱一样的图像。这就是这张图片是怎么被制作出来的。最低的频率在底部。最高的频率则在顶部。X 轴则是时间。
- Freq，展示的是频谱。通常当你分析一个音频文件的时候，频率面板将会充满灰色像素。有时会有稍微有点黑，有时又会有点白，但是大部分都是灰色的像素。
- All，将 gain 以及 freq 面板结合起来。

增益面板展示的是哪些小振荡器将会工作，默认振幅是多少，以及工作的时间点。水平的 x 轴控制的是时间，垂直的 y 轴中，每个像素会决定哪个小振荡器会被影响。黑色的像素代表没有声音，白色的像素代表的是有声音。

频率面板可以改变增益面板中小振荡器的震动频率。50%的灰色代表着不改变任何频率。比灰色更黑的像素的震动频率究竟是更高还是更低则取决于频率设定。当你在分析一个音频文件的时候，你会发现频率面板会变灰，虽然有可能会有会一两个稍浅一些或者深一些的条带。下面的图像展示的是 Laurie



Webb Falling 这个音频文件的频率面板。你可以在 Packs->Vocals 中找到这个音频文件。

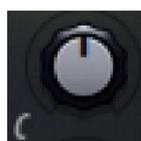
在面板的右侧你可以找到再合成/图像合成控制器。这些控制器在两种模式下都会以相同的方式工作——除了 sharp 旋钮，这个旋钮只会在再合成模式下工作。让我们看看下面这些不同的旋钮吧。

## 图像微速度 (Image fine speed)



这个旋钮，SPEED，控制的是图像的播放速度。旋钮的右侧是 0%到 200%，这时是正向播放。左侧是 0%到-200%，这时是反向播放。

## 图像过程速度 (Image course speed)



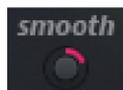
这个旋钮没有任何标签。这个旋钮的位置在 SPEED 旋钮的上方，左侧有一个小 C。这个旋钮的作用是去增加或者减小图像微速度旋钮。默认的设置是 1.000。当 knob 被拧到右侧的时候，这个 knob 会将图像微速度乘以一个 1 到 64 的一个值，而当被拧到左侧的时候，这个 knob 会将图像微速度乘以一个 1 到 0.0016 之间的一个值。

## 图像时间偏移 (Image time offset)

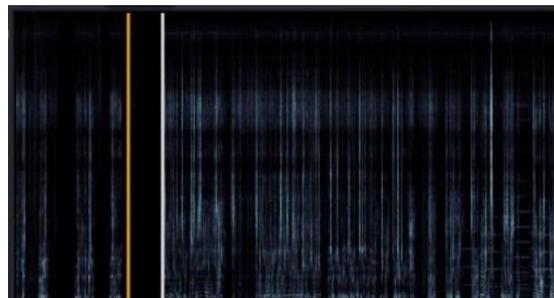


这个旋钮，TIME，控制的是开始播放图像的位置。你也可以在图像中左键点击来设置起始点。现在如果我们在 Harmor 的钢琴窗中加入一个音符的话，那么当你播放这个样式的时候，在 Harmor 中你会看到一个红色的垂直的线，以及一条在图像上移动的白色的线，这条线代表的是正在播放的图片的位置。如果你将速度设置到一个正值，那条白线就会正向移动到右侧，如果是负值的话就会向逆向移动到左侧。

## 图像时间偏移平滑化 (Image time offset smoothing)

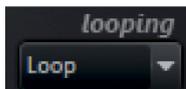


这个旋钮, smooth, 是 Harmor 中最小的旋钮之一! 这个旋钮十分的小, 以至于看起来像一个开关。这个旋钮控制的是图像时间偏移的变化会多么平滑。如果我们将速度 (speed) 设置到 0, 那么我们就可以尝试着对偏移平滑化这个功能做试验研究。将光标放在图像中, 然后拖动。你会发现白色的光标会紧密跟随着黄色的光标。现在我们将时间偏移旋钮拧到左侧, 然后再拖拽图像, 我们会发现这条黄线会跟随光标, 但是白色的线却会出现延迟。



当在右侧的时候, 平滑化旋钮与在左侧时相似, 除了当白色以及黄色之间的距离变大的时候, 黄色的线会直接跳到白色的线那里。

## 循环模式 (Looping mode)



循环模式控制的是循环是怎么产生的。在默认模式下, 整个窗口确定了循环的开始, 以及结束的点。在后面我们会看到这个循环区域是怎么被重新定义的。循环模式的下拉菜单, looping, 提供了三个选项:

- One-shot (一次性): 从音频开始的地方开始播放, 然后在音频的终点结束。
- Loop (循环): 在起始位置开始播放, 然后再从音频开始的地方继续播放
- Ping pong (来回): 在起始位置开始播放, 然后当到达终点的时候将播放方向反过来, 从最后再开始播放, 然后回到起始位置。

## 图像改善 (Image sharpening)



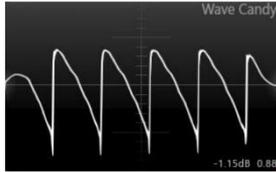
这个旋钮叫做图像改善旋钮, 其标签是 sharp, 但是他只在再合成的模式下被激活。为什么他不叫做再合成改善旋钮呢? 我不知道。

当旋钮被拧到左侧时, 旋钮控制的是 0%到 100%的瞬间改善 (transients only sharpening)。而在右侧时, 则是 0%-100%的瞬间以及相位改善 (transients and phase sharpening)。



如果我们录下一段完美的锯齿波音频, 然后在用 Harmor 的再合成模式播放出来的话, 那么其波形几乎不会受到任何的变化-100%瞬间, 当然这是在保证其他旋钮在默认位置的时候。

0%也可以很好地再生成输入进去的音频。



在右侧的 100% 的时候，则会改善瞬间以及相位。这样你就会观察到示波器的上下两部分展现的是一个被滤波器过滤过的信号——波的边边圆润了。

我发现了一个特性，应该是个程序上的 bug。当我们把旋钮完全拧到最右边的时候，这时的改善设定对音频没有任何的影响。对于所有其他的设定来说，当录音的重放到头的时候，根据循环模式的不同，要么这段录音会重新再从头播放一遍，要么会再倒着播放一遍，也就是说改善旋钮在内部会将其一直拧到右侧。如果重复播放通过了起始点，那么改善设定会退回到我们设置的地方。

## Level 分区



再合成/图像合成的这部分与增益平面有关

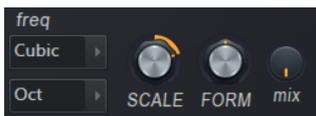
## 图像增益像素比例 (Image gain pixel scale)

这个旋钮，SCALE，控制的是增益平面中像素的大小。50%是默认值，当大于 50% 的时候，声音会更加安静，当小于 50% 的时候，声音就会更大。

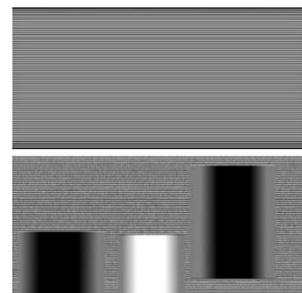
## 图像增益混合 (Image gain mix)

这个旋钮，MIX，控制的是频率平面 (frequency plane) 是怎样与增益平面 (gain plane) 相混合的。正常情况下，其被设定为 100%，这也是我觉得最好的设定。当旋钮的值被降低的时候，频率平面就会被放大。如果你将一个有用的图像加载进这里，那么说不定这个功能会很有用。

## 频率分区 (Frequency section)



频率分区调整的是小振荡器的频率。当是 50% 的灰色的时候，没有任何调整发生。





上图中展示的是一个锯齿波的增益图。下图则是一个频率图来帮助我们理解其背后的原理。当频率图渐渐的从灰到黑或者从灰到白来改变其数值的时候，我们可以看到所发生的一切。如果我仅仅只插入一个黑色的区块或者一个白色的区块的话，我们将更难看到其过程。

## 图像频率像素比例 (Image frequency pixel scale)

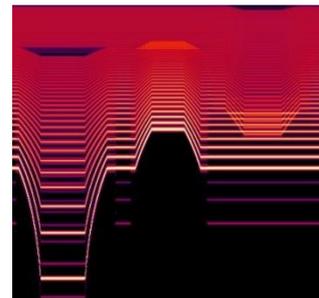
图像频率像素比例的旋钮，SCALE，的默认频率是+100%。当这个旋钮指向中间的时候，是 0%，在右侧的时候最大值是+200%，在左侧的时候最小值是-200%

在默认位置中，也就是 100%的时候，50%的灰色更暗的像素会降低那个小振荡器的频率。比 50%的灰色更亮的像素会使得频率更高。当比例旋钮被设成一个负值的时候，情况就会相反：更明亮的像素会降低频率，而更暗的像素则会增加频率。

## 图像频率模式 (Image frequency mode)

图像频率模式的下拉菜单可以让我们改变改变频率的方式。默认选项是 Oct 模式。其选项有：

- Oct，这是最常见的模式。更低的频率会在这个模式下有更大的变动，而更高的频率变动就会更小。
- Hz。在这个模式下，所有的频率都会同时移动
- 宽 Hz。下面的图就是宽 Hz 模式
- Unison。在这个模式下，图像会控制 Unison 的失谐。你可以打开 Unison 来试试这个效果。我必须说，这个音效非常难以注意到，但是其的确有一些区别。



## 图像共振峰漂移 (Image formant shift)

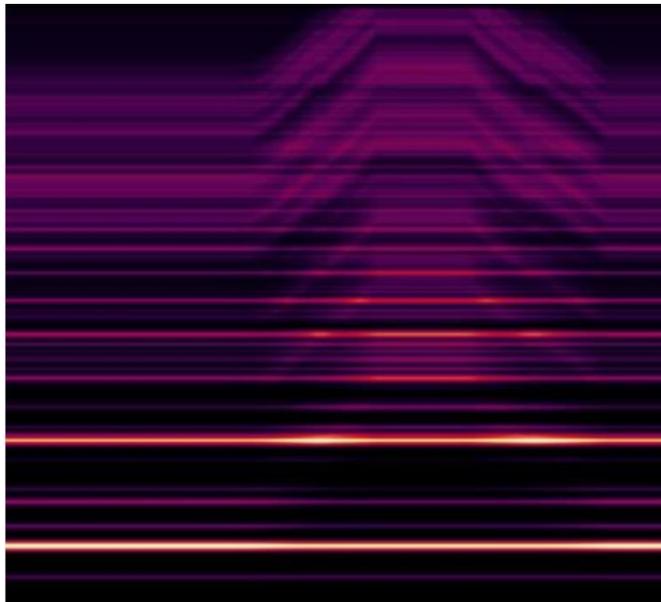
当我们改变合成声音的音调的时候，频率就会升高或者降低。但是对于人来说就不一样了。当一个歌手改变他的音调的时候有些频率不会升高也不会下降。这是因为人类的声道有一个特殊的形状。

鼻腔可以与声音产生共振，然后创造出一种特殊的声音，这个声音与歌手唱的音调无关。一个歌手可以唱的高，但是他的声道却不会变化。由人的声道通过共振创造出来的这种特殊的声音叫做共振峰 (formant)。在发元音的时候你会特别注意到这种声音。Harmor 中对共振峰漂移的保护是为了当音调改变的时候，依然保证共振峰不变。



旋钮 FORM，控制的是以分为单位的共振峰保护能达到的最大值。

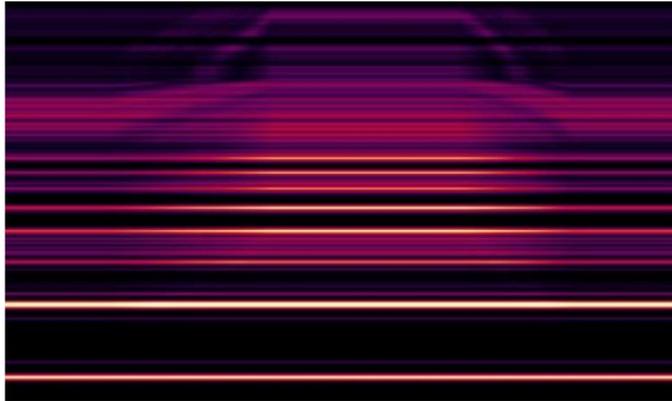
我建议你可以尝试将一个带有共振峰的音频文件加载进来，来把玩一下这个功能，比如说 FLS\_FreeStyle\_05，你可以从 Packs->Legacy 中找到这个文件。将速度 (speed) 设置成 0%，图像频率像素 (Image frequency pixel scale) 比例设置到 100%。将共振峰设成 -600 分，然后混合设成 +100% (我们随后会讨论这一块)。现在当我们播放这段声音的时候，他就会暂停在图像的其中一个位置上。接下来再将共振峰提升到 +600 分，然后再调回 -600 分。你现在就可以看到某些小振荡器的音调是怎样的保持不动，同时“混乱的”共振峰小振荡器被放大和缩小。这个功能的应用之一就是去当我们提高歌手的声调的时候，避免 chipmunk (花栗鼠) 效应，也就是说可以降低共振峰的位置。



在默认设定情况下，我们会根据共振峰旋钮的设置来移动所有的共振峰我们甚至可以通过打开 Editor target 下拉菜单，然后选择 Image formant mutation，来更加精确地设置共振峰突变。



包络线 (envelope) 使得我们可以十分精确地调整共振峰的变化。在下面的图片中，你可以看到使用包络线之后的结果。



## 图像共振峰漂移混合量 (Image formant shift mix)

在 FORM 旋钮旁边你可以找到这个 mix 旋钮。这个旋钮使我们可以将共振峰与原声信号相互混合起来，然后再输出。你可以在 0%到 100%之间调节这个旋钮。

## 图像复制 (Image copy) , 图像粘贴 (Image paste)

当我们复制一张拥有 516 个像素的图像的时候，那么每一个像素都对应一个 Harmonic 中的小振荡器。这张图像的宽度就是这个声音的长度。像素的亮度就是小振荡器的振幅。黑色的像素代表小振荡器没有被激活。图片最底部的像素对应的是排列序数最低的小振荡器。如果图像小于 516 个像素，那么剩下的小振荡器就不会被激活。

图像被按照布朗噪声进行了校准。我们在关于音色的那部分 (Harmonic Timbre) 讨论过布朗滤波。



这两个面板是增益以及频率面板。就我的理解来说，Plane 中的 all 模式不是一个面板，而是一种视图模式。但是用户界面却允许你从 all 模式中复制或 者将图片粘贴进 all 模式。假设你调到了 all 模式，然后复制了这个图像，然后粘贴进了一个图像编辑器。你可以对这个图像进行编辑，比如添加绿色或者蓝色区域，然后现在你再将其粘贴回去，那么按理说属于频率的部分应该进入频率面板，而增益则会进入到增益界面中，对吧？但是实际情况却不是如此……你得到的竟然是一种混合？！

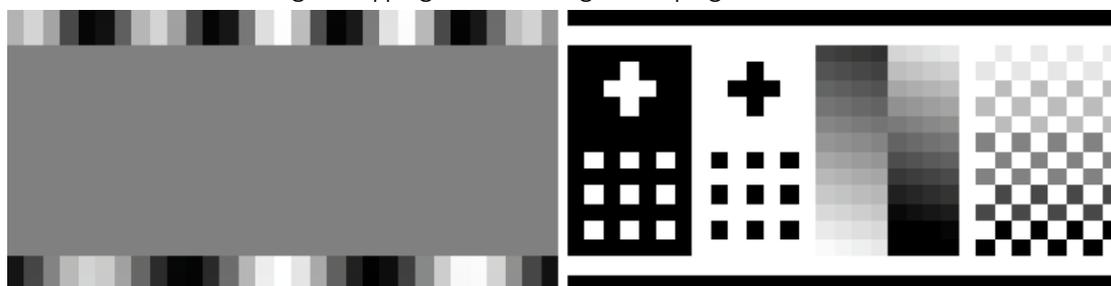
## 图像频率补充曲线以及图像增益补充曲线 (Image frequency interpolation curve and Image gain interpolation curve)

图像中的像素可能会突然地波动，这样的话你从声音引擎中得到的结果就不会是很理想

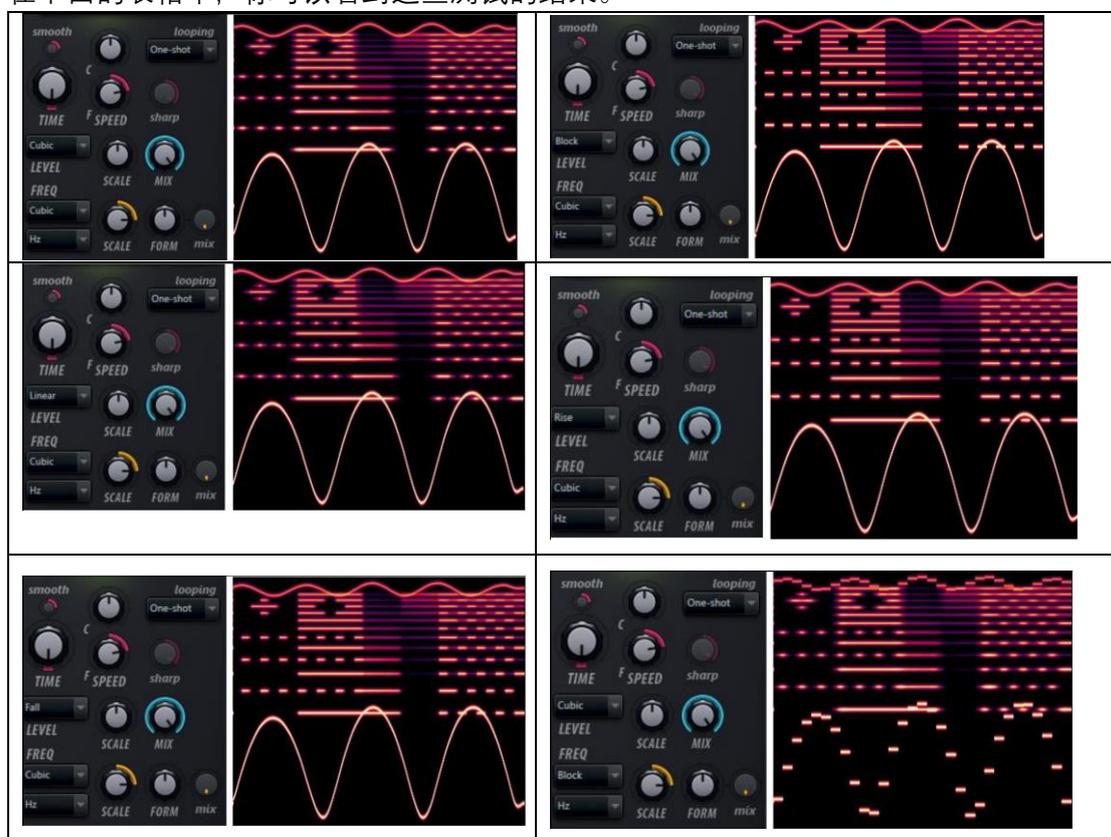
的。为了很好的应对这种现象，Harmor 提供了不同的图像补充算法，来计算不同的均值。在这些下拉菜单中，你可以选择：

- Block
- Cap
- Bell
- Linear
- Cubic
- Rise. 只对逐渐上升的值有用 (rising values)
- Fall. 只对逐渐下降的值有用 (falling values)

我制作了下面这些试验照片。都是 31X16 像素的。你可以从这里下载：<https://www.malmgren.nl/Harmor/ImageGain.png> 以及 这里 <https://www.malmgren.nl/Harmor/ImageFreq.png>。在这里我将这两张图片放大，方便你们看到这两张图。ImageFreq.png 在左，ImageGain.png 在右。



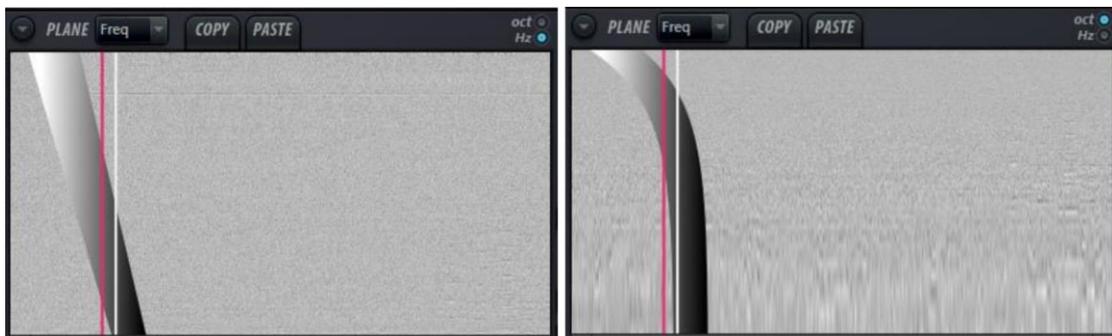
在下面的表格中，你可以看到这些测试的结果。



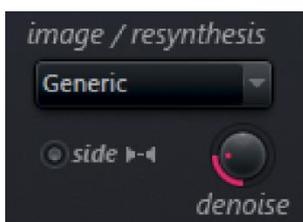


## 显示比例 (Display scale)

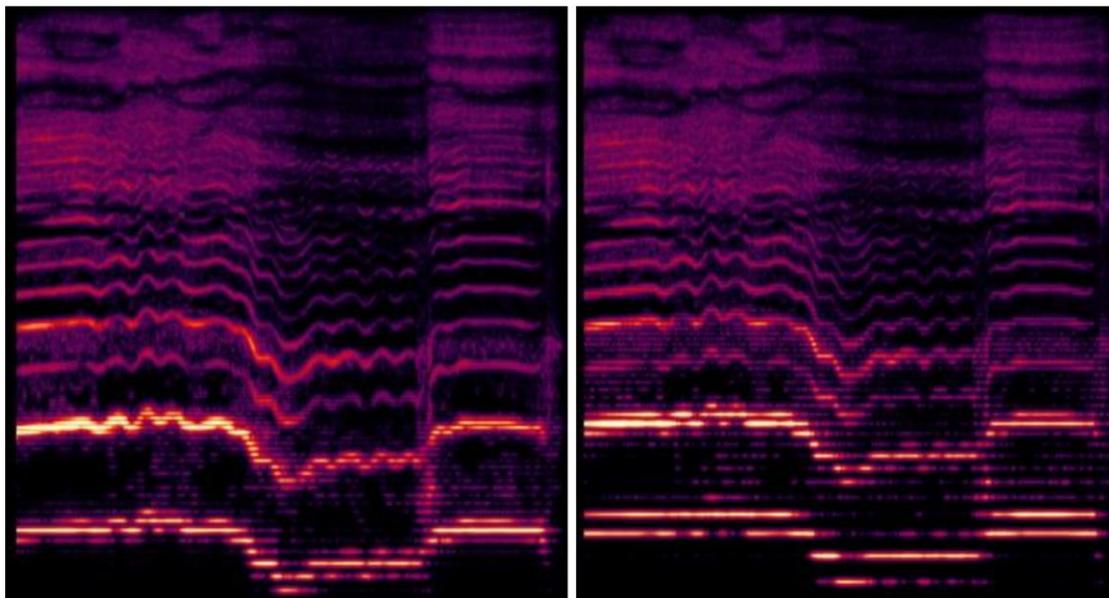
在图像区域的上方你可以找到显示比例开关 oct/Hz



## ADV image / resynthesis 设置: 质量模式 (Quality mode)



在默认设定下, Harmor 基本上可以制作出对原声的一个近似效果。我们也可以切换到高精度模式 (high precision)。有了更高的精度, Harmor 就可以用更少的小振荡器来制作声音, 每个小振荡器的使用也更加有效。更高的精度会占用更多 CPU 的内存。当你在缩放频率的时候可以考虑使用高精度模式。



上面是我在用再合成的模式播放 FLS\_FreeStyle\_05 这段音频的结果。右边的图像是以高精度的模式播放的，右边的则用的是一般模式。高精度模式可以将被激活的小振荡器聚拢的更近，而且小振荡器也会用更少。

## ADV image / resynthesis 设置：去噪声 (Denoise)

去噪声旋钮可以被用来过滤图像中模糊的地方

## ADV image / resynthesis 设置：Side processing / Invert right channel - stereo

在默认中，Harmor 提供单声道的再合成/图片合成功能。如果想激活立体声的话，你得使用 Harmor 的 A, B 两个部分分别负责左声道以及右声道。

1. 激活 A 部分，加载音频数据
2. 激活 B 部分，打开 B 部分。再次加载音频数据
3. 在 ADV 面板中激活开关：side processing/invert right channel.

现在再合成就是立体声模式了。如果你想把 A 部分的设置应用到 B 上，或者把 B 部分的设置应用到 A 上，那么你可以激活部分连接按钮。

即使你可以直接从比如说 Edison 中将音频直接拖入 Harmor, 但是如果你将音频先保存, 然后再将保存好的音频拖拽进 Harmor 的话, 你会更加节省电脑空间, 因为对于后者来说, FLStudio 中只有一个复制品, 而前者则有两个。

## 图像选项菜单：基本知识（Basics）

在再合成模式中，图像选项菜单展示的是工具菜单以及音频文件菜单的前两个音频文件菜单选项。在再合成模式中，图像以及选中的平面选项都会变成灰色。

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

在图像合成模式中，最后的两个部分也会被点亮。

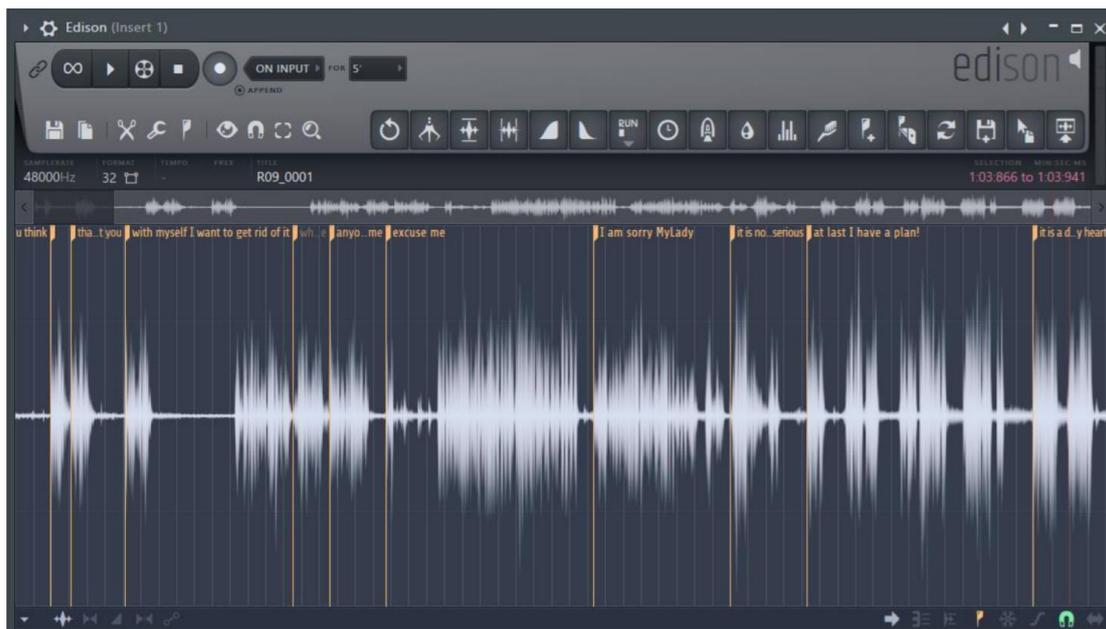
Tools	Audio	Image	Selected plane (gain)
Clear all	Analyze audio file...	Generate random cloud	Open image file...
Prepare time envelope	Convert to image synthesis mode	Degrade horizontally	Save as image file...
Create bouncing loop time envelope	Map audio markers to keys	Degrade vertically	Edit image
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

我们已经讨论了分析音频文件以及转成图片合成模式这两块了。在下面的部分中，我们会讨论所有不那么直观，不那么好理解的部分。

## 图片选项菜单：将地图音频标记/地区映射到键盘

一个音频文件可以被标记，并且我们其实可以通过按下键盘上的某一个键，直接跳到音频的某一个位置。这听起来好像是电影中的一个场景，就是某些好人（或者是坏人）将另一个人的声音录下来，然后在接下来给别人打电话的时候播放这个录音，这就会给接电话的人一个错觉，就是好像是那个人在讲电话一样。好的，让我们来看看吧！

我们首先要在 Edison 中将声音加载进来。在我们需要标记的地方你首先选中声音，然后按下 M 键。一个小窗口就会出现，并且你也可以对该标记进行命名。通过按 Alt+M 你可以对一个区域进行标记。



当你做完这些之后，将文件保存。在 Harmor 中再将这个音频加载进来。当这个音频文件有标记的时候，音频标记的菜单就会被点亮。

Map audio markers to keys  
Map audio regions to keys



现在通过键盘就可以激活音频中的点了。然后呢？你还记得在打电话的时候播放每个单词的诀窍吗？祝你好运。

## 图像选项菜单：准备鼓组的时间包络线（Prepare time envelope for drum loop）

我们上一个部分所做的一切也都可以用于鼓组。音频中的每一个部分都可以用键盘中的一个键激活。为了达到这个目的，有一个分析功能。你可以通过打开 Packs->Loops 然后将一个鼓组拖进图像窗口。选择 prepare time envelope for drum loop。现在你就会的

到一个 image time offset 目标，发生器跟上一部分一样，是 keyboard mapping。

## 图像选项菜单：准备图像时间包络线 (Prepare image time envelope)

这项功能还没有被准备得十分充分。当这个包络线被使用的时候，x 轴就指代图像中的位置。



这个会按照之前那样播放图片。这个也可以被修改。



这里是被修改图像中每个部分的特点：

1. 这有一个原来的斜率，因此也有一个原来的速度
2. 录音的重放会变得更快
3. 录音的重放会变得更慢
4. 正常速度
5. 缓慢地反向播放
6. 快速的正向播放
7. 正常速度

## 图像选项菜单：创建有弹性的循环时间包络线（Create bouncing loop time envelope）

这个功能好像跟上一个功能没什么区别

## 图像选项菜单：在循环区域中慢下来（Slow down within loop region）

这个功能会在图像 fine speed 包络线中设置一个点。如果我们将这个改成一个有坡度的包络线的话，就会根据 fine speed 旋钮降低 fine speed。



## 图像选项菜单：时间偏移的音调（Tune pitch for time offset）

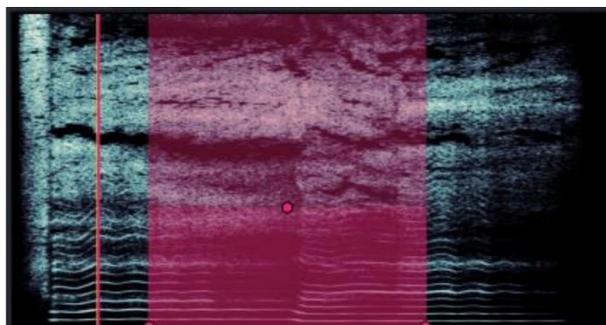
当你选定了这个功能之后，这个功能可以在时间偏移的地方调整音调。

音调包络线（pitch envelope window）窗口打开的时候会在垂直音调位置的左侧添加一个点。音调包络线指的就是主要音调滑杆。关于这个特点我们可以联想一系列的问题。什么时候你才会将音调上下调整仅仅几分？也许是当你将音调向下调整了-600分或者-1200分来制作 bass 音色的时候。假设你想改变你声音的声调，那么你真的想改变主音调吗？



## 图像选项菜单：将看到的区域设置成循环区域（Set viewed zone as loop region）

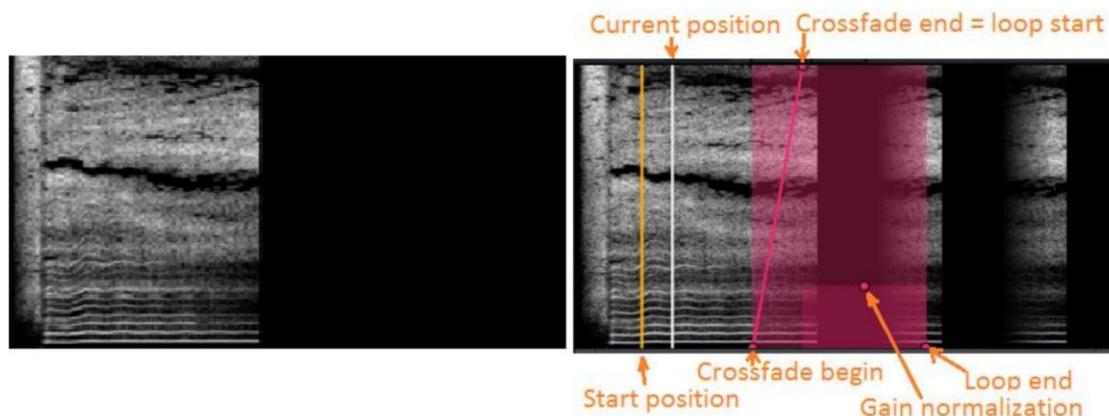
在使用这个功能之前，你可以将你比较感兴趣的地方放大。然后当你选择了这个选项的时候，窗口的一部分会变成红色。然后你就可以再次缩小了。



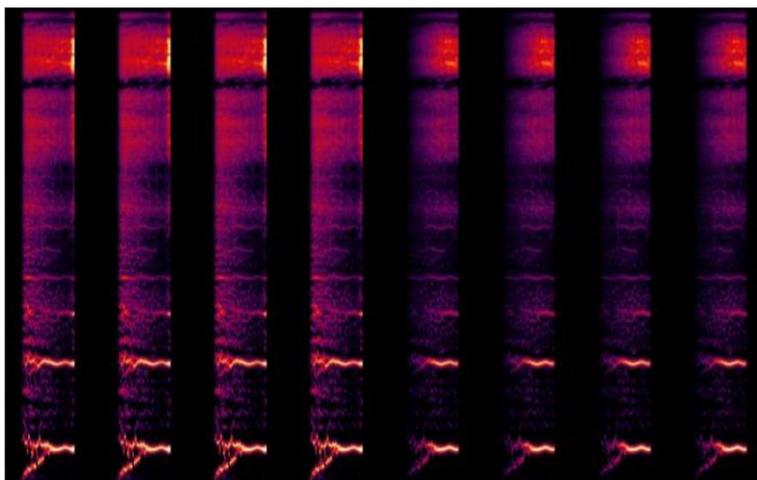
如果循环被调成了 one-shot 模式，那么这个区域将会被忽略。在右侧的图像中，录音的重放将会在红线开始，然后通过红色的区域。这里有三个红点。左边和右边的代表这片区域的开始以及结束。中间的点这是增益正则化控制的把手。增益正则化在 one-shot 模式中不会被激活。你也许会想，如果增益正则化不会产生任何效果的话，为什么会有这样一个选项摆在那呢，我也不知道。

在 ping pong 模式中，重放可以在区域外就开始，但是一旦进入了这个区域，重放就会在内部显现出 ping pong 模式的作用。将起始光标设置在该区域之后，然而当前的位置却是在区域内部时，该区域将不会从起始光标开始——它将会从当前的位置开始。

只有当循环被设置成了 Loop 模式时，增益正则化才会产生效果。为了搞清楚这个功能的作用我分析了一个声音片段，然后再将其中的一部分抹除，就像下面这张图左半部分这样。



在循环模式中，我们也可以控制交叉渐变（crossfade）。循环微端后面的区域与交叉渐变的区域相重叠（在交叉循环区域的起始与末端）至整个区域的一半。你可以将这个交叉渐变区域末端的标记拖拽至右侧来减少循环的开始，以及减少交叉渐变的重叠。



上面的图像展现的是前 4 个循环都在以最大增益被演奏，然后最后 4 个则是最小增益。

右键循环区域的开始或者结尾来将其删除。

## 图像选项菜单：将上一个演奏的区域设置成循环区域 (Set last played zone as loop region)

播放图像的一部分，然后选择这个功能来讲刚刚播放过的区域变成一个循环。对于剩下的功能来说，就与之前讲的一样的了。

## 图像选项菜单：将时间映射到刚刚按下的键 (Map time offset to last hit key)

如果你想使用这个功能，首先按下一个键，然后设置图像时间偏移 (Image time offset)。然后从选项菜单中选择这个功能。你想重复多少次就能重复多少次。

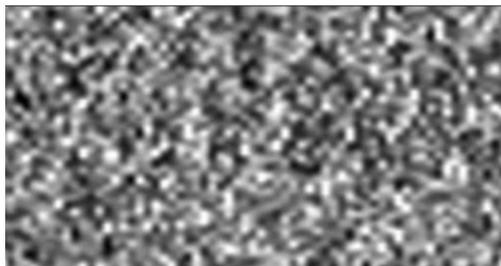
结果就是你会在 Editor target 的 image time offset 中，在 articulator 部分的 Keyboard mapping 中会得到一些点。如果你想重新来一遍的话，你可以将这些点在图中删除，然后再来一次。



另一个选项就是直接在这张图中编辑这些点。

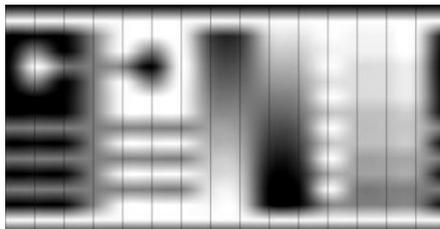
## 图像选项菜单：创建随机云 (Generate random cloud)

会产生随机的像素点。



## 图像选项菜单：横向降解 (Degrade horizontally)

这个选项将会使图像被横向涂抹。下面的例子是我用之前我们使用的图像做出来的。



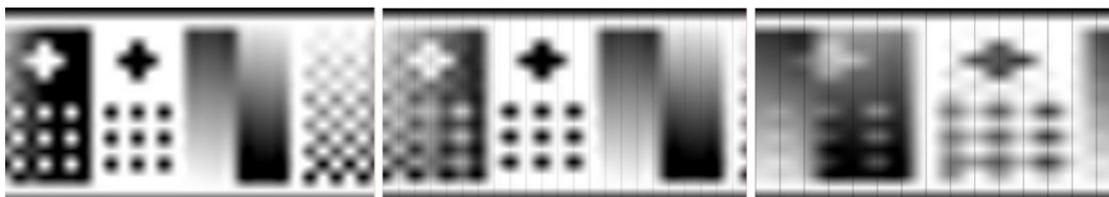
## 图像选项菜单：纵向降解 (Degrade vertically)

这个选项将会使图像被纵向涂抹。



## 图像选项菜单：交叉渐变 (Crossfade) 10%, 25%和 50%

从左到右是 10%, 25%以及 50%。



## 图像选项菜单：剪切顶部（Crop top）

将图像的顶部去除



## 图像选项菜单：反转（Invert）

将图像的颜色信息反转



## 图像选项菜单：垂直反转（Flip vertically）

将图片垂直反转



## 剪切板监视 (Clipboard monitoring)

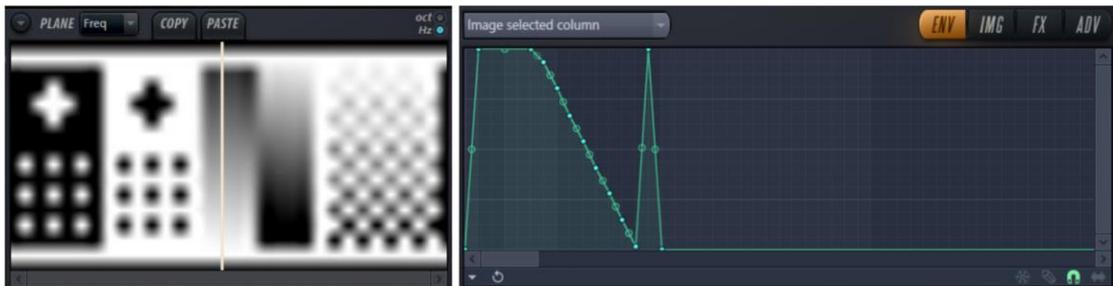
Harmor 可以监视剪切板上的变化。你可以右键复制按钮来开启这个功能。当开启之后，复制按钮的红灯就会亮起。现在你可以比如说开始播放一个声音片段，然后你将图片编辑器中的图像复制，Harmor 就会检测到剪切板上出现了一张新的图片，然后就会自动将其粘贴进 Harmor 的突变窗口，然后在现在被激活的平面上播放。

## 编辑图片 (Edit image)

这个功能会开启一个外部编辑器来编辑图片。默认的图像编辑程序会被打开。对大部分人来说，你可以把这个当成一个画板。

## 图片选择列 (Image selected column)

这个功能可以让我们在 Harmor 内对目前的图片一列进行编辑。首先将图片速度设置成 0，然后点击图片，这时你就会播放图片的其中一列。刚刚播放过的那一列就会被显示在 Image selected column 中。

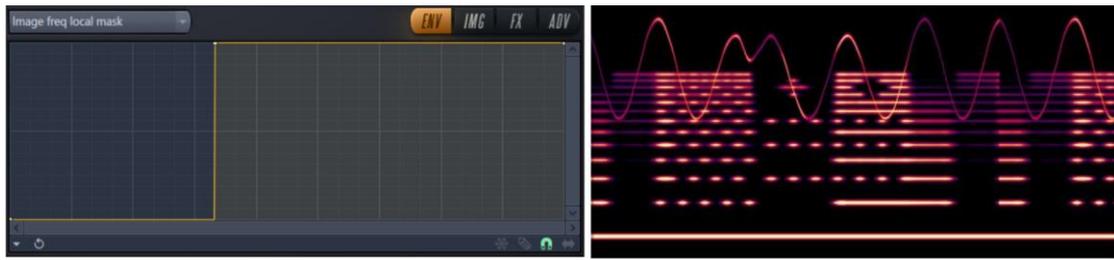


如果你在 Editor target 窗口中改变了图像，那么图像就会显示出你的改变。

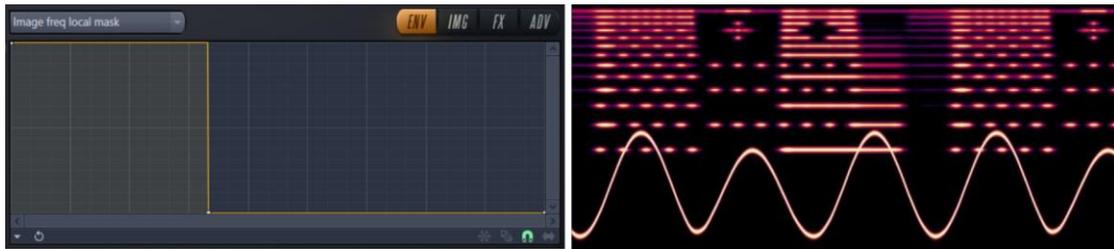
## Image freq local mask

这张 Editor target 图像使我们可以从频率平面中将频率删去。通过我们测试图像的例子，

我们会更容易理解其背后的意义。在我们的第一个例子中，低频被过滤掉了。



下面的图中则是相反的情况，也就是高频被过滤掉了。频率平面在两个例子中都是相同的。



## 第 23 章 . 高级目标 (Advanced targets)

这本书的第一部分讨论了当我们在不使用包络线，发声器，以及下拉菜单，而仅仅只使用旋钮和滑杆时的情况。在 21 章中，我们讨论了编辑器目标 (Editor target) 的基本知识。由于图像合成/再合成有非常多的目标，因此我将那一章放在了基本知识的后边。现在在这一章我将会讲完所有剩余的目标，并研究他们是怎么工作的，以及我们应该怎样使用他们。

### 过滤包络线量 (Filter envelope amount)

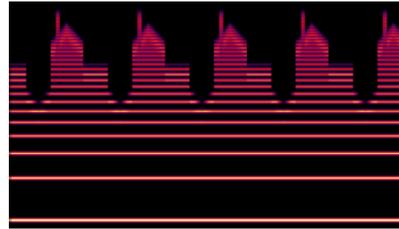


在过滤器部分中，我们有两个 ENV 旋钮，这两个旋钮在默认情况下是没有任何作用的。如果我们想使用这些旋钮的话，我们同样也需要对滤波器频率的包络线进行设置。

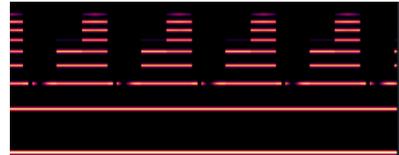


右键 FREQ 然后选择 Edit articulator。这样 Filter 1 的频率目标以及 Envelope 的部分将会显示出来。我决定在滤波器频率包络线中画个房子。在左下角中打开包络线。这里我将开始设置成 loop start，将终点设置成 sustain/loop end。然后我也加入了一点共振。

然后将包络线旋钮拧到-30%的位置。现在房子就上下颠倒了。



现在可能会有点令你感觉有点乱，因为当你调整那些 ENV 旋钮的时候，他们在中间显示的都是 0%，最左侧是-100%，最右侧是+100%。正常来讲，在 Harmor 中这应该属于双极乘积的模式，或者双极加和的模式，但是在这里却是单极乘积。这意味着你可以用 Filter envelope amount 来编辑一半 ENV 旋钮。要么你设定 0%，要么一个正值，要么一个负值，但是你不能在一张图中同时有两个。我觉得这是个 bug。



如果你想在实际应用中试试的话，将 ENV 设置成+100%，然后右键 ENV 旋钮并选择编辑 articulator。按照下图进入一张图像。现在当演奏 C5 或者更高的音的时候，建筑物就会出现在视觉反馈中。当演奏低于 C5 的音是，就不会有建筑物，仅仅是“一片平地”。这是因为最低值是 0%而不是向我们想当然的那样是-100%。



如果将 ENV 旋钮拧向-100%，现在再演奏高于 C5 的音时，我们会得到相反的图像。当我们演奏低于 C5 的音时，我们会得到“一片平地”。

趁着我们在讨论 filter 的这部分时，让我们来讨论讨论 Adaptive filter 1 envelope 模式。根据 Harmor 的官方手册来看，滤波器带宽的变化是音符的一个函数，这样的话施加在低音或者高音上的包络线时长就会比较相似。

## 全局包络线 ADSR 设定 (Global envelope ADSR settings)

我们首先需要通过拧动左下角的旋钮来开启音量包络线开关。现在假设我们想让冲击根据不同的因素变得更快，比如根据我们正在演奏的音



如果我们能做到这一点我们就要打开 articulator Global envelop attack time (译者注：在 Editor target 菜单中间那一列)。较高的音会有更快的冲击时间，较低的音会有更慢的冲击时间。

Harmor 中也有一个包含了冲击比例，释放时间比例以及衰减时间比例的包络线。持续

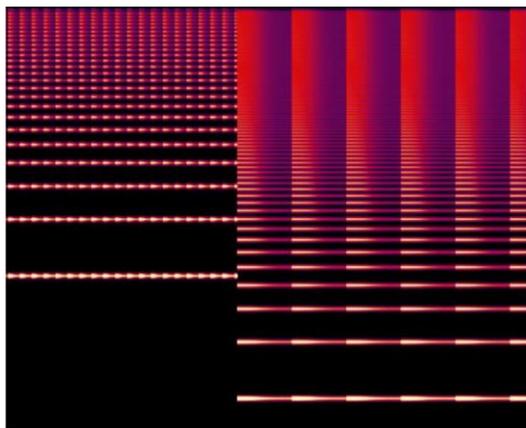
偏差也可以用一个包络线来控制。

## 全局包络线 LFO 设定 (Global envelope LFO settings)

我们首先需要开启一些 LFO 音量包络线。然后再打开全局包络线 (global envelope)。如果你不知道怎么开启 LFO 的话, 先将 Editor target 换成 volume, 然后将 articulator part 换成 LFO 部分。当你满意了之后, 你再换回将 Editor target 换回 global LFO speed。



当我们演奏一个低音的时候, LFO 速度会很慢, 而演奏一个高音的时候 LFO 速度则会很快。



## 全局 Global EQ

均等化是指频率被放大或者变得更加安静。在 Harmor 中有两个 EQ 系统:

- 全局 Global EQ: 不管正在被演奏的是哪个音都会被均等化
  - 局部 Local EQ: 均等化并且跟随正在被演奏的音, 详见下一部分
- 这两个部分都在当音频数据被整合成一个声信号之前作用在小振荡器上。

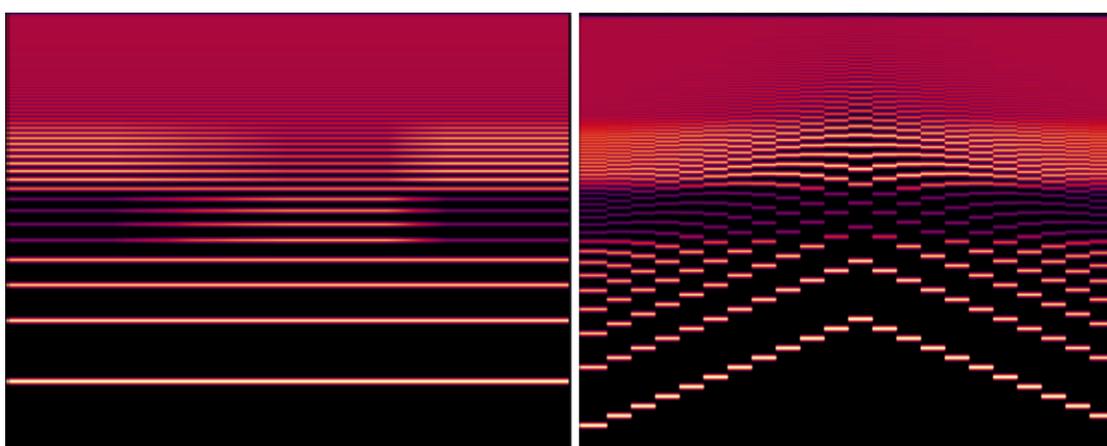


全局 EQ 在用户交互界面中有一个旋钮。当你右键这个旋钮时, Editor target 中的局 GlobalQ envelope 界面就会被打开。这与直接选择 Editor target global EQ 相同。其实除非这张图像被更换过, 这个旋钮不会有任何作用。

如果我们制作出了跟下面这张图展示的一样的图像, 那么从 C6 到 D7 之间的频率就会变得更加安静。从 D7 一直到 E8 之间所有的频率都会被放大。



全局 EQ 在音色上不会有任何影响。当我们用这个音色来演奏一个音，然后将 EQ 旋钮降低到 0%，然后再将其增高到 100%之后，结果就跟下面左侧这张图一样。



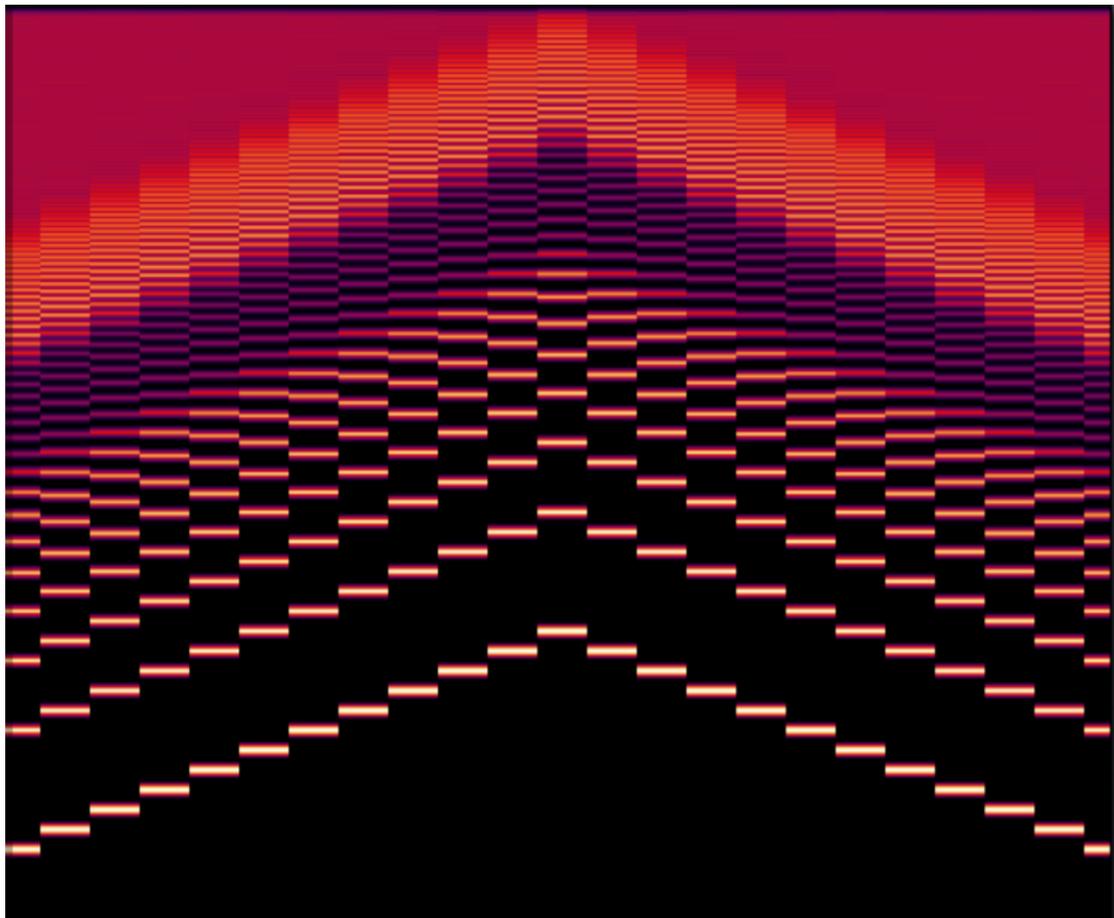
当我们演奏音符的时候，全局 EQ 则保持不变，就像上面右边这张图一样。

## 局部 Local EQ

Harmor 中的局部 EQ 是会跟着正在被演奏的音符改变的。如果你想修改这个包络线，在 Editor target 的下拉菜单中选择 Local EQ。假设我们做出了下面的这张图：



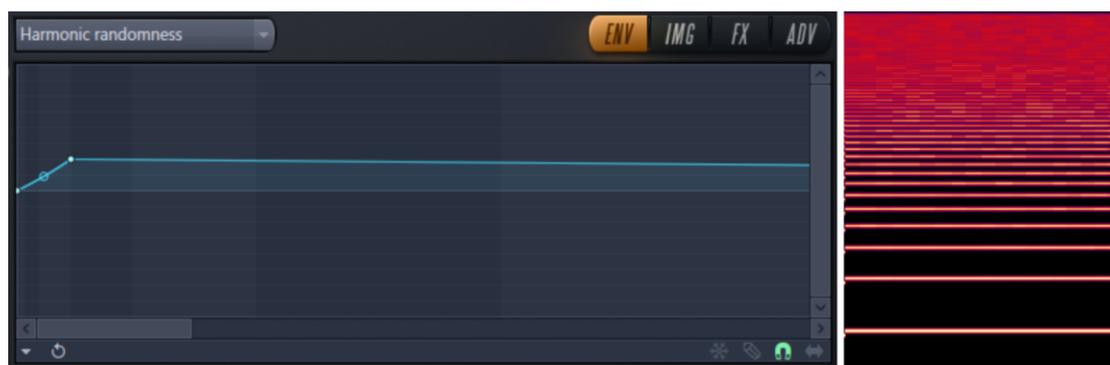
这张图同样也会对音色有一定的影响，因为一些小振荡器中有些被放大了，有些则变得更加安静了。在这种情况下，音色变得更“尖”了，但是区别并不是很大。



## 和声随机 (Harmonic randomness)

假设你想在波形中假如一点随机性。那么打开 Editor target 中的 harmonic randomness

然后对包络线进行编辑。



每当当一个音符被演奏的时候，小振荡器都会被稍稍的被修改。这种修改不是一直都在不停地修改的，而是一旦当这个音符被播放之后就保持不变。下一个音符又会有其他随机的变化。

## 和声剪切 (Harmonic clipping)

在和声剪切那一部分，和声剪切讲的非常细。但是在那一章我们没有讲和声剪切 Editor target 窗口，所以现在就来讲一讲这个。和声剪切与滤波器十分类似，这个滤波器的默认设置是一条平线。在这个窗口中我们可以将这个剪切包络线变得更加复杂。没有什么事是不可能的。



## 和声棱镜 (Harmonic prism)

在第8章棱镜 (Prism) 中，我们讨论了关于和声棱镜的所有内容，除了如何改变默认曲线，我们就现在来看一看吧！下面你看到的是默认的棱镜包络线，以及当我们在改变棱镜量旋钮的时候的结果。

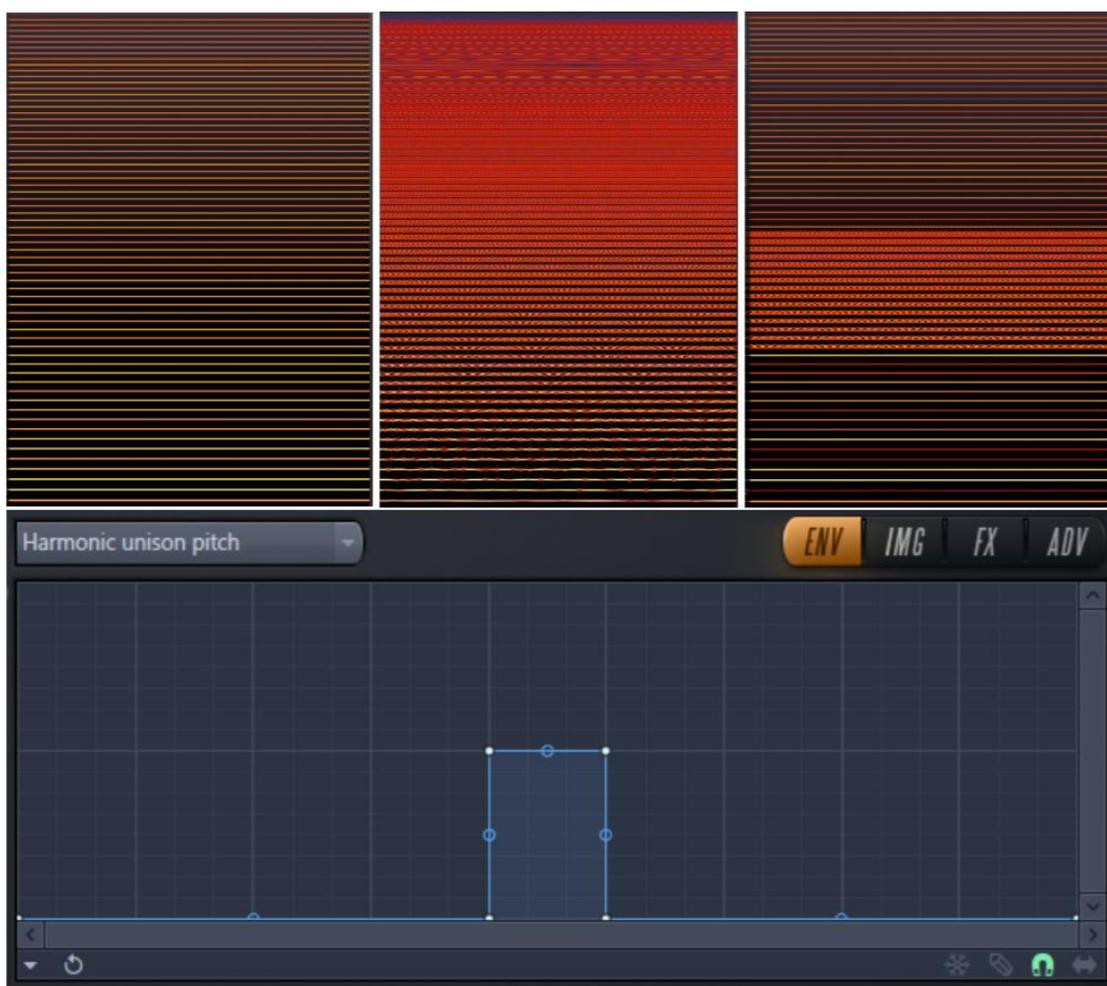


当我们修改了棱镜图之后，它就会变得不一样。



## 和声合唱音调 (Harmonic unison pitch)

在一般的 Unison 中，其效果可以作用于整张频谱。我们可以改变这一点。在左下方你可以看到一个没有 Unison 效果的锯齿波。在中间的是添加了 4 阶 Unison 以及添加了均匀的音调之后的图像；最右侧的是当 Unison 只应用于中间部分时的图像。



在上方的图中，Unison 应用在了频谱中的一小部分上。

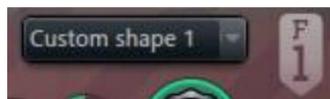
## 图像共振峰突变 (Image formant mutation)

在 Image formant shift 关于共振峰突变的部分中，共振峰漂移旋钮将所有的共振峰在频谱中根据共振峰旋钮的参数向上或者向下移动。通过使用图像共振峰突变的包络线，我们可以将某一特定频率的共振峰向上移动，同时让处于其他频率的共振峰向下移动。



我发现“Laurie Webb Ooh C”这个音频文件中有许多共振峰。当图像被修改之后，共振峰旋钮也会将所有的共振峰进行移动，但却是根据这个图像做出的移动。

## 滤波器形状 (Filter shape)



滤波器形状的下拉菜单有两个自定义形状，一或者二。当我们选择了一个自定义形状的时候，Editor target 窗口将会展示这个形状的曲线。



在这里我制作了两个自定义滤波器的形状，并且我也调整了滤波器宽度的旋钮

## 共振形状 (Resonance shape)

我们也可以定义两个自定义共振形状。



在上面的图中，当被演奏的时候，波峰在视觉反馈窗口中被点亮了。

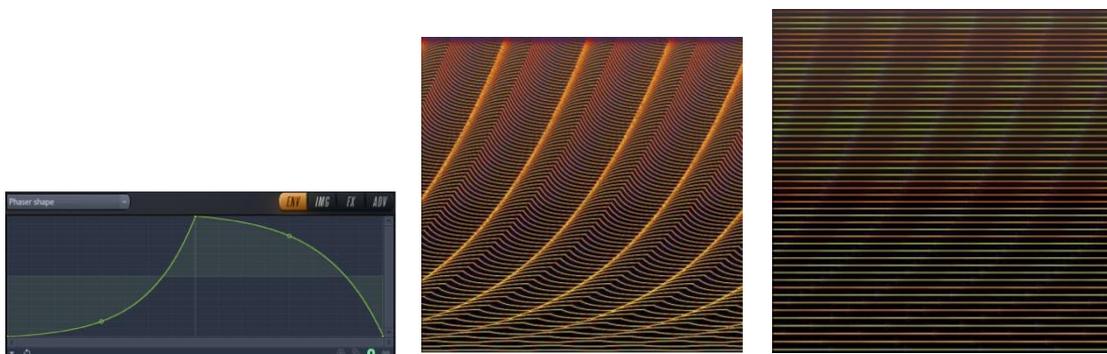
## 弹拨音形状 (Pluck shape)

默认的弹拨音形状是一条从最高小振荡器频率，以及最短时间到最低振荡器频率以及最长的时间的一条线。这也是弦类乐器上的线被弹拨的时候发出的音的理想情况。我们可以改变其发声的形式。下面你看到的是弹拨音的默认直线。以及在视觉反馈窗口中被演奏时的情况。这张图的下面还有一张图，这张图展示的则是修改过的弹拨音。



## 移相器形状 (Phaser shape)

当你从移相器的下拉菜单中选择了 Freq，移相器形状 Editor target 窗口将会开启

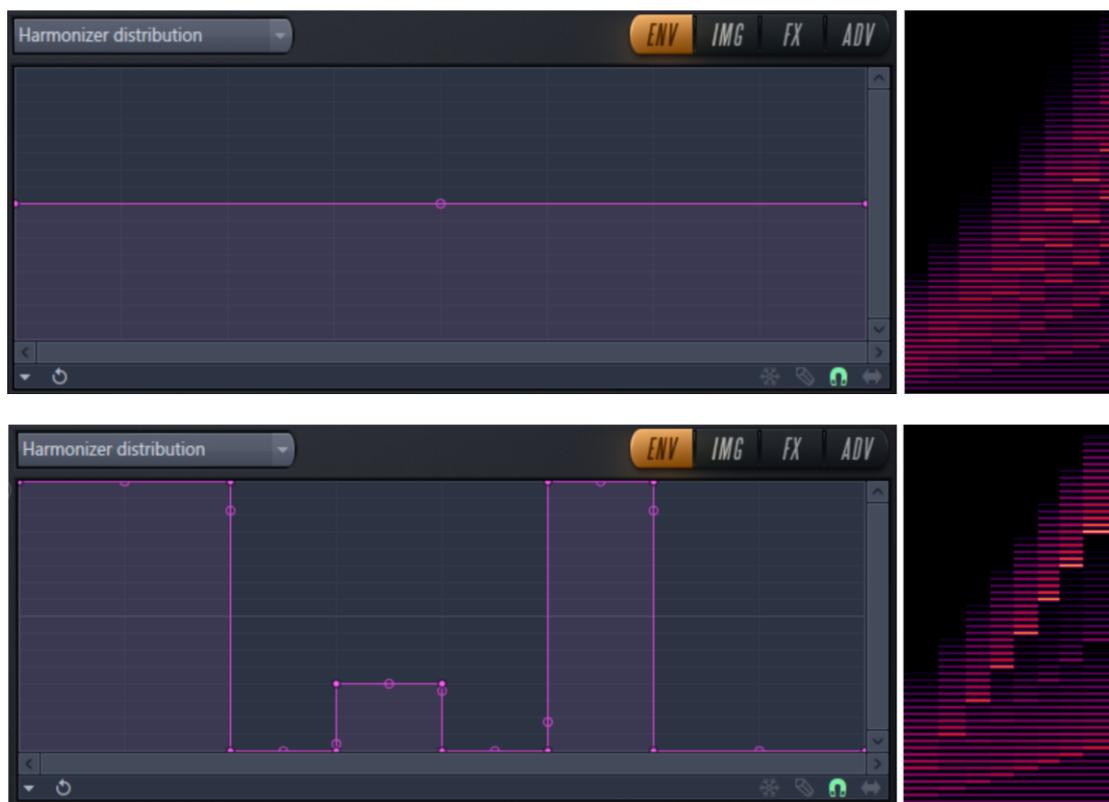


上面最左侧的图像中我制造了一个声音，你可以在中间的 WaveCandy 中看到，但是估计在一般的音乐中不会有太多的作用。Freq 图像是双极的。当选择了自定义移相器形状时，你会得到相同的图像，只不过是单极的，而且会有更适合音乐创作，请看上面最右侧的图像。

## 谐音器分布 (Harmonizer distribution)

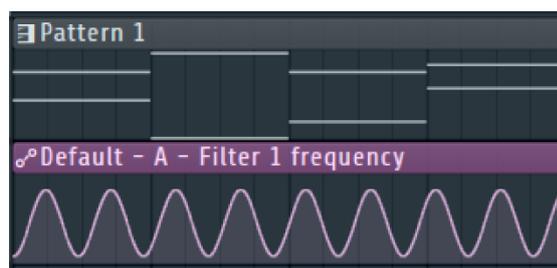
就像我们在第九章谐音器 (Harmonizer) 中讨论的那样，谐音器效应的作用是复制并将谐音根据 shift 以及 gap 的参数进行转换。不同模式的 '+' 以及 'x' 可以使得转换的过程分成不同的步骤。默认的谐音器分布在同一列包含了 9 个 step。这个 Editor target 图像使得增加或者减少每一个 step 的谐音器算法的振幅成为可能。下面图中展示的是默认的

谐音器图像，在右侧的图中你看到的是当 gap 被设置成 0x1，然后 shift 从 0+1 开始渐渐上升到 0+8 时的视觉反馈。这张默认图像的下面我放了一个经过修改的图像，这张图的 shift 也是从 0+1 到 0+8。

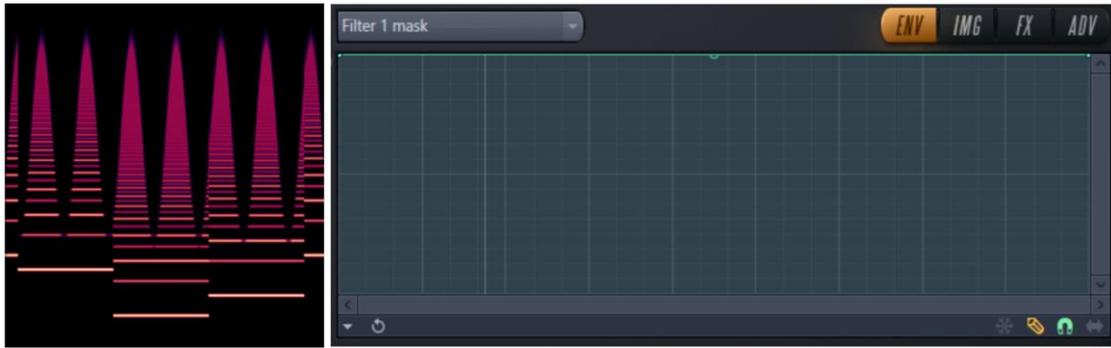


## 滤波器面罩 (Filter mask)

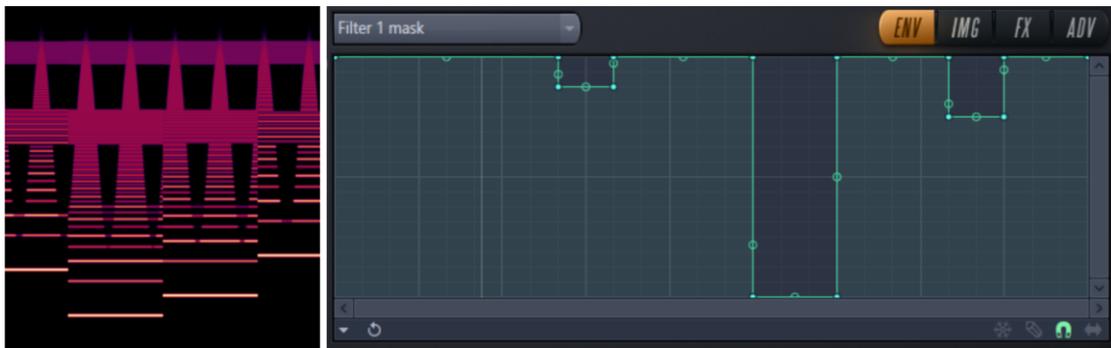
滤波器面罩可以设置一个频率范围，在这个频率范围中原本没有经过过滤的声信号应该呆在输出端。你也可以将其看成一张渐渐将干声以及湿声混起来的图。你可以创建一个简单的样式，比如添加一个锯齿波，来试试这个功能。我创建了 4 个随机的和弦，你能在这里下载这个工程：



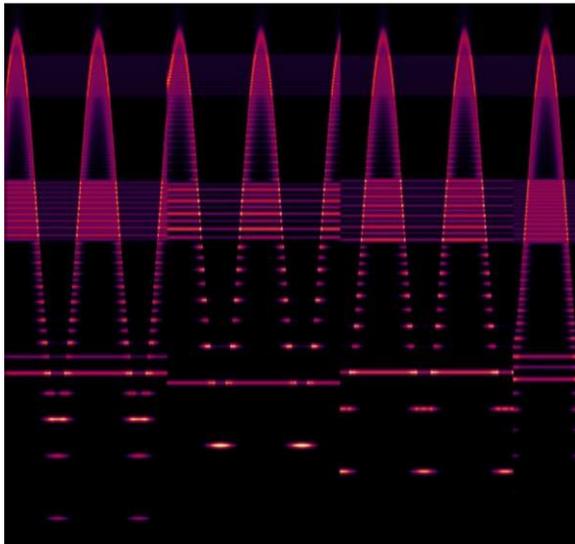
<https://www.malmgren.nl/Harmor/Harmor-51.flp>。我将滤波器频率添加进了一个自动化包络线。有了这个低通滤波器，以及默认滤波器面罩，左下的图就是结果图。在右边你看到的是默认滤波器面罩。



当这条线在顶部的时候，所有的频率都会被经过滤波器的过滤。在下面的图中，滤波器面罩就经过了改变。上半部分是滤波器“湿”的一边，底部是“干”的一边。左边是频率最低的部分，右侧则是频率最高的地方。越是靠近“干”的一边，波形就越不会被过滤。



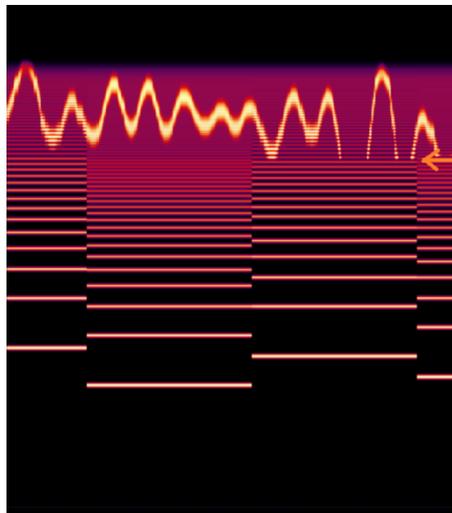
这个功能还有更多的应用场景呢！下面我设置成了经典带通，稍微提高了一点共振。这将会使得面罩处于“地下”的“干”部分消失，但依然将面罩保持在过滤的部分。



## 滤波器共振面罩 (Filter resonance mask)

共振面罩与滤波器面罩的工作原理类似，除了共振也是滤波器效果的一部分这一点以外。展示这一点的一个方法就是设置一个骤变面罩 (abrupt mask)，然后改变共振偏移量 (resonance offset)。

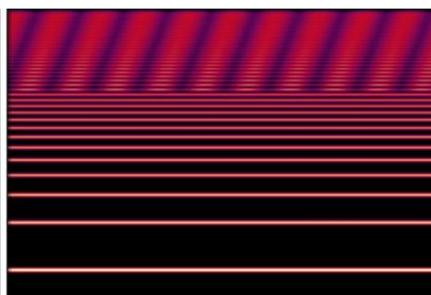
我创建了工程：<https://www.malmgren.nl/Harmor/Harmor52.flp>。在右侧的图像中你能看到我是怎么手动调节偏移量的，我在出现偏移量的地方插入了一个橙色的箭头。



在上面的图像中，你可以看到 Filter 1 resonance mask 的图像，我在截止点那里画了一个箭头。

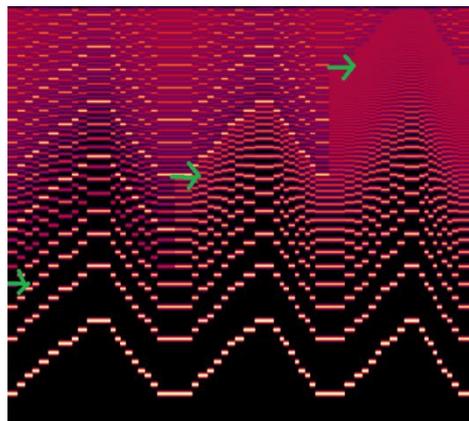
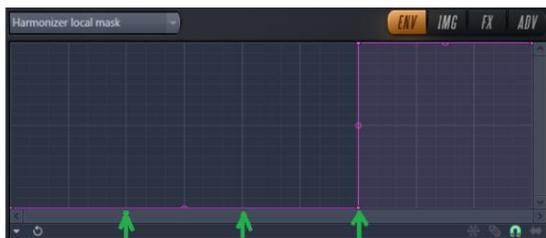
## 移相器面罩 (Phaser mask)

移相器面罩 Editor target 可以关闭部分移相器效果。在下面的左图中你能看到 Editor target 的图像，右侧是结果



## 谐波器局部面罩 (Harmonizer local mask)

Editor target 中的谐波器局部面罩可以用来部分关闭谐波器效果。右侧你看到的是我在键盘上从 C4 一直弹奏到 E5 然后再弹回 C4 的情景。我弹奏了三遍，然后在每一遍中都按照绿色箭头所指的那样改变了谐波器面罩的图像。局部的意思是指面罩本身是相对于音符来讲的。



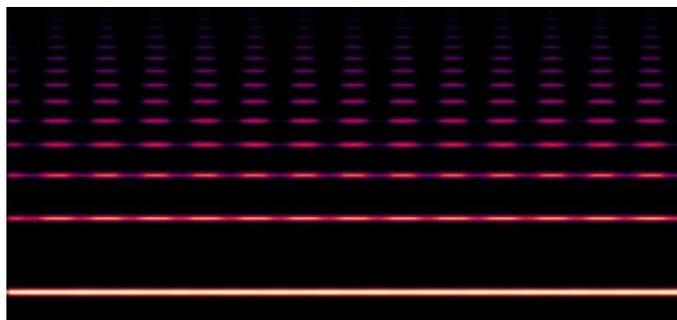
## 模糊局部面罩 (Blur local mask)

为了说明模糊局部面罩的作用，我创建了一个更加复杂的设置。首先我合成了一个正弦波，以保证只包含一个基音。



然后我根据右侧的图像打开了模糊设置。除了这之外，我右键了谐波模糊量旋钮，harm，然后选择了 edit articulator，然后又选择了 LFO articulator 部分。

现在当我们在演奏这些音符的时候，他们就会看起来有些模糊。接下来就是打开模糊局部面罩 Editor target 的时候了，然后再建立一个像左下方图的一个曲线。当再次演奏的时候，上面那几层的模糊效果消失了，在右下角的图中我们可以很清楚的看到这一点。





## 第 24 章 . ADV 标签

Harmor 中的高级标签既有重新定义我们所学的一切的能力，也有点像堆满了杂物的杂物间那样。既牛逼又单调无趣。

请注意标签“Current part”指的是你正处在的 A 或者 B 部分。Global 对 A 和 B 都有影响。



### 预延迟 (Predelay)

在 ADV 标签中你能找到 delay 旋钮。工具提示框中写的是预延迟。这个旋钮在你按下 midi 键盘，和 Harmor 开始播放之间添加一个延迟。旋钮的中间是 0，在左侧的时候延迟是以毫秒为单位，在右侧的时候则是以整个工程的时间为单位。

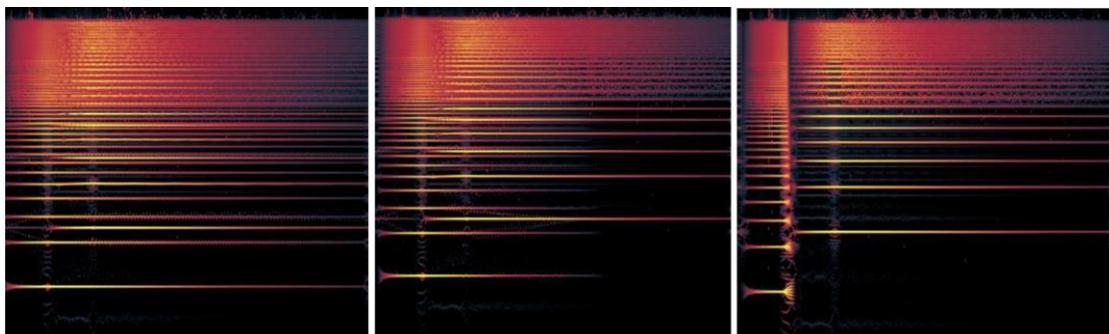
如果你右键这个旋钮，然后选择 edit articulator 时，Editor target 界面就会打开。上半部分是留给当旋钮指向右侧的时候用的，下半部分是给当旋钮指向左侧的时候用的。如果你将旋钮拧到了左侧，那上半部分是没有用的。

### 包络线释放复音比例 (Envelope release polyphonic scale)



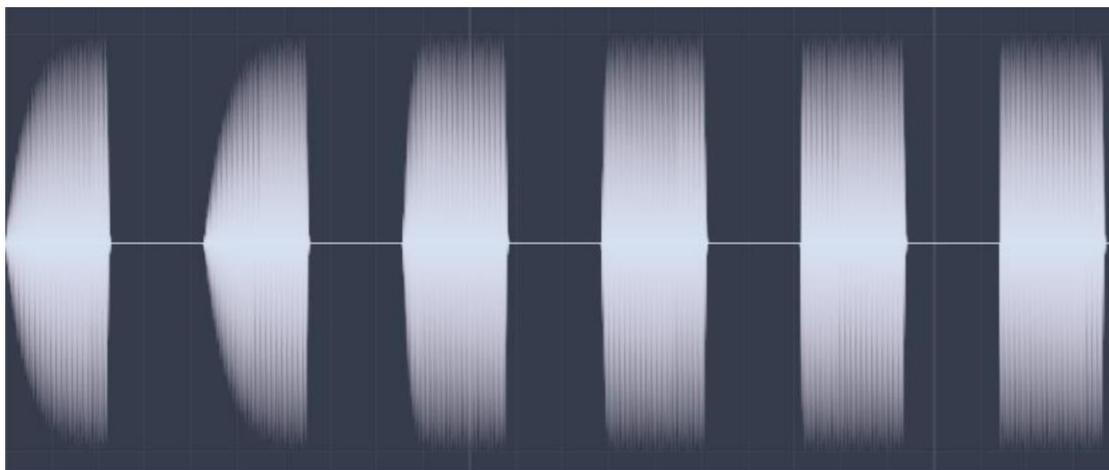
旋钮 poly rel 控制的是包络线的释放时间部分与其他音之间是怎样产生影响的。当被向右拧到 100% 的时候，一个音符的释放可以与其他音同时演奏。当被拧到左侧 0% 的时候，一个音符的释放就会在其他音被弹奏的时候停止。我创建了这个工程：

<https://www.malmgren.nl/Harmor/Harmor-53.flp>。在这个工程中我添加了5个Harmor，然后在每个Harmor中我都添加了一个包含了C5以及E6这两个音的模式。每个Harmor的复音释放比例都不一样。下面左侧的图中是100%，中间的是50%，最右边的是0%。可能看起来不是非常明显，但是你能听到其中的不同。



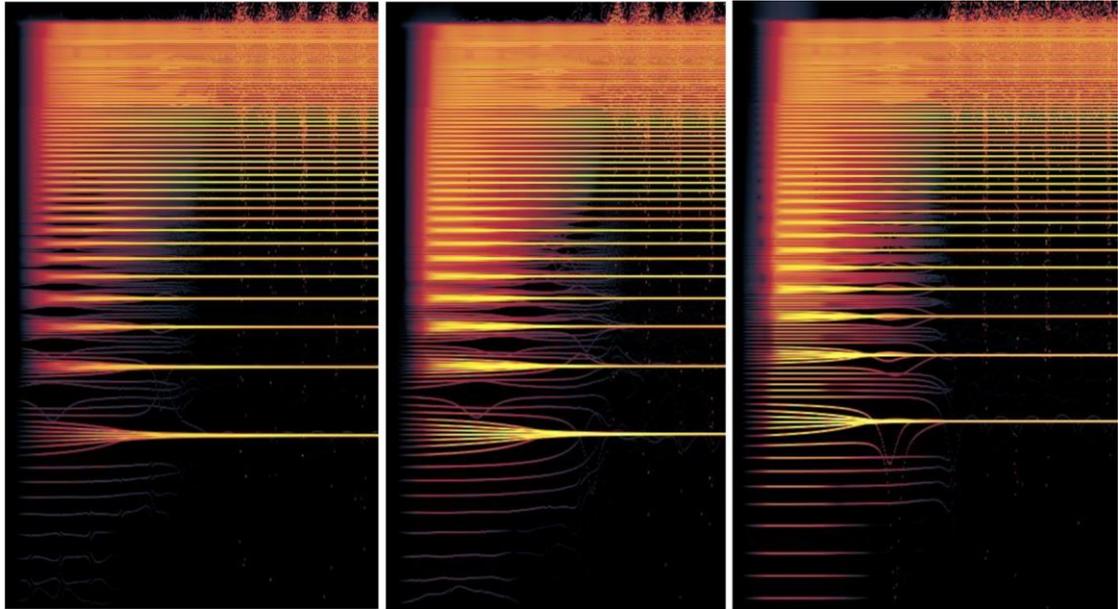
## 斜坡时间 (Ramping time)

Ramp 旋钮与弹拨模式对小振荡器产生的作用差不多。下面我重复的弹奏了一个音。一开始 ramp 的设置是 200ms，也就是将旋钮完全拧到左侧。然后我一边弹奏一边拧动旋钮，一直到 0 为止。你会发现越靠近 0，音的音量就越完整。



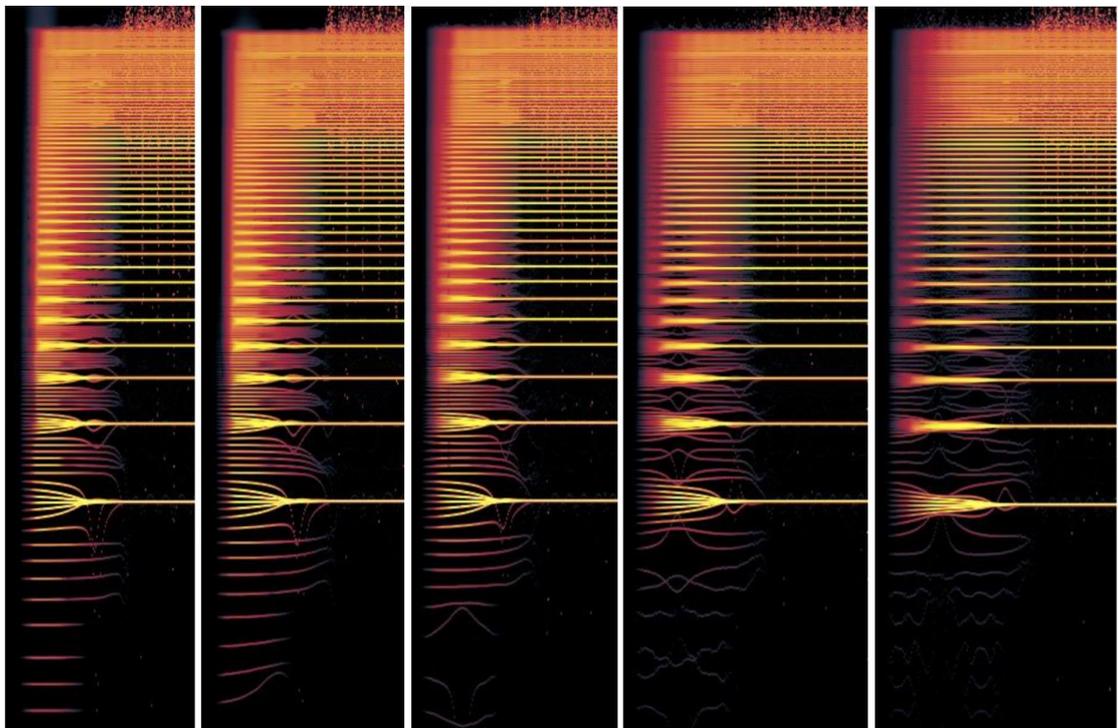
当 Ramp 旋钮拧到右侧的时候，设置的是小振荡器的相对 ramping 时间。这个与旋钮在左侧时不同，但是我会尝试解释这种不同。(译者注：左侧是 absolute ramp，绝对，而右侧则是 partial relative ramp)

下面最左侧的图片是当小振荡器的相对 ramp 被设置成 200 ms 的时候。小振荡器的频率越低，其音量就越小。在中间的图中，ramp 被设置成了 25 ms，这个也是默认值。最右侧的图像中 ramp 的值是 0。被影响的是小振荡器之间的空间。在参数是 200 ms 的时候，小振荡器之间在一开始会有一些洞。在 25 ms 的时候，这些洞就从左下方开始，频率越高这些洞就越有顺序的排列。当 ramp 是 0 的时候，小振荡器就会同时被激活。



## 杂音消除 (Declick)

当从四分之一周期开始演奏一个正弦波的时候，Harmor 需要以最大动力来产生声音，这是很困难的。结果往往是会出现一个点击 (click) 的声音。对于叠加合成来说，我们需要将不同的正弦波叠加起来，考虑到刚刚讲的问题，最好在开始的时候要慢一点。这就是杂音消除的功能。下面是将这种功能视觉反馈时候的情况，其中杂音消除的设置都不同。从左到右，杂音消除旋钮从处于最左边的 0 ms，到了处于最右边的 25 ms: 0 ms, 2 ms, 7 ms, 15 ms 以及 25 ms。



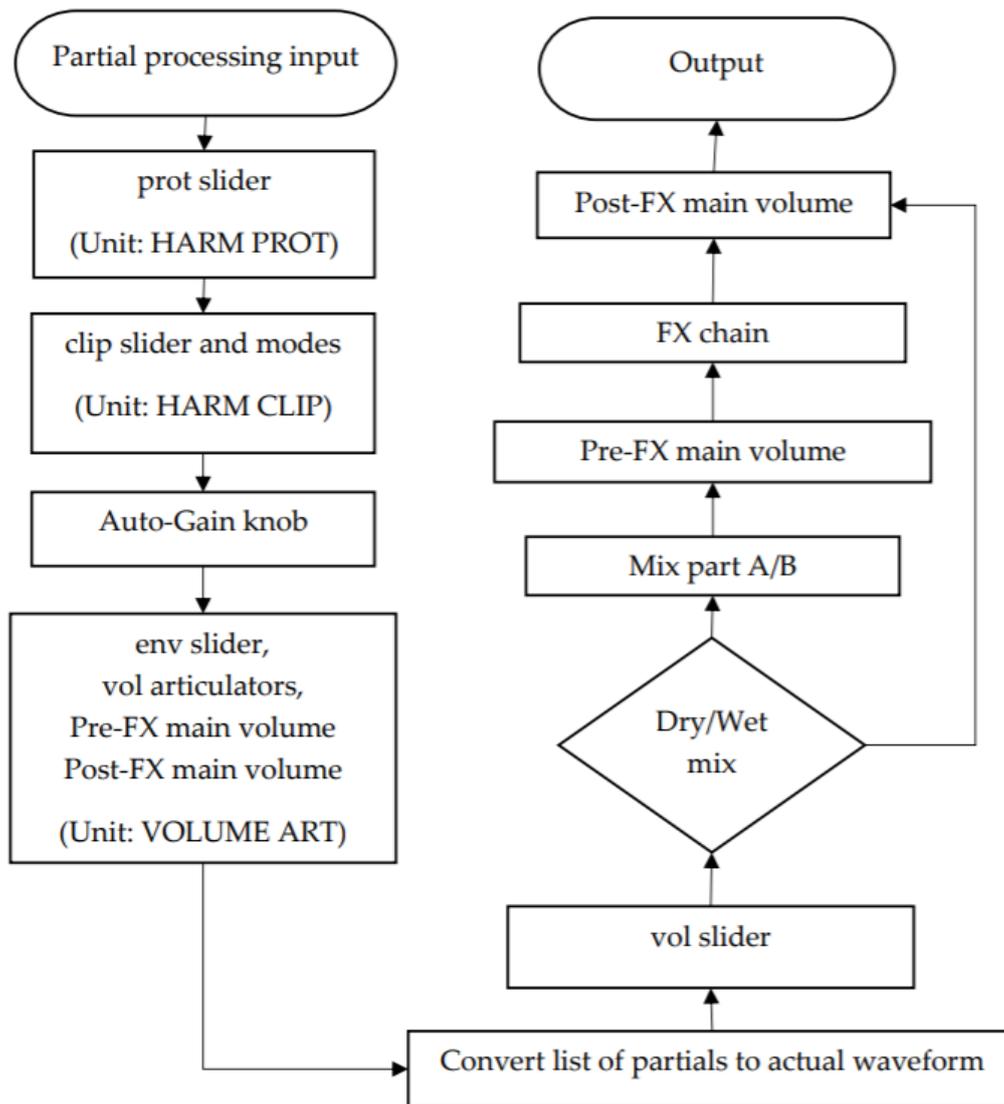
请注意小振荡器之间黑洞的变化，以及越高频率的小振荡器被激活的越来越晚。在上面的图片中我将 ramp 设置为了 0。

## 单位顺序 (Unit Order)

单位顺序部分使我们可以重新定义 Harmor 中的单位。右键列表中的选项让他们向上移动，或左键让他们向下移动。单位顺序作用在一个部分上。

为了你的方便，图片中的选项都被我设置成了可以点击的选项，你可以点击他们来跳转到你想去到的部分。





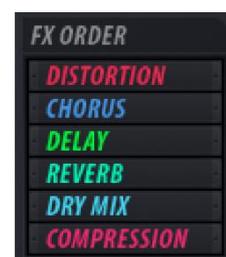
(译者注：实在是非常抱歉，作为一个电音小白，这一点没有看明白作者想要表达的意思，所以我把原文复制粘贴在下面)

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.

## 效果器顺序 (Effects order)

FX 标签的效果也可以被重新排列。其交互界面与单位顺序的工作方式一样。右键以将其顺序提前，左键以降低其顺序。

并且在这里如果你阅读的是这本书的 PDF 格式的话，你可以在列表中点击其中的选项，然后跳转到你想去的部分。



效果器顺序同时作用在 A, B 两个部分。

## 自动增益模式 (Auto-gain mode)

这是一个包含了两个选项的下拉菜单：

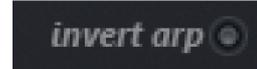
- 相对 (relative)
- 绝对 (absolute)

之前我们在第四章自动增益那一章讲过增益。我还没有搞明白自动增益模式这个菜单中的选项是干什么的。我觉得这个模式在单位顺序中移动了自动增益模组，但是我还没有搞明白具体是怎么做到的。

我未来会和 Image-Line 公司讨论关于 auto-gain 的问题的。当我知道了之后会回来重新书写这一部分，到那时候我会与你分享，但是现在你不用担心，因为自动增益这个效果器不是非常重要，因为在真实场景中不论怎样 Harmor 之后都会有一个压缩器，这个压缩器将会把你添加的增益进行压缩。

## 倒转琶音 (Invert arp)

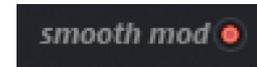
在第 20 章的 arpeggiator points 中我们讨论了 arp。这个 ADV 标签中的开关使我们可以将 arpeggiator points 倒转过来，将上面变成下面，下面变成上面，等等。



## Smooth mod

在第 21 章的 Articulator part: Modulation X, Y and Z mapping 中我们讨论了如何将 X, Y 以及 Z MOD 控制器连接到不同的发生器上。

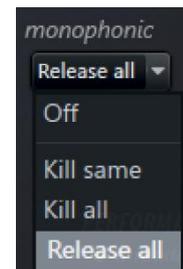
MOD 控制器有一个默认开启的内置平滑化功能。假设你通过 MOD 控制来去制作一个“打击的”自动化，那么你也也许想在这里关掉平滑化



## Monophonic

这个下拉菜单控制的是 midi 音符之间如何互相作用的，特别是音符是怎样被停止播放的。下面就是这个选项：

- Kill same。当相同的音符被再次演奏的时候，上一次被演奏的音符将会先被暂停。



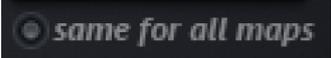
- Kill all。对于任何的音符，在其被弹奏之前，所有的音必须先暂停下来。
- Release all。新的音符将会强制所有正在被演奏的音符进入他们包络线的释放阶段。请查看第 21 章的 Articulator part: Envelope 来确认如何编辑音量包络线。
- Off 这是 monophonic 的默认值，任何正在被演奏的音都不会被阻断。

我理解为什么用 Kill（杀死）这个词，但是难道我们不能找一个相对来讲没那么暴力的词吗？

## 随机 (Randomness)

随机总体来讲是通过给出一个初始值，叫做“种子”，然后随机算法就会根据这个种子给出一系列随机值。如果使用另一个种子，就会得到完全不同的另一系列随机值。如果种子相同，那么你得到的随机值就是相同的。我们可以从电脑的时钟当中的到不错的种子值。随机的下拉选项设置的是什么时候获得随机值的种子。我们可以从下面的选项中选择一个：

- Per voice，每个音都有一个新的种子值
- Per part voice，每个部分都有其本身的随机种子值
- Per unison voice，每个 unison 都有其本身的随机值
- Off，Harmor 仅仅只提取一个种子，然后用他创建出所有的随机值

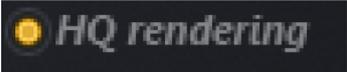
 在下拉菜单的下面，你能找到开关 same for all maps。这个选项对所有的映射只使用了一个种子。

现在如果你阅读了官方文档中的这个部分，你读到的会是完完全全另一个故事。我在这里写的是我有根据的猜测，我得跟 Image-Line 公司再确认一下。

## 表现 (Performance) VS 质量 (Quality)

Image-Line 公司创建的所有插件中都有一个声音设计师模式 (sound designer mode)，以及一个渲染模式 (render mode)。当在用户界面对正在创作的音乐进行编辑的时候，我们只需要保证音乐质量过关即可，并不需要非常好。当这音乐本身被渲染的时候，FLStudio 会进入渲染模式，这个时候就会让音乐变成最好的质量。

 Threaded 开关在默认情况下是开着的，而且允许 Harmor 使用多线程工作方式。正常情况下，多线程工作都是开着的，但是如果你的电脑比较卡，那么你可以将这个关闭，也许这会缓解一点。

 HQ rendering 开关可以在你开始渲染的时候开启高品质渲染。这个开关在默认情况下是开着的。如果你想让你制作出来的音乐的音色是最高的，那就让这个开关保持在开的状态。



Partial count 旋钮可以被用来减少小振荡器的数量。默认设置是 516 个小振荡器，也就是当旋钮被拧到最右边的时候。如果你往左拧就会从频率最高的小振荡器开始，减少小振荡器的个数。这样就会减少占用的 CPU 内存，但是相应的，音色就会降低。最少有 12 个小振荡器。



旋钮 safety 控制的是安全天花板。这个旋钮按理说控制的是高于根音的小振荡器有多少个，但是我发现这个旋钮一点也没用。我建议就将这个旋钮保持在默认值就好了。



包络线颗粒度旋钮，grain，控制的是平滑包络线是如何被加工的，但是我并没有看到任何变化。更加精细的颗粒度意味着对声音更加细节的打磨，但同时也会占用更多的 CPU。我同样建议就将这个旋钮保持在默认值就好了。

下拉菜单 Computation precision 包含了 3 个选项：

- 一般 (Average)，对你的 CPU 几乎没有任何影响。这是默认选项，而且对于音色设计来说完全够用。
- 高 (High)，音色当然会更好，但是更加占 CPU
- 完美 (Perfect)，最好的音色，但是最占 CPU

我建议直接去阅读 Image-Line 公司出的 Harmor 手册中这部分的内容，十分值得一读！

## 第 25 章 . 全局菜单



在全局菜单中，我们可以将所有的预设复制到粘贴板上。当然我们也能只复制其中的一部分。在另一个 Harmor 中，你可以将全部预设，或者部分预设复制进去。



通过使用锁定功能，我们可以保证其中一个部分的设定保持不变，以防止被重写。

你可以在全局菜单中设置合唱以及混响效果。这些效果在第 16 章以及第 18 章中讲过了。

菜单项中有一个 Harmor 的吉祥物！她的尾巴尖指着 Harmor 中的字母“o”。如果你移动整个窗口，她会跟着移动。在这个窗口中，你可以看到所有之多 Harmor 的员工名字。在这个菜单中，你也能看到 Image-Line 公司的网站。

# 第 26 章 . 预设

预设指的是提前制作好的设置。FLStudio 中的大部分插件都可以加载预设。只要你购买了 FLStudio, 你就会立即拥有一些预设, 但是你也可以在网上购买预设。根据你的情况, 有很多种方式来接触预设。下面我列举了对待预设的不同看法的阶段:

1. 如果你刚刚接触音乐制作, 那么我觉得你一点不用对预设感到恐惧。正相反, 一定要多用预设, 多从中学习!
2. 过了一段时间, 当你自己可以制作一些歌曲的时候, 你可以根据你正在制作的歌曲的需求对预设进行修改。即使是对预设很小的改动有时候很可能就会产生奇迹。并且当你这样做了之后, 你就会学会很多分解并学习预制的方式。
3. 随着你试验的越来越多, 以及歌曲做的越来越多, 你可以尝试从 0 开始自己制作预设, 为自己所用。这就是创造属于你的自己的声音的基础! 常见的操作是创建一系列预设, 然后当你制作一首歌的时候对预设进行改进, 然后将新版本保存下来。这样的话你的音色以及预设就会随着你制作音乐的过程慢慢进化了。
4. 当你已经好几次从 0 开始制作预设的话, 你也许会考虑将你的预设分享或者卖给别人。这时也许你要考虑考虑你是不是能像你的客户解释清楚预设中的设定, 以及 CPU 的使用使用是否有效率。

在 Windows 系统中, 一般来说 Harmor 的预设文件储存在:

```
\\Image-Line\\FL Studio  
xx\\Data\\Patches\\Plugin  
presets\\Generators\\Harmor
```

如果你在默认位置安装了 FL Studio, 你需要在 Program Files 文件夹中寻找 Image-Line 文件夹。然后 xx 就是主要版本数字。

在这里, 你会找到一个预设文件 Default.fst。这是你在右上角找到的预设菜单文件。

当你制作你自己的预设, 并且将他们保存到相同的位置之后, 他们将会在默认预设的下面显示出来。如果你创建了一个文件, 然后将你的预设放在了那里, 他们就会出现在菜单中的相应部分中。当 FLStudio 的主版本更新的时候, 你将如要将预设移动到新版本中。

我创建了一个测试工程来把玩这些预设。你可以在这里下载: <https://www.malmgren.nl/Harmor/Harmor-54.flp>。这是一个十分简单的旋钮, 虽然没有特别, 但是十分适合用来尝试不同的预设。但是请然我先介绍其中的一些预设。默认预设部分



- ARP: 琶音。这些预设十分独特，因为就像在琶音的相关章节中讲的那样，他们可以在一个和弦的音与音之间跳跃。我建议用 4 个 bar 制作一首简单的歌，在其中同时弹奏 2 到 3 个音，将其长度设置成 1 或者 1.5 个 bar。并且在另一个模式中，添加一个每个 bar 4 个底鼓的鼓组。现在你可以更加轻易地尝试琶音效果了，特别是尝试用自动化包络线来改变 X, Y 以及 Z。



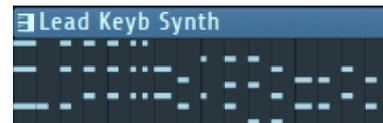
- Bass: 所有在 80 Hz 到 252 Hz 之间的 Bass 预设表现得都不错。在许多情况下，bass 声与和弦同时播放都不是十分和谐。因此如果你想试试 Bass 预设，我建议将一个每个 bar 两个音符，4 个底鼓的旋律。一定要注意不是所有的音高/频率都适合所有的 bass 音色。你也许希望将整个旋律的音调升高或者降低。一个好的建议是不要想当然的认为所有带有 Bass 预设标签的就是一个 Bass 音色。比如“fate oh fate”就不是一个正常的 bass，而是一个 bass 鼓。为什么不把这个放在 percussion 区域中呢？



- Bell: 不同的铃声，你需要尝试才能知道他们的音色是怎样的。

- FX: 声音效果。

- Lead: 这些时 lead 音色的预设，lead 的目的是引领一首歌，正是这个原因，他们在建立起对某一首歌的识别度的过程中更快，允许快速的音符运动，有时会包含渐变的音调或者特点，以及一个更加狭窄的立体音图像或者甚至就根本是单声道。我建议你尽量用同一个音来尝试不同的 Lead 音色。



- Keyboard: 这些预设常常是用来演奏和弦的。在这个组当中，你可以找到听起来像钢琴以及管弦乐器的预设。如果你已经试过底鼓以及 bass 了，那么你可以尝试将再加上 keyboard。

- Synthesizer: Synthesizer 中的预设一定是由于过于特殊，或者难以分类。

- Pad: 这些预设也可以演奏和弦，但是他们主要是被设计来烘托整个音乐氛围，在增强声音的过程中更慢，音调不变，音符的运动更加缓慢，并且立体声图像更加宽阔。我建议制作一个试验歌曲，在这首歌曲中更长的音在每个 bar 结束之前就结束。当尝试这些预设时，可以将 pad 的预设上下调整几个八度，来看看这个预设听起来如何。在钢琴窗中按住 ctrl 键然后按下上或者下箭头来移动。



- Percussion

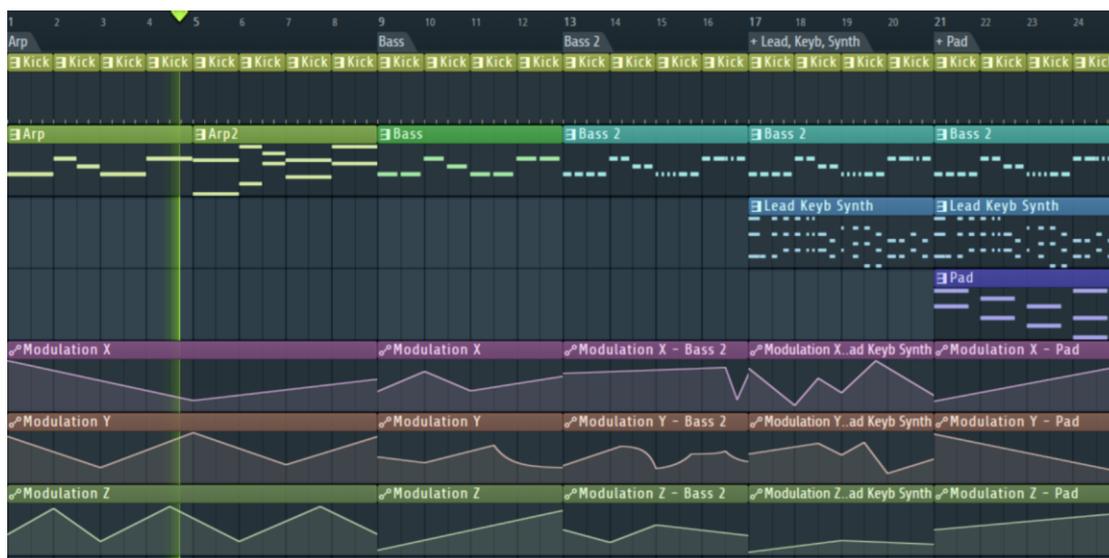
- Sequence: 这部分包含的是精心制作的 sequence programming，几乎一整个歌曲都能被在这里找到

- String

- Template: 这些预设包含了一些模板，你可以将这些模板当成你设计自己的预设的开始。我会在这章的后面谈论这些预设

- Tutorials and tricks。下面你会分别找到这些预设

这是整个测试工程的样子：



你可以从 bar 1 一直选到 bar 8，然后播放这个片段。你可以打开 ARP 预设菜单，然后更换预设，来尝试尝试不同的 ARP 预设听起来如何。

接下来你可以从 bar 9 一直选到 bar 12，然后按下播放键，然后在不同的 Bass 预设之间进行切换。

从这里开始，这个工程从的 bar 13 到 bar 16 一直在用 Kick 以及 Bass2 进行 build up。然后 bar 17 到 20 之间，Lead 音色 Keyb Synth 模式被加入了进来。最终，pad 模式加到了 bar 21 以及 24 之间。

## 如何分解预设。

当你读完了这本书之后，你会很快知道所有而东西是怎么组成的。当我制作完上面的测试工程之后，我突然发现 Cubist 很有趣，所以我将用 Cubist 为例来教大家如何分解一个预设，以更好地学习其中的精髓。

首先在 Harmor 中加载 Cubist。我建议先简单的写一段旋律用来测试，比如在 1 个 bar 中写 4 个音。



当关闭了 B 部分的时候，我们发现当 Z 被设置到 0% 的时候有声音，但到 100% 的时候就每有声音了。当我们关闭了 A，然后打开了 B，然后再次拧动 Z 旋钮，这个时候 0% 没有声音，而在 100% 的时候就有声音了。这与对 Z 的描述完全一致。这是如何做到的呢？

如果想知道 Z 是如何设置的，请打开 Editor target 菜单，然后注意，当菜单中某一项的设置与默认设置不同的时候，就会出现一个三角形。当我们将这些所有调整过的 Editor targets，以及其效果列举出来，那么这个列表就会像是这样的：

- 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

在上面的列表中，你可以看到我将每一部分才 Cubist 预设中的作用都标上了数字：

1. 旋钮 Z。当 Z 被旋转到了最右边，只有 A 部分被激活，并且当慢慢向左扭动的时候，B 部分也被渐渐的激活了。下面展示的是这是怎么做到的。A 部分在左侧，B 部分在右侧。



2. Sequence settings。顺序是被音量，滤波器频率以及音调所设置好的。你可以尝试将这些关闭，然后将旋钮 Z 从 0 拧到 100%，来听一听两部分。A 部分在左侧，B 在右侧。



上面的音量包络线也有 arpeggio 标记。A 部分不断向上。B 部分只有一个向下走的标记，然后剩下的都是向上走。Arpeggio 标记在演奏和弦的时候非常有用，因为 Harmor 可以在音与音之间跳跃

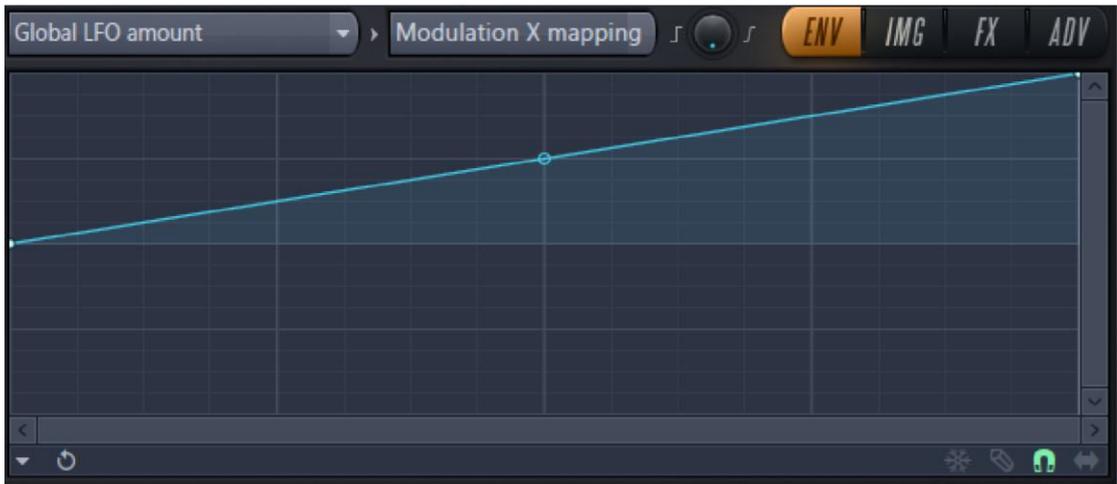


所有的频率包络线都将输出平滑化开启了一点点。上面的 filter 1 frequency，不论是 A 还是 B 都有 30 ms 的 articulation output smoothing。



音调包络线中的 articulator smoothing 被设置成了 15 ms。

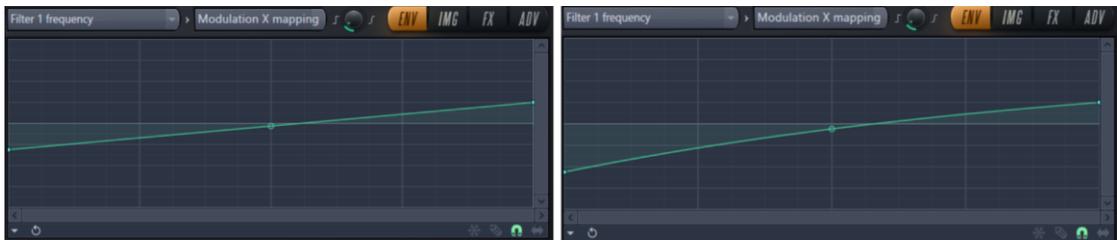
3. 全局 LFO 量。这个是被 Modulation X 所控制的。A 和 B 用的是同一张图，LFO 的使用将在下一点被解释。



- LFO 也被用来控制滤波器频率。A 部分的相位被设置成 002:00:000，B 部分的相位是 004:00:000。A 比 B 要更快。在下面的图片中，左边是 A，右边是 B。



- Filter frequency 的 Modulation X 包络线是修改滤波器频率的最后一部分。A 部分是一条直线，B 部分则是一条曲线。



- Filter 1 resonance 被 Y modulation 旋钮所控制。正常来讲，默认的设置是一个曲线，但是在 Cubist 这里，A 部分和 B 部分就都变成了一条直线。



下面第一张截图上的橙色长方形显示的是默认预设中与 Cubist 预设不同的地方。Cubist A, B 部分的截图下面我将所有 A, B 部分的不同点都总结在了一张表里。

默认设定:



Cubist A:



Cubist B:

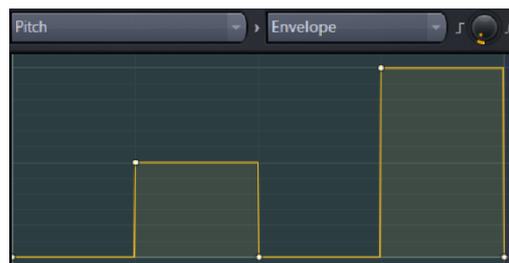


在下面的表格中，我列举了 Cubist 设定中所有旋钮的值。

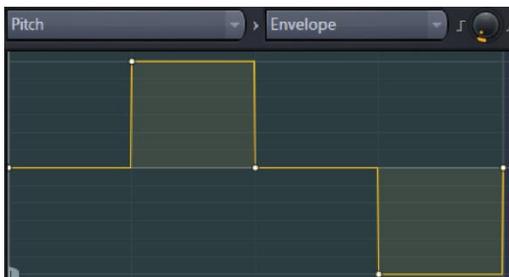
Knob / Dropdown	Default	Part A	Part B
Timbre 1 & 2 mix	Saw waveform	Square waveform	Saw waveform
Phase randomness	0%	100% spectral random	0%
Harmonic protection	-INF dB 0.00	-7.2 dB 0.44	-3.3 dB 0.69
FX dry/mix	0%	23%	0%
Time domain volume	0 dB 1.00	0 dB 1.00	-2.0 dB 0.80
Auto-Gain	0% absolute	100% absolute	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

A 与 B 之间最大的不同就是波形的不同，一个是锯齿波，一个是方波，我想你也已经发现这一点了。

A 部分中的方波被降低了两个八度，也就是说它的频率被设定为  $1.000/4$ 。然后 A 部分还有一个音调包络线，这个音调包络线会向上跳起 1200 分，然后再回来，也就是上下一个八度。然后它再跳起来两个八度。



B 部分的锯齿波被降低了一个八度，因为其频率被设置成了  $1.000/2$ 。也就是频率上下跳跃了一个八度。



相比较于 B 部分，A 部分拥有更多的和声保护。B 部分有一个比较低的时间域音量，使得它比 A 部分的音量更低。因为锯齿波常常具有更多的声音能量。这属于一个平衡两部分的小技巧。

A 部分有一个 25% 的 FX 干/湿混合比。也就是说原声在最后输出的声音中占比更大，但是在 B 部分中则全部都是湿音。接下来我们来看看效果器部分。下面左边的图中你能看到 Cubist 中的 FX 部分，右边则是默认的设置。



没有失真，也没有压缩。合唱的阶数是 4，然后剩下的就和默认设定中的完全一样了。延迟以及混响也一样。虽然被激活了，但是参数设置基本上没有变化。当效果器的效果和 B 部分的效果叠加起来之后的效果是，相较于 A 部分来说，B 部分对回声以及混响的效果影响更大。

ADV 标签没有大的变化，除了 ramp 从 24 ms 变成了 43 ms。

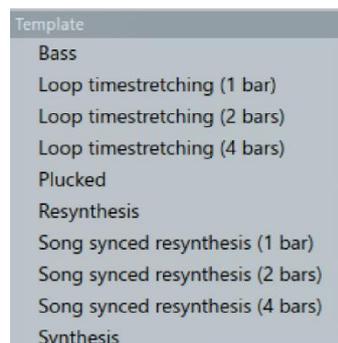
现在你应该对 Cubist 预设有了一个很好的理解了吧！

## 模板，教程，以及窍门预设。

预设菜单的模板部分包含了一些可以作为制作属于你自己的预设的起点。

通过分解这些预设从中学习是十分有利于设计你自己的预设的。

为了帮助你理解，让我们先从 Bass 开始。这个预设是做什

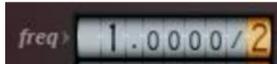


么用的呢？我们可以先看看他的信息：

Preset name - Author

This spares CPU if your preset doesn't have high-frequency content.

这个功能可以设置频率：



全局 EQ 有这么一个包络线：



在教程预设（tutorial preset）部分，你需要对每个预设的信息窗口进行仔细的阅读。这些信息窗口中的一些信息是你在其他任何地方都找不到的。我估计随着 FLStudio 的更新，这个部分会变得越来越完善。

如果能为这些模板，提示以及小技巧预设提供更多信息的话，那一定是很好的，但是我感觉就我所知道的信息还不足以解释的很清楚。也许你可以在 YouTube 上找到更多的信息，或者也可以在网上寻求帮助。在这本书接下来的版本中，我也许会再回来讨论这些预设。

## 创建你自己的预设

如果你用过一些预设，你也许会好奇这些预设是怎么被创造出来的至少我就不止一次想过这个问题。很明显，有一些是我不知道的秘密，如果我知道的话，我会在这里写出来的。

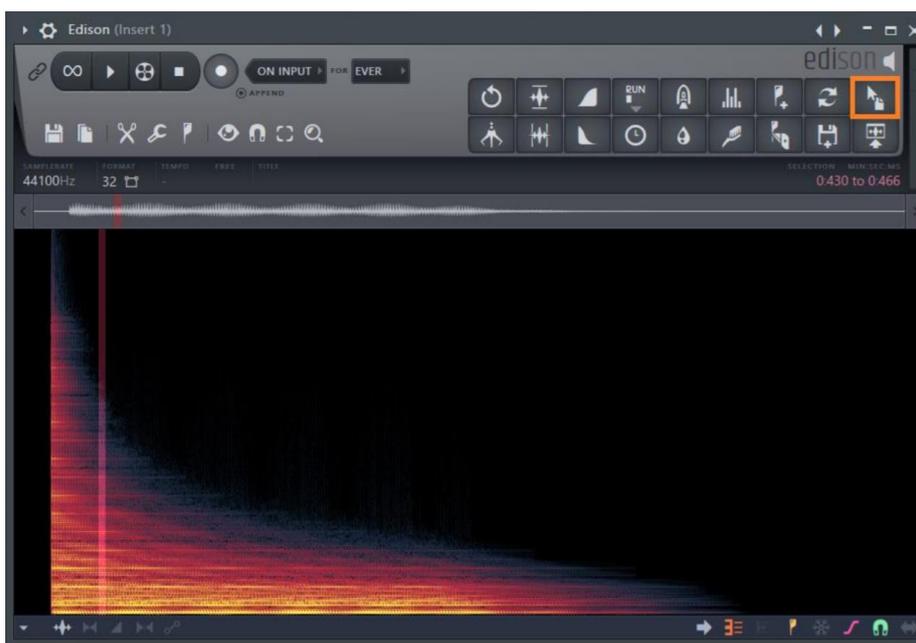
就拿预设 Analog bass 2 中，音色 1 的和声水平来举例子吧。即使我有能力制作出来这么一张图，那我也得用手画个好几天！所以很明显设计这个音色的设计师没有用手画。我相信很多时候音色设计师是先创造一个声音，然后对他进行采样，然后再拖进一个新的 Harmor 中，然后再对这个声音进行编辑，然后再在下一个 Harmor 中插入。所一个音色可能是好几次迭代的结果。所以很多时候，最后的结果都是好运和经验的结合



为了研究这个预设我创造了工程 <https://www.malmgren.nl/Harmor/Harmor-55.flp>。在这个工程的通道架上我插入了两个 Harmor。在 Harmor A 中，我制造了一种音色，然后输送给了 Edison。



这是一个方波，音色中包含的效果有模糊，棱柱，移相器以及一点点亚音保护。最后还有一点 Hill 形式的失真。



然后我录制了 C5 音，然后切出来了一个非常小的片段，然后将其拖 ((with the drag copy / sample / selection tool marked with an orange rectangle) 进了 Harmor B 的音色窗口之中。

结果就是下面这张图，需要手动画上好几天的图！



上面的就是一个例子。如果想将其更加精细化的话，我建议分析 Analog Bass 1，我们会在下一部分这样做。

## 为他人制作预设

Image-Line 公司的手册中强调，如果你想将预设分享给他人，那么你就应该尽量不要让这个预设占用太多的 CPU，而且要尽量的向对方描述你这个预设。在这以部分，我将好好地探索一下这些目标，以及你可以怎么做。

当你自己创造你自己用的预设的时候，如果你创造了一个十分占用 CPU 的预设，那么只有你会深受其害。而当你为他人，比如没有经验的声音设计师，创造预设的时候，如果你将这个预设设计的十分占用 CPU 的话，也许他们就没有办法很好的利用 Harmor。或许你不会挨骂，但是 Image-Line 会，而且我敢肯定他们绝不想挨骂。

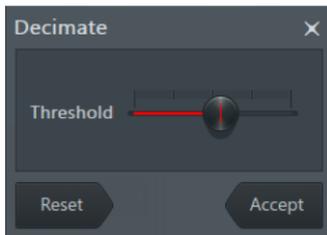
在之前的章节中，我们用了很少的力气就创造出了十分复杂的音色，但是这种复杂真的十分有必要吗？为了好好检视这种想法，我将会用 Analog Bass 1 所谓一个例子，并不是一个批评制作者的例子，而仅仅只是一个例子。其实，Analog Bass 1 正是我购买 Harmor 的原因。我太爱使用这个合成器了。我们的策略就是不断去将各种效果器关闭，然后看看在哪一点音色的质量开始大幅度的下降。首先在 Harmor 中添加 Analog Bass 1。然后我们将会对其进行简化。如果你想的话你可以自己来做，或者直接下载我做好的工程：<https://www.malmgren.nl/Harmor/Harmor-56.flp>。在这个工程中，通到架的第一个 Harmor 是没有被修改过的 Analog Bass 1 预设。下一个 Harmor 叫做 Modified #1，

下面我将对这个 Harmor 进行介绍。

我们首先看一看原版的 Analog Bass 1 预设。这里是原版 Analog Bass 1, B 部分的 timbre 1 窗口。



这里估计得有 1000 个控制点位。左下方选择移除点位选项。



12 个点，现在看起来简单多了。但是听起来差别却不是很大。现在让我们转去 A 部分的 Timbre 1。



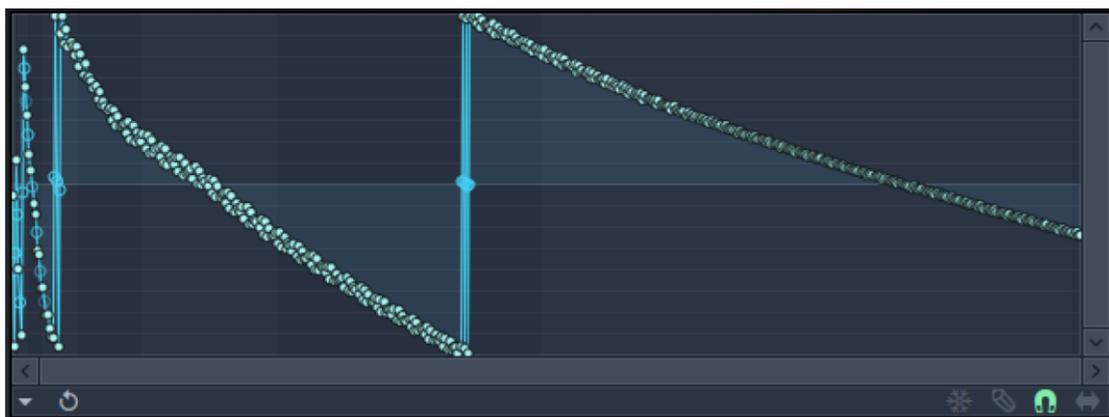
又是这样，估计得有好几千个点位。这次你选择滤波器。



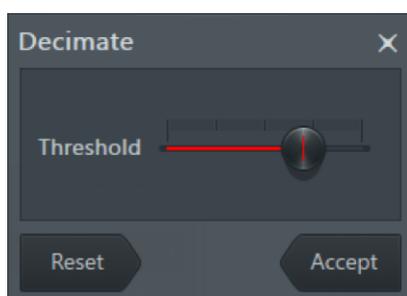
然后你就会得到下面的 Timbre 1 的图：



听起来跟原版非常的像！现在让我们进入到和声相位窗口。在 Editor target 下拉菜单中，选择 Harmonic phase.



现在我们要将这张图中的点位删去。



一开始我试着使用滤波器功能，但是结果是我得到了一种“响铃”的声音，但是那种声音可以用点位删除功能来避免。

这跟原声听起来依然十分的相似！我们竟然在移除了好几千个点位之后，音色与原声依然很像素。这对 CPU 有任何影响吗？不，并没有。我们过会儿再讨论 CPU 的事。所有我们做的这一切都是为了掌握一个音色的主旨，但是银色的主旨到底是什么，在哪儿呢？

如果我们转到 FX 标签，然后关闭了失真，混响，以及压缩。这些效果器是音色的本质吗？失真让声音有了一种特殊的感觉，混响赋予声音以空间感。压缩给音色整体添加了一点点低音，但是总体来讲，效果器本身并不是音色的灵魂，所以让我们再把效果器打开吧。

A 部分以及 B 部分在预设中扮演着不同的角色。B 负责的是高音。如果我们将其关闭，那么我们将我们的注意力放在预设的低音部分。让我们打开 Editor target 菜单来看看这个预设中还用了什么东西吧。接下来我将用“A=B”来代表“A 和 B 都用到，并且都相同”的意思，用“A<>B”来表示“A 和 B 都用到，但是不相同”的意思，用“B”来表示“只有 B 用到了”的意思。

- Filter 1 frequency (A=B)。使用到的部分：Envelope, Velocity mapping, Modulation X
- Pluck amount (A=B)。使用到的部分：Modulation Y
- Local EQ (B)
- Harmonic phase (A<>B)
- Timbre 1 harmonic level (A<>B)
- Filter shape (A=B)
- Global EQ (A<>B)

你能在 Harmor 中的某些地方找到这些重要的元素。比如 A 部分的 Global EQ 长这样：

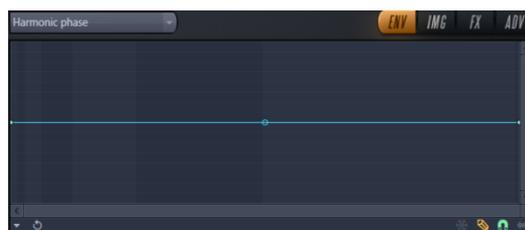


这个图中的小坑是 bass 音色中一个十分重要的组成部分。如果你用鼠标拖拽出那个最低点，然后上下左右的移动，你会发现这个点的位置在决定 Bass 音色上有多么重要，而且通过将其摆放在不同的位置，你会得到不同的 bass 版本。

现在让我们讨论一下 B 部分。当然 B 部分的和声相位是非常重要的。在右侧，你可以看到一个原版的和声相位。在这张图中，你可以看到一个需要人工调节好几天的一个图像。



假设我们将这个和声相位完全删除掉，然后换成一条想下图中展示的那样的一条线。在我们复制之前，你觉得这会对声音产生负面影响吗？为了试验，我关上了 A 部分，然后只播放了原预设中的 B 部分。然后与简化过的版本相对比。在测试工程中，这个 Harmor 的名字叫做“Modified#1”以及“Modified #2”。我发现差别并不大。为什么呢？



这是因为 B 部分的目的是提供弹拨声音的高频部分。B 部分有一个局部 EQ 来剪切基音的低频，以及一个全局 EQ。但这并不是全部



滤波器有一个被自定义过的形状，看起来像一个低通滤波器，然后滤波器频率也被调成了“弹拨音”。尽管其和声相位十分惊人，但是这个音色是一个弹拨音，而且还被一遍又一遍的过滤，所以最后基本上就不剩啥了。在这之外，A/B 混合滑竿给 A 的权重要大于 B 的权重。



你可以不要滤波器频率但是保留弹拨音，你可以不要全局 EQ 但是保留局部 EQ。这些改变并不会给声音的特点带来十分巨大的变动。

B 部分中还有一些东西也值得拿来讲一讲。我们开启 Unison 难道只是为了弹拨音效果吗？如果我们需要一个占用更少 CPU 的 bass，那么我们可以将 Unison 弃之不用。另一个值得思考的问题是 FX 标签中的压缩。在这个预设中，压缩被设置成了“Heating”来稍微加强 bass 音色。在许多的场景中，你会对音乐整体添加一个压缩器。如果你不论如何都要压缩，那何必还要在 Harmor 内部再压缩一遍呢？所以去掉压缩效果可以成为一个节省 CPU 的一个方法。如果使用压缩器的原因仅仅只是对音色的低音进行一点点加强，那么你可以进入 A 部分，然后对低频进行加强，而不是在 Harmor 中添加一个压缩器。

尽管在默认情况下，大部分节约 CPU 的选项都是开启的，还是有很多种其他方式来降低 CPU 的使用率。

请考虑下面这些：

- 如果你仅仅只使用了加载进 Harmor 中的图片或者音频中的一部分，你可以把图片或者音频进行剪切，然后只加载你需要用的那部分。
- Harmonic 模糊会占用很多 CPU
- 如果你使用了再合成，那么你就不需要再使用 cubic interpolation
- 只激活你需要用到的小振荡器的数量，把剩余的都关闭
- 只激活你能听得见的效果器

## 第 27 章 . 例子

### 制造白噪声

白噪声是你几乎在所有的现代音乐中能找到的声音。白噪声可以用做声音音效，Build

ups, 渐入, 扫进, 扫出。在 Harmor 中, 没有白噪声预设, 所以我们在这个练习中我们来自己制作白噪声预设, 并且在这个练习中, 让我们来用一用相位随机这个功能 (左侧)

1. 加载默认预设



2. 将小振荡器的音调调到“oct”模式, 然后将频率调成 1.0000/9, 然后将失谐调成 1.000/6.这时, 音色听起来像是老电影中具有未来感的射线。



3. 然后将 Unison 的阶数调成 3。不要 panning, 将 Unison pitch thickness 以及 phase 跳到最大。设置 blurred distribution。这个时候我们就能听到白噪声效果了。
4. 现在我们想让白噪声处于 C4 这个位置。我们需要对音调进行修改。将上界设成+4 个八度。将下界设成-5 个八度, 在垂直的 B8 那里。



这样的话我们就能将音调的中心放在 C4 哪里了。不管我们按下的是哪个琴键, 最后的结果将始终都是 C4.

5. 将相位的 rand 设置成-100%
6. 为了补偿内置的布朗滤波, 我们提高高频的和声水平。我们开始在 -24.08dB, 终于+15.92dB.





7. 转到 ADV 标签，然后将 ramp 设置成 0。



在默认情况下，ramp 的参数是 25ms。这会推迟小振荡器产生作用的时间。不同的小振荡器从底部向上一个一个被激活。首先是频激活，然后就是一个接着一个小振荡器被激活，最后一个小振荡器被激活的时间则是 25ms 之后。当我们把这个参数调整到 0 的时候，冲击会更加有力。在大部分我为了研究相位而创建的例子中，我将 ramp 设成 0，这样的话更易于我们观察 Harmor 正在发生的一切。

现在我们就成功地用 Harmor 制作出白噪音啦！

## 第 28 章 . 开放式问题

在这一章中，我收集了在写这本书的时候产生的问题。我联系了 Image-Line 公司来询问这些问题，当我有了答案之后我会更新这本书。

第 28 页。默认的相位值是 180。为什么当导入一个 wav 文件之后相位就会跳到 0？（当我导入一个用 Sytrus 制作 wav 文件的时候，就不会跳转）

第 41 页。在 Unison alternate distribution 中，我发现 Uniform 和 Random 是完全一样的。为什么会这样呢？这是一个已知的 bug 吗？

第 57 页。模糊冲击和延迟时间。这个 Time blur decay 旋钮是干啥用的？

同一个部分，但是另一个问题：模糊冲击和延迟时间旋钮，100%代表什么？

第 60 页。弹拨音替换模糊衰减，但是个东西的作用是什么？我玩不转这玩意儿。

第 132 页。图像打磨，也就是旋钮 sharp，当回放到达窗口的末尾的时候，它就会失去他的设置。这个这么避免？这是个已知 bug 吗？

第 140 页。创建一个 bouncing loop time envelope。这个功能和 prepare image time envelope 之间的区别是什么？

第 146 页。Filter envelope amount。为什么这个 Editor target 不是双极的？

第 163 页。ADV 标签中的自动增益模式。我不知道这个怎么用。希望可以给我一点点提示。

第 164 页。随机性。我对 Harmor 官方文档中对这一块的解释苦思冥想了好久，但是完全看不懂。相反，通过我自己的不断尝试，好像看懂了一些。我到底是对了，还是完全错了？

第 164 页。Performance versus quality, safety 旋钮。这个旋钮的功能是什么，能告诉我，我怎么才能知道它的的确确在工作呢？

▶ Global LFO amount

▶ Global LFO speed

为什么有时候经过编辑的 Editor targets 显示的是一个全三角，而有的时候三角里面却是空的呢？