emagic

es2

Software Instruments

>> Version 1.0, January 2002

>> English Edition

User Manual

License Agreement

Important! Please read this licence agreement carefully before opening the disk seal! Opening of the disk seal and use of this package indicates your agreement to the following terms and conditions. Emagic grants you a non-exclusive, non-transferable license to use the software in this package.

You may:

1. use the software on a single machine.
2. make one copy of the software solely for back-up purposes.

You may not:

1. make copies of the user manual or the software except as expressly provided for in this agreement.
2. make alterations or modifications to the software or any copy, or otherwise attempt to discover the source code of the software.
3. sub-license, lease, lend, rent or grant other rights in all or any copy to others.

Except to the extent prohibited by applicable law, all implied warranties made by Emagic in connection with this manual and software are limited in duration to the minimum statutory guarantee period in your state or country from the date of original purchase, and no warranties, whether express or implied, shall apply to this product after said period. This warranty is not transferable-it applies only to the original purchaser of the software. Emagic makes no warranty, either express or implied, with respect to this software, its quality, performance, merchantability or fitness for a particular purpose. As a result, this software is sold “as is”, and you, the purchaser, are assuming the entire risk as to quality and performance. In no event will Emagic be liable for any direct or indirect damages resulting from any defect in the software or documentation.

This agreement will terminate if you fail to comply with any term or condition in this agreement.

This agreement shall be governed by the laws of the Federal Republic of Germany.

Chapter 1
Welcome…

…and thank you for choosing the ES2 Emagic Synthesizer Two. The ES2 is one of the most versatile virtual-analog synthesizers ever designed. The features and sonic potential of many well-respected, rare, and vintage analog synthesizers pale in comparison to the power offered by the ES2.

Don't be fooled by the compact and easy-to-understand user interface of the ES2. Its lack of additional menu items, windows and lengthy parameter lists may give you the impression that its sonic power might not be that immense. Belying this simple appearance, the ES2 offers facilities that exceed those found in most of the "legends" of analog synthesizer history.

Following is the vast list of the ES2's key features. If you're a beginner, this brief description of the ES2's key features might initially seem a little overwhelming. Don't be thrown by this! And don't fret if you're unfamiliar with the terminology used here—the manual will explain everything. We have also included two tutorial chapters in this manual to help you learn the ropes of synthesis, and as you become more familiar with the ES2, you'll find that it's not at all complicated to use. Additionally, the ES2 ships with a large set of sounds we're sure you'll enjoy checking out. These presets can also be used as a basis for your own sound design visions.

If you're new to synthesizers, please invest the time in reading the *Synthesizer Basics* section, on page 97.

Most Spectacular Features

So, for the synthesizer connoisseur, here's the "brief" list of key features to whet your appetite:

The ES2 features an advanced modulation matrix in addition to a number of "hard-wired" modulation routings. Basically, the concept of combining any modulation *source* with any modulation *target*, is as almost as old as the synthesizer itself. Most important to the concept is a huge set of modulation *targets*, *sources*, a sufficient number of modulation *channels* and modulation *processes* which can be "inserted" into modulation *channels*. The ES2—featuring 10 modulation channels—represents the new "standard" when it comes to matrix modulation. You can modulate parameters such as filter resonance or the intensity of Cutoff Frequency modulation by Oscillator 1.

The three-Oscillator concept is reminiscent of the Minimoog and EMS VCS 3. The oscillators can be synchronized and ring-modulated. Pulsewidth modulation is also possible. Oscillator 1 can be modulated in frequency by Oscillator 2, and is capable of producing FM synth sounds.

Further to the "classic, standard" waveforms, the ES2's Oscillators also feature 155 single-cycle waveforms, known as "DigiWaves". Each has a totally different sonic color. The *DigiWave* Parameter can be modulated, allowing dynamic cross-fades between the waves. Intense use of such modulations will "scroll" or "step" through several waves, resulting in sounds somewhat like those of wavetable synthesizers.

Two dynamic multi-mode filters, with a number of selectable slopes guarantee the fattest analog sounds. The filters can be mixed freely in parallel and series modes. The filters can be overdriven by the *Drive* control.

Distortion, *Chorus*, *Phaser* and *Flanger* effects are included. The unique *Unison* mode delivers a density of sound normally only associated with big analog synthesizers, including classic machines such as; the Roland Jupiter 8, the SCI Prophet V and Oberheim OB 8, amongst others. The great thing about the

ES2's *Unison* mode is that it can be used in *both* monophonic and polyphonic modes. The unison voices are intelligently spread across the entire stereo spectrum, but you can still modulate the panorama position of the voices with any modulation source. This latter facility is unprecedented.

As with the classic Yamaha CS Series analog synthesizers, a sine wave derived from Oscillator 1 can be mixed directly into the dynamic stage to "fatten up" the sound. This makes the penalties normally associated with the use of highpass, bandpass and band rejection filters much less of an issue.

A virtual-analog Synthesizer

The ES2 is termed a software instrument, as its surface only appears as a window on the computer screen. The sound is calculated in realtime by the computer's CPU (Central Processing Unit). The ES2 is a virtual "analog" synthesizer. Its design and feature set (despite the DigiWaves) is reminiscent of the analog synthesizers of yesteryear, although all processing in a computer is digital, of course. Two things are "typical" of virtual-analog synthesizers:

- The feature set of the ES2 Oscillators emulates those of analog synthesizers. They deliver sawtooth, rectangular and pulse waves, but no samples, i. e.—no digital recordings of "original" instruments. The ES2 features 100 modulateable "DigiWaves". The oscillators of virtual-analog synthesizers often provide realtime features such as synchronization or pulse width modulation. The Filters and Envelope Generators feature a design reminiscent of analog synthesizers. The Filters can be overdriven, and the Envelope Generators often only feature a handful of parameters. Being digital, and simplified, however, the envelopes are capable of very fast, percussive transient output.
- Each important parameter is represented by a dedicated control on the virtual instrument's "surface". The use of these controls is convenient and fast, and it's fun to play with the controls live and in realtime.

Virtual *hardware* synthesizers have been available since the middle of the 1990's. These devices feature sonic characteristics comparable with those of genuine analog synthesizers, and are capable of convincing the most critical of musicians that they're listening to the real thing. The introduction of these instruments saw the disappearance of a number of limitations associated with genuine analog synthesizers. Among such limitations: tuning instability, the lack of program memory and automation functions, high prices and also the lack of a "musical" velocity sensitivity implementation. The latter being a point sorely lacking in most classic analog synthesizers.

Extremely Low Latency, Fast Attack and High Polyphony

The earliest software-based (CPU-generated) virtual-analog synthesizers suffered from a number of problems. Among these were latency troubles, limited polyphony and other performance issues.

The growing processing power of computer CPU's over recent years, coupled with the processor-efficient DSP programming of the ES2 and highly optimized soundcard drivers, has improved the situation somewhat. The ES2's seamless integration into the audio signal flow of Logic guarantees extremely fast response times to "live" keyboard performances. The ES2 can be played while Logic is stopped or playing back. It responds just like a hardware synthesizer with its own keyboard. With the ES2, there are no compromises when it comes to complex voice architectures and the playback of multiple voices simultaneously. Using Logic Platinum and a fast computer, you can arrange several ES2 "units" simultaneously, each playing up to 32 voices.

Efficient Use of Processing Power

The ES2 was painstakingly programmed to make intelligent use of processing resources—i.e. unused "modules" don't eat processing power when they're not in use. Many "real world" musical sounds require a mere handful of modulators or oscillators. For such sounds, the processing power of the CPU is used economically. This allows a high polyphony count (the playback of lots of notes simultaneously—up to 32 per unit), even when the CPU is used for the simultaneous calculation of effects.

The ES2 has even been optimized for specific processor types! From its initial release version, the ES2 has been specifically optimized for Pentium III, Pentium IV, Athlon XP and Power Macintosh G4 processors.

You'll find tips on this in the *Handling Processing Power Economically* section, on page 56.

Integration in Logic

As the ES2 is fully integrated into the program structure of Micrologic AV, Logic Audio, Logic Gold and Platinum, it can be totally automated. In practical terms, this means that every movement of any parameter knob, slider or switch on the ES2's front panel can be recorded and played back.

All available effect plug-ins can be inserted directly into the *Audio Instrument* channel strips, and/or you can make use of the auxiliary *Sends* in order to process the ES2's output with further effects, such as reverbs or delays etc. Logic's *Bounce* function allows the entire virtual synthesizer (ES2) track to be recorded as an audio file (at up to 24 Bit depth and a 96 kHz sampling rate). This "*Bounced*" audio file can then be assigned (as an audio region) to a standard Audio Track in Logic, allowing you to reassign the available processing (CPU) power for further synthesizer tracks.

Have Fun!

Beyond the dry, electro-technological topic of sound synthesis theory, dealing with waveforms, ring modulation, oscillator synchronization, frequency modulation and low pass filtering, you'll discover sonic possibilities you've never experienced. Synthesizer fanatics are addicted to diving into the minutiae and playing with the "fundamentals" of musical sounds. This "electron microscope" view into the world of sounds provides a great "kick" for "synth-heads". And before you back away nervously from these types, don't discount the possibility that you'll end up like them one day!

We wish you many years of inspired synthesizer playing, successful music production and a great deal of fun with the ES2 Emagic Synthesizer Two!

Your EMAGIC Team

Chapter 2
Installation

2.1 What the Package Includes

Your ES2 Emagic Synthesizer Two package contains the following:

- The Emagic *Software CD* with the current Logic version. There's also a collection of settings (sounds) for the ES2.
- This user's manual.
- The sealed *Registration Return Envelope*. Please do not open it until you have read the following paragraph. The sealed *Registration Return Envelope* contains the *International Registration Card*. Attached to the card, you will find a barcode sticker with a number on it.

Sealed Envelope and Registration Card

The envelope is sealed. The act of opening the envelope indicates your agreement with, and acceptance of, our licensing conditions and terms of trade.

Please open the envelope carefully along the seal. Do not tear the envelope. It can be resealed, and reused for registration, by enclosing the *International Registration Card*, and posting it, if you wish to register by mail.

A barcode sticker with a number is attached to this card. This number is a temporary authorization code. It must be typed into the *XSKey Authorization window*, as described in *The XSKey Authorization Window* section, on page 18. After you have entered this authorization code, you may use the ES2 for a period of one month, with no functional restrictions. During this time (preferably right now), you will need to register your product. After registration, you will receive a further authorization code, for unlimited use of the ES2. This "unlimited use" authorization code must also be entered in the *XSKey Authorization window.*

Registering Online

If you have Internet access, please register the ES2 online. This is the simplest and fastest method.

Keep the *International Registration Card* and the serial number of your XSKey handy. You'll need both to register your ES2. You can find the XSKey serial number on the barcode stickers that came with your Logic 5 version, and under Help > XSKey Authorization (Windows) or > XSKey Authorization (Mac OS).

Start your web browser and navigate to:

www.emagic.de/registration

- Input the requested data.
- A successful online registration will be indicated by an onscreen confirmation message and via e-mail.
- After a short processing period, you will receive your authorization code for unlimited use via e-mail.

Registering by Mail

If you don't have Internet access, you may register by mail.

- Please enter the appropriate details on the *International Registration Card* carefully, and completely.
- Attach one of the XSKey serial number stickers (as supplied with your Logic package) onto the respective field of the card.
- Insert the *International Registration Card* into the *Registration Return Envelope.*

Please allow for a period of 10 working days to process the card. You will receive the authorization code for unlimited use by mail. The online registration method is preferable, and more convenient.

Updates and Support

The newest software versions are available for download from our website. Should you encounter any difficulties, we also offer a free online support center. You may also consult one of our technicians by telephone.

- Visit www.emagic.de
- Support via our Hotline:
  In the USA: e-mail: support@emagicusa.com
  phone 1-530-477 1050, fax 1-530-477 1052
  In Germany: e-mail: support@emagic.de
  phone +49-(0)4101-495-110
- The InfoWeb is an almost inexhaustible resource for all Emagic products. Its clear layout offers instant access to up-to-date insider information, answers to compatibility issues and troubleshooting help. This is the URL:

http://www.emagic.de/english/support/infoweb/

2.2 Getting Started

The ES2 is integrated within the Logic software. To make use of the ES2, an installed copy of Logic Audio, Gold, Platinum or MicroLogic AV 5 (or higher) is required.

Please start the Logic 5 installer located on the Emagic Software CD which ships with the ES2. Follow the on-screen installer instructions. The installer updates the ES2 and Logic 5 software components, if necessary.

Your stored Plug-In settings will not be erased by this process.

ℹ If your Logic 5 series program (i. e. MicroLogic AV, Logic Audio, Logic Gold, Logic Platinum) is not available as an installer option, the most up-to-date version is already installed on your hard drive.

Depending on the volume number of the Emagic Software CD, the installer also offers the option to install other programs of the Logic 5 series. These can then be tested by switching the XSKey to the respective demo setting.

The ES2 is copy protected and authorized via the XSKey (Expandable System Key). This is how it works:

The XSKey Authorization Window

Open the *XSKey Authorization window* by selecting **Help > XSKey Authorization** (Windows) or > **XSKey Authorization** (Mac OS). The window shows the authorization status for all *available* software instruments and add-on modules. The authorization code for each is stored in the XSKey. Please take good care of your XSKey!

The window also shows the serial number of your XSKey. All codes, for all products, are entered in the "*Enter Authorization Code Here!*" field. Click once on the field to enter a code. The following describes the messages you may see in the *XSKey Authorization window.*

authorized:

The module is purchased, authorized and ready for "unlimited" use.

expiring in … days:

This module is in a fully functional demo period for the specified number of days. Purchase, and registration, with Emagic will provide you with a code to permanently authorize the module. If no code is supplied within the time period, the demo will expire.

It is recommended that you do *not* attempt to change the date of the system clock during an active demo period, as this may reduce the time before the demo expires.

activate Demo…

The module is not active, but it is possible to enable it’s demo mode. To do so, click once in the desired “Activate Demo…” field. Please note that the first time it is started, the demo mode can not be stopped, and will continue to count down! If a permanent license/authorization code is not purchased within the demo period, use of the module will expire.

expired

The demo period is over. It is not possible to use the module until a valid license code is entered.

empty field

The module is not active, and no demo mode is available. The only way to activate such modules is by entering a license code.

2.3 The “Audio Instrument” Object Type

The ES2 can be found in the hierarchical menu of the top insert slot of any Audio Instrument channel, once the authorization or demo mode is engaged. Routing possibilities for the ES2 are determined by the version of Logic used. The number of ES2 instances which can be run simultaneously is dependent on the availability of computer processing resources, and also on the version of Logic used.

In Logic’s Track Mixer (or Audio Environment Layer in Logic Audio, Gold and Platinum), there is an audio object type called an “Audio Instrument”. Audio Instrument objects appear as channel strips in the Environment window's Audio layer and the Track Mixer window. These objects allow synthesizer and other instrument plug-ins to be inserted in their top insert slot. Among these are Emagic's ES1 and ES2 synthesizers, the EVP 88 electronic piano and the EXS24 sampler.

The default song—the song that opens automatically if you move your Autoload song away from the Logic program folder—features a number of pre-configured Audio Instruments.

An Audio Instrument is an audio object (an Audio Track in MicroLogic AV) with the **Cha** parameter switched to one of the **Instruments**. Any audio object can operate as an Audio Instrument by changing the **Cha** parameter in the Object Parameter box. Audio objects are created by selecting **New > Audio Object.** To create a new Audio Instrument in MicroLogic AV, you can simply select **Track > Create Audio Instrument**.

Important!

You can only insert the ES2 Synth plug-in into the *top* slot of an Audio Instrument channel.

Loading and Playing the ES2

The ES2 comes with a library of ready-to-play sounds known as *Settings*. Following installation of the ES2, the *Settings* can be found in the *Logic* > *Plug-In Settings* > *ES2* folder. Please follow these steps in order to audition the ES2 *Settings*:

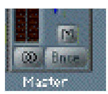
- Start Logic (or MicroLogic AV).
- Select or create a new Audio Instrument object (see above).
- Select the ES2 from the **Stereo > Logic** group of the plug-in list which appears after clicking on the first (top) plug-in slot of the Audio Instrument. Once selected (highlighted), release the mouse button.
- Launch the Arrange window, if not already open, via the Windows menu or Key Command.

- Select the corresponding Audio Instrument object—i. e. the one with the ES2 inserted—in the Arrange Window's Track List. This selection will activate the object, enabling it to receive MIDI data from your keyboard.
- The synthesizer is now ready to play.
- Launch the Mixer window or Audio layer in the Environment, if not already open, via the Windows menu or Key Command.
- Double-click on the blue "ES2" label in the top slot of the Audio Instrument object to open the plug-in window.
- You may select any of the ES2 *Settings* by click-holding on the flip menu in the gray panel area (to the right of the **Bypass** switch).

2.4 The ES2 in the Audio Mixer

Dependent on the version of Logic installed on your system, several instances of the ES2 can be inserted into the various Audio Instrument objects. Therefore, the ES2 can be called "multi-timbral".

The output signal of the ES2 is always stereophonic. It is fed into the input of the Audio Instrument channel strip, where it can be processed via inserted plug-ins and/or sent to busses (as shown beside). Given a fast enough computer, you could conceivably arrange and mix an entire song using several Audio Instruments, such as the Emagic ES1, ES2, EXS24 or EVP88, or VST instruments. This has the added benefits of superior sound quality and timing as the signal never leaves the digital domain, and you can freely edit these software instrument parts, change the tempo and more, right up to the final mix.



The **Bounce** button found on the Master audio object allows you to write submixes of ES2 tracks—as an audio file—to disk at any time. Audio Instrument tracks which have been recorded in this way can then be used as normal audio tracks in the arrangement. You can make use of this facility to free up CPU resources when a song requires more processing power than your CPU is capable of delivering, and does not allow all desired ES2 tracks to be played in realtime.

All parameters of the ES2 and all associated Audio Instrument channel parameters—Volume, Pan etc.—can be fully automated by Logics track automation feature.

The Plug-in Window

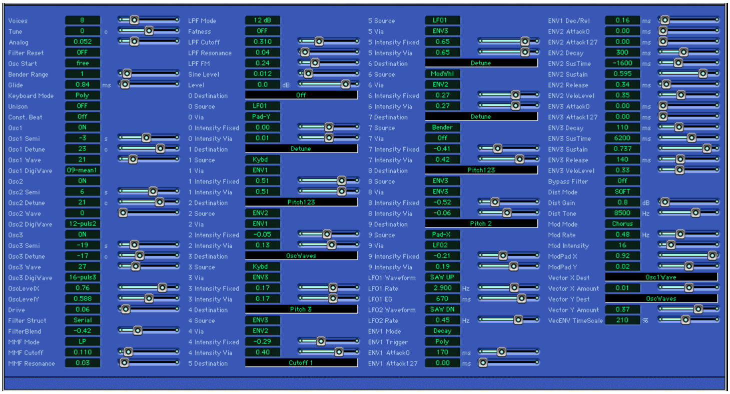
Hands-on operation of the ES2 is performed in the plug-in window. The plug-in window can be accessed by double-clicking on the blue ES2 label on an Audio Instrument object. When launched, the plug-in window allows access to all ES2 parameters. Each instance of the ES2 has its own plug-in window, which allows each to have discrete settings.

The plug-in window of the ES2.

Controls View Window

The ES2 parameters can also be displayed and edited in the Controls view. This displays all ES2 functions as a set of horizontal faders. Click-hold the **Editor** button in the gray area at the top of the plug-in window to switch the view modes. Tip:

With the PC version, it's possible to toggle between any two menu options with the right mouse button.



The controls view of the ES2.

Chapter 3

The ES2 Parameters

If given just a few words to explain the principles behind a subtractive synthesizer, it would go something like this: The *Oscillator* generates the oscillation (or waveform), the *Filter* takes away the unwanted overtones (of the waveform), and the *Dynamic Stage* sets the volume of the permanent oscillation (the filtered waveform) to zero as long as no key is pressed.

In an analog synthesizer, these three sections are commonly called the *VCO*, *VCF* and *VCA*, with *VC* being the abbreviation for *Voltage Controlled*, and the other letter(s) standing for *Oscillator*, *Filter* and *Amplifier*, respectively.

The basic parameters of a synthesizer are controlled (modulated) by voltages: pitch in the oscillator, timbre in the filter, loudness in the amplifier.

These voltages are generated by modulation sources. In the ES2, the Router controls which parameter is controlled by which modulation source.

Finally, the synthesizer's sound is refined by effects like distortion or Chorus.

Following this simple signal path, we would like to introduce you to the modules of the ES2.

Please note: You can reset each parameter to its default value by ⌥-click (Mac) or ctrl-click (Windows).

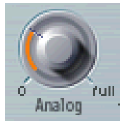
If you hold ⇧ before clicking and moving a control, its value can be fine-tuned.

3.1 Oscillators

Tune

Tune sets the pitch of the ES2 in cents. 100 cents equals a semitone step. The range is ±50 cents. At a value of **0 c** (zero cents), a^1 is tuned to 440Hz or "concert pitch".

Analog



Analog alters the pitch of each note, *plus* the Cutoff Frequency in a random fashion. Much like polyphonic analog synthesizers, all three Oscillators used by each synthesizer voice maintain their specific deviation but are shifted by the same amount randomly. Medium values simulate the tuning instabilities of analog synthesizer circuitry which can be useful in achieving that much sought after "warmth" of the ES2's analog hardware counterparts.

If the ES2 is set to **mono** or **legato**, **Analog** is only effective with **unison** activated. In this case **Analog** sets the amount of detuning of stacked voices.

If **voices** is set to **1** and **Unison** is not activated, the **Analog** parameter has no effect.

Read more on this parameters in the *Keyboard Mode (Poly/Mono/ Legato)* section, from page 27 onwards.

CBD (Constant Beat Detuning)

With **Fine Tuning**, you can detune Oscillators 1, 2 and 3 in Cents (percentages of a semitone). The detuning results in "beats" (phasing), the speed of which is determined by the difference between the two Oscillator frequencies (given that these frequencies are nearly identical). The higher the pitch, the faster the phasing beats are. High notes may therefore seem to be further out of tune than lower notes.

The **Constant Beat Detuning (CBD)** parameter matches this natural effect by detuning the lower frequencies in a ratio proportionate to the upper frequencies. Beside disabling

Constant Beat Detuning (CBD) altogether, four values are at your disposal: **25**, **50**, **75**, **100 %**. If you choose 100%, the phasing beats are (almost) constant across the entire keyboard range. This value, however, may be too high, as the lower notes might be overly-detuned at the point where the phasing of the higher notes feels "right". Try lower values for **CBD** (and **detune**, of course) in cases where the bass appears to be a little too far out of tune.

The center of **CBD** is C3 (middle C): Its detuning stays the same, regardless of the **CBD** value.

Glide

The **Glide** parameter controls the portamento time. This is the time it takes for the pitch to travel from one note to another. Glide behavior depends on the value of the **Poly**, **Mono**, **Legato** **Keyboard Mode** parameter setting.

If keyboard mode is set to **Poly** or **Mono**, and **Glide** is set to a value other than **0**, portamento is active. If keyboard mode is set to **Legato**, and **Glide** is set to a value other than **0**, you have to play legato (press a new key while holding the old one) to activate portamento. If you do not play legato, portamento will be inactive. This behaviour is also called "fingered portamento".

Bend Range

Bend Range determines the pitch range in semitones for pitch bend modulation. The range is ±12 semitones.

Keyboard Mode (Poly/Mono/Legato)

A *polyphonic* instrument allows the simultaneous playing of several notes—e. g. an organ or piano. Many synthesizers are *monophonic*, especially the older ones. This means that only one note can be played at a time, much like a brass or reed instrument. This shouldn't be viewed as a disadvantage in any way, because it allows playing styles that are not possible with polyphonic keyboard instruments.

You can switch between monophonic and polyphonic modes by clicking on the **Poly** and **Mono** buttons. **Legato** is also monophonic, but with one difference: The Envelope Generators are only retriggered, if you play staccato (release each key before playing a new key). If you play legato (press a new key while holding the old one), the Envelope Generators are only triggered with the first note you play legato and then continue their curve until you release the last legato played key. If you switch to **Mono**, legato or staccato playing does not matter: The Envelope Generators are retriggered with every new note.

Please note: If you switch to **Legato**, you have to play legato to hear the **Glide** parameter taking effect.

ℹ On several monophonic synthesizers, the behavior in **Legato** mode is referred to as *Single Trigger*, while **Mono** mode is referred to as *Multi Trigger.*

Voices

The maximum number of notes that can be played simultaneously is determined by the **Voices** parameter. Maximum value for **Voices** is **32**.

The value of this parameter has a significant impact on the computer processing resources demanded by the ES2 when played at its maximum polyphony. Reduce this value to the number of voices that you actually *require* for the part. Setting it to a higher value places higher overheads on the CPU, and wastes resources.

Unison

A forte of polyphonic analog synthesizers has always been "unison" mode. Traditionally, in unison mode, classic analog polysynths run monophonically, with all voices playing a single note simultaneously. As the voices of an analog synthesizer are never perfectly in tune, this results in an extremely "fat" chorus effect with great sonic depth. Switch the ES2 to **Mono** or **Legato** and switch on **Unison** in order to achieve and hear this effect. The intensity of the unison effect depends on the number of **Voices** selected. Remember that the amount of processing power required is correlated to the number of voices! The intensity of detuning (voice deviation) is set via the **Analog** parameter.

In addition to this classic monophonic unison effect, the ES2 also features a polyphonic unison effect. In poly/unison, each note played is effectively “doubled”, or more correctly, the polyhony value of the **Voices** parameter is halved. These two voices are then used for the single triggered note.

Switching on **Poly** and **Unison** has the same effect as setting the ES2 to **Mono**, **Unison** and **Voices** = **2**, except that it can be played polyphonically.

Oscillator Start

The oscillators can run freely, or they can begin at the same phase position of their waveform cycle, each time you hit a key (every time the ES2 receives a “note on” message).

When **Osc Start** is set to **free**, the initial oscillator phase start-point is random, with each note played. This gives the sound a more life and a less static feel—just like an analog hardware synthesizer. On the other hand, the output level may differ each time you play a note, and the attack phase may sound less punchy.

If you set **Osc Start** to **soft**, each initial oscillator phase will start at a zero crossing, each time a note is played. This mimics the sonic character of a normal digital synthesizer.

If **Osc Start** is set to **hard**, each initial oscillator phase begins with the highest possible level in its waveform cycle, each time a note is played. This “punch” is only audible if the **ENV3 Attack Time** is set to a minimal value—i. e. a very fast attack. This setting is highly recommended for electronic percussion and hard basses.

Osc Start **soft** and **hard** result in a constant output level of the initial oscillator phase every time the sounds is played back. This may be of particular importantance when using Logics bounce feature, at close to maximum recording levels.

Filter Reset

If you increase the filters **Resonance** parameter to higher values, it will begin to internally feed back and will start to self-resonate. This resonance results in a sine oscillation, which you may be familiar with, if you’ve used subtractive synthesizers before.

In order to start this type of oscillation, the filter requires a “trigger”. In an analoge synthesizer, this “trigger” may be the noise floor or the oscillator output. In the digital domain of the ES2, noise is all but eliminated. As such, when the oscillators are muted, there is no input signal routed to the filter.

When Filter Reset is engaged, however, each note starts with a “trigger” which is used to make the filter resonate immediately.

Frequency Switch

Switches the pitch in semitone steps over a range of ±3 octaves. As an octave consists of 12 semitones, settings ±12, 24 and 36 represent octaves. You can click on these marks to reach the corresponding octaves quickly.

The value display works as follows: the left numbers show the semitone (**s**) setting, the right numbers show the cent (**c**, 1 cent = 1/100 semitone) setting. You can adjust these two values independantly. So an oscillator with the value **12 s 30 c** sounds an octave (12 semitones) and 30 cents higher than an oscillator with the value **0 s 0 c**.

The fifth (7 semitones), and all settings that correspond with harmonics of an Oscillator set to 0 semitones (19, 28 semitones) result in “harmonic” sounds.

Muting Oscillators

By clicking on the green numbers to the right of the Oscillators, you can mute and un-mute them independently. This saves processor power.

Wave

Each of the three Oscillators features a rotary switch where you can select the waveform. This is responsible for the basic harmonic content and tone color of a sound. Oscillators 2 and 3 are almost identical to each other, but different from Oscillator 1. Oscillator 1 is capable of generating a sine wave, the frequency of which can be modulated in the audio range, for true FM synthesis sounds. Oscillators 2 and 3 can be synchronized to, or be ring-modulated with, Oscillator 1. They also feature rectangular waves with freely definable fixed pulse widths plus pulse width modulation (PWM) facilities. Via the *Router*, the rectangular and pulse waves of Oscillator 1 can be modulated in width in conjunction with the synchronized and ring-modulated rectangular waves of Oscillators 2 and 3.

With the **Filter** switch, you can disable the entire Filter section. This make listening to the pure oscillator waveforms easier.

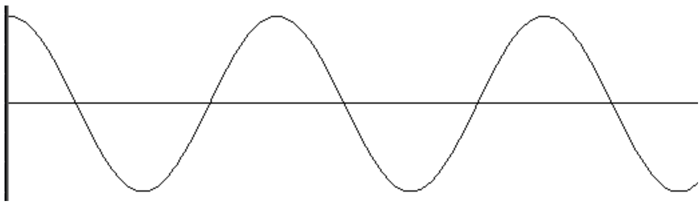
Oscillator 1 Waveforms

Oscillator 1 outputs "standard" waveforms—pulse, rectangular, sawtooth and triangular waves—or, alternately, any of the 155 available DigiWaves. It can also output a pure sine wave.

The sine wave can be modulated in frequency by Oscillator 2—i. e.—in the audio frequency range. This kind of linear frequency modulation is the basis on which FM synthesis works. FM synthesis was popularized by synthesizers such as Yamaha's DX7 (the architecture of which is much more complex, when it comes to FM synthesis).

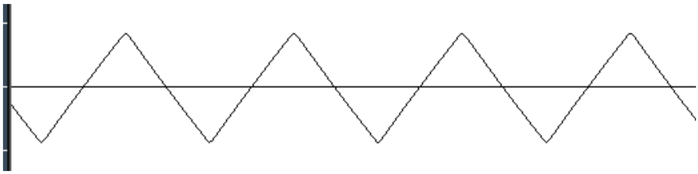
With a click on the Oscillator number, the output of Oscillator 1 can also be turned off. Even when the Oscillator 1 is turned off, it is still available for use as a modulation and synchronization source for Oscillators 2 and 3.

We will now take a closer look at the different waveforms of Oscillator 1.



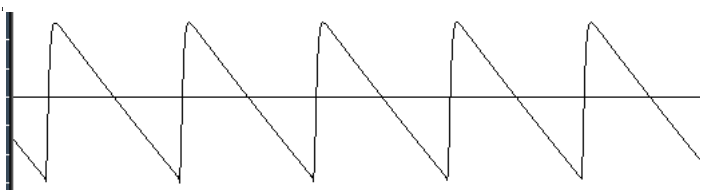
Screenshot of ES2's Oscillator 1, sine wave selected. The sample was created with Logic's Bounce *function and is displayed in the Sample Editor. The sine wave shown is in its basic oscillation form. It contains no harmonics. According to the theorem of Jean Baptiste Fourier, all regular waveforms can be interpreted as the sum of sine oscillations with defined frequency, amplitude and phase position, with their frequencies being "harmonic"—i. e. having integer frequency ratios.*

With a Fourier transformation, complex oscillations can be divided into their basic sine components. In additive synthesis, complex oscillation forms can be re-synthesized. The most simple additive synthesizer is the drawbar organ (the Hammond organ, for instance). With such an organ, you can mix 9 sine “choirs” with drawbars. Try selecting sine waves for all three Oscillators and semitone settings –12 (16’), 0 (8’) and +7 (5*1/3*’), and set all Oscillators to the same level. Select an organ envelope, and voila—a classic organ sound!



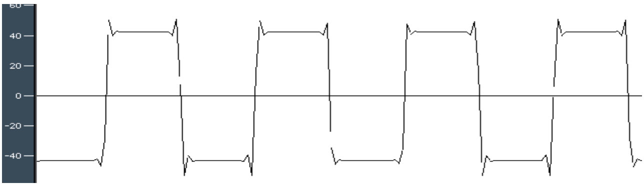
Screenshot of the ES2’s triangular wave, created and shown as above. The triangular wave only contains odd harmonics (no octaves), the amplitudes of which decrease Square-proportionately to their number. This means that its sound has few overtones. This corresponds with its appearance which is quite reminiscent of a sine wave.

Classic synthesizer literature encourages the use of the triangular wave for the creation of flute-like sounds. In the age of sampling, however, it’s pretty hard to sell a triangular wave as a “flute” sound to anyone.



Screenshot of the ES2’s sawtooth wave. The sawtooth wave contains all harmonics, the amplitudes of which decrease proportionately with their number.

Classic synthesizer literature indicates the use of the sawtooth wave to create a sound similar to that of a violin. The rich and full sound of the sawtooth wave is the most popular synthesizer waveform and serves as a basis for synthetic string and brass sounds. It is also handy for synthesized bass sounds.



Screenshot of the ES2's rectangular wave. The 50% rectangular wave contains all odd harmonics, the amplitudes of which decrease proportionately with their number. The pulse width can be set to any value and serves as a modulation address.

Classic synthesizer literature likens the rectangular wave to the sound of a clarinet, as the clarinet doesn't feature any even harmonics in a certain frequency range either. The typical "hollow" sound of the rectangular wave is achieved with its pulse width value set to 50%. Heavily low-pass-filtered rectangular waves are popular as techno bass sounds. Changing the pulse width to something around 75% results in a quite nasal sound, resembling an oboe.

DigiWaves

The ES2, as we've mentioned, not only features the most popular synthesizer waveforms, but also a selection of 100 additional waveforms, called DigiWaves.

You can select the digiwave by touching its name and moving the mouse vertically. You can select it numerically when you hold ⇧.

The number of the Digiwave is a parameter that can be modulated. By modulating the target **OscWave**, you scroll through the list of digiwaves. Choose modulation intensity and speed sufficiently low, to hear the single digiwaves being crossfaded. The digiwaves of the three oscillators can be modulated individually or all together. The modulation targets are discussed under *Osc Waves* section, on page 63 up to *Osc3 Wave* section, on page 64.

Thanks to its modulatable digiwave feature, the ES 2 can produce sounds resembling of the famous wavetable synthesizers that manufacturers PPG and Waldorf (and also the Korg Wavestation) are known for.

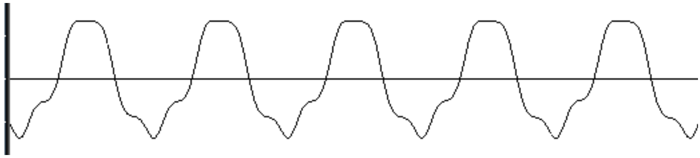
Linear Frequency Modulation

The principle of linear frequency modulation (FM) synthesis was developed in the late sixties and early seventies by John Chowning. It's such a flexible and powerful method of tone generation that synthesizers were developed, based solely on the idea of FM synthesis. The most popular FM synthesizer ever built is Yamaha's DX 7. FM synthesis is also found in other models of the Yamaha DX range and some Yamaha E-Pianos. In the discipline of pure FM synthesis, the ES 2 can't be compared with these synthesizers, but it can certainly achieve some of their "signature" sounds.

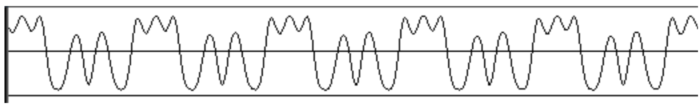
The DX7 featured 6 Oscillators (called "operators") per voice, with each controlled by a separate Envelope Generator. It's bell-like, synthetic electric piano sounds became de rigeur in popular music, especially in soul ballads (Whitney Houston and others).

Between the **Sine** setting (when the sine symbol is selected) and the FM symbol of the *Oscillator* 1 knob, a range is available which allows stepless control over the amount of frequency modulation. This parameter is also available as a modulation address. The frequency of Oscillator 1 can be modulated by the output signal of Oscillator 2. Whenever it outputs a positive "voltage", the frequency of Oscillator 1 will increase. Whenever it is negative, its frequency will decrease.

The effect is similar to an LFO modulation, being used to create a vibrato (a periodic undulation of the frequency) or a slow siren effect. In comparison to an LFO, Oscillator 2 does not oscillate slowly. In the audio domain, it oscillates a little faster than Oscillator 1 itself. Thus, Oscillator 1's oscillation also accelerates and slows down within a single phase, resulting in the distortion of the basic sine shape of Oscillator 1. This sine wave distortion has the side benefit of a number of new harmonics becoming audible.

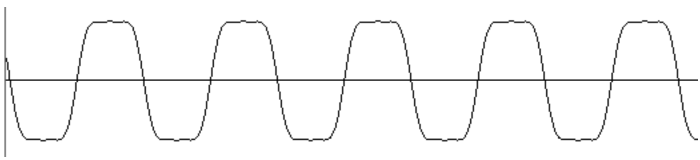


*Oscillator 1's frequency modulated sine wave, modulated by Oscillator 2 set to sine wave. Oscillator 2 was set to three times the frequency of Oscillator 1 (+19 semitones). The modulation intensity is low (***Wave** *control at about 12 o'clock). As the wavelength (i. e. the duration period) of the modulating Oscillator is a third of that of the modulated Oscillator, the sine is accelerated and slowed down three times within a phase.*

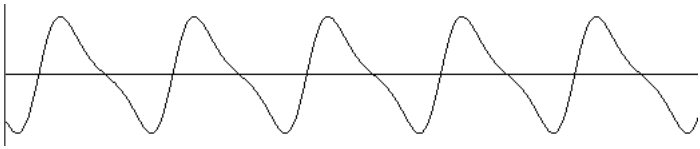


*Oscillator 1's frequency modulated sine wave, modulated by Oscillator 2 set to sine wave. Oscillator 2 was set to three times the frequency of Oscillator 1 (+19 semitones). The modulation intensity is much higher (***Wave** *control at about 3 o'clock). The distortion of the basic sine wave is much stronger, resulting in more harmonics becoming audible.*

The effect of frequency modulation depends on *both* the modulation intensity and on the frequency ratio of both Oscillators.



The upper graphic shows a slightly deformed sine wave of Oscillator 1, modulated in frequency by Oscillator 2 at double the speed of the carrier (Oscillator 1). The resulting waveform resembles a rectangular wave or clipped sine wave.



The upper graphic shows a slightly frequency modulated sine as the output signal of Oscillator 1, with the modulator frequency being identical to the carrier frequency. The resulting waveform resembles a low pass filtered sawtooth wave.

The resulting spectrum not only depends on the frequency modulation intensity and frequency ratio, but also on the waveform used by the modulating Oscillator (Oscillator 2). The modulation which takes place varies according to the waveform selection for Oscillator 2—it might even be an oscillation that is synchronized to Oscillator 1! Keeping the 100 available DigiWaves, countless combinations of modulation intensities and frequency ratios in mind, the frequency modulation of the two Oscillators delivers an infinite pool of spectra and tone colors.

Waveforms of Oscillators 2 and 3

Basically, Oscillators 2 and 3 supply the same selection of analog waveforms as Oscillator 1: sine, triangular, sawtooth and rectangular waves. The pulse width can be scaled steplessly between 50% and the thinnest of pulses, and can be modulated in a number of ways (See *Pulse Width Modulation* section, on page 38)

Oscillators 2 and 3 also offer the selection of:

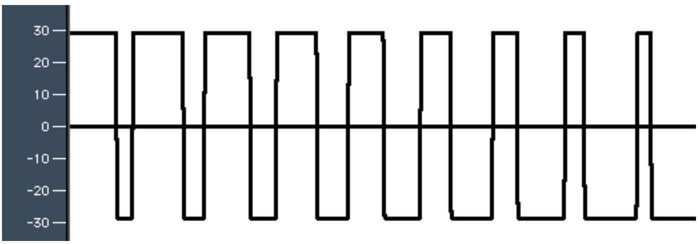
- a rectangular wave which is synchronized to Oscillator 1,
- a sawtooth wave synchronized to Oscillator 1,
- a ring modulator, which is fed by the output of Oscillator 1 and a square wave of Oscillator 2,
- and colored noise for Oscillator 3.

Synchronization and ring modulation afford the creation of very complex and flexible harmonic spectra. The principle of Oscillator synchronization is described on page 39, and ringmodulation on page 41.

Pulse Width Modulation

With Oscillators 2 and 3, you can scale the width of the pulses to any value. The spectrum and tone color generated by these Oscillators depends on the pulse width. The pulse width can be modulated. You can even modulate the pulse width of the square and puls wave of Oscillator 1, the pulse width of the synchronized pulse waves of Oscillator 2 and 3, and the square wave of Oscillator 2's ring modulator.

This width modulation is controlled in the *Router* (the modulation matrix). The pulse width is defined by the waveform rotary control. The graphic below shows a pulse wave, with the pulse width modulated by an LFO. You can clearly see how the width of the pulses changes over time.



Screenshot of an ES2 Sample, created with the ***Bounce*** *function of Logic Platinum. The pulse width is modulated. It is easy to see how the width of the pulses varies between a rectangular shape and very thin pulses. An LFO is selected as the modulation source, and its waveform is a sine wave. You can see about half a phase of the sine wave. If a rectangular wave had been selected for the LFO, you would see the pulse width periodically changing between the two fixed extreme values.*

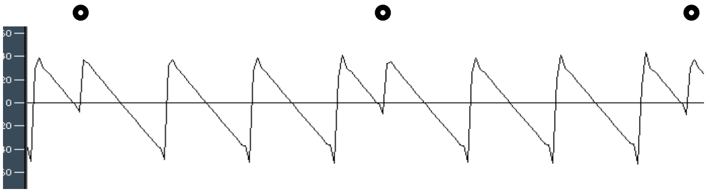
A pulse wave with its width modulation controlled by an LFO set to a sine wave makes a single oscillator sound vivid, undulating and rich with overtones. Sonically, this is somewhat like the sound of two slightly detuned, phasing oscillators. The effect sounds great with sustained bass and pad sounds. Select the intensity and speed of the modulation with care, as the overall volume (and level of the first partial) decrease and slight detuning occurs when the pulses become very thin (below 10%).

Pulse width modulations by velocity sensitive Envelope Generators sound very dynamic—a great effect, especially for percussive bass sounds.

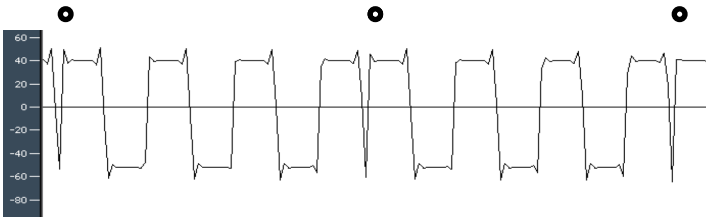
Sync (Oscillator Synchronization)

You can see that the rectangular and sawtooth waveforms also feature a **Sync** option. In this mode, the frequency of Oscillator 2 (or 3, respectively) is synchronized to the frequency of Oscillator 1. This does not mean that their frequency controls are simply disabled. They still oscillate at their selected frequencies, but every time that Oscillator 1 starts a new oscillation phase, the synchronized Oscillator is forced to restart its phase from the beginning, too. This is practical as long as the frequency of the synchronized Oscillator is significantly higher than that of Oscillator 1.

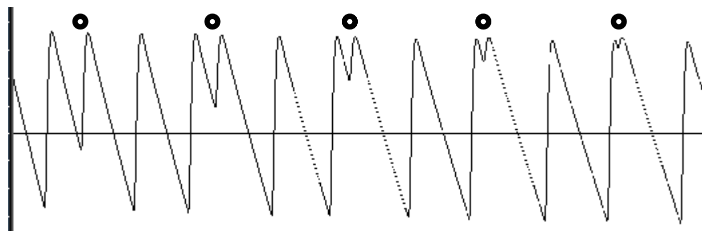
The graphic shows the output signal of Oscillator 2, set to a synchronized sawtooth wave. Its frequency is about three and a half times that of Oscillator 1. Oscillator 1 was set to 0 semitones, and Oscillator 2 to +22 semitones. The oscillogram clearly shows how the phase of Oscillator 2 is forced to restart after about three and a half phases. Between the pulses of Oscillator 1, Oscillator 2 runs freely.



Synchronized sawtooth wave as output by Oscillator 2. Oscillator 1 is set to 0, Oscillator 2 to +22 semitones. The dots in the graphic indicate the phases of Oscillator 1.



Synchronized rectangular wave as output by Oscillator 2. Oscillator 1 is set to 0, Oscillator 2 to +22 semitones. The dots in the graphic indicate the phases of Oscillator 1.



Output signal of Oscillator 2 (sawtooth), synchronized to Oscillator 1. The frequency of Oscillator 2 is modulated by an Envelope Generator, changing the duration of the phases over time. On the right-hand side, the sawtooth shapes are broader, due to the sinking frequency of Oscillator 2.

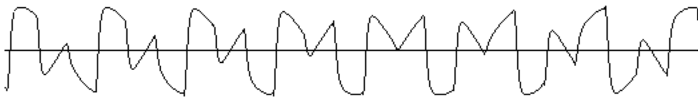
At regular intervals, defined by the phase duration of Oscillator 1, the waveform is forced back to its beginning. The dots in the graphic indicate the phases of Oscillator 1. The distance between these dots remains constant, as the frequency of Oscillator 1 is not being modulated.

Synchronized oscillator sounds are especially cool when the frequency of the synchronized oscillator is modulated by an Envelope Generator. This way, the number of phases within a section (phase) of the synchronization cycle always changes, and so does the spectrum. Typical oscillator sync sounds tend towards the "aggressive" and those "screaming leads" that synth manufacturers like to talk about.

Ring (Ring Modulation)

The waveform control of Oscillator 2 also features the **Ring s**etting. In this mode, Oscillator 2 outputs the signal of a ring modulator. This ring modulator is fed with the output signal of Oscillator 1 and a square wave of Oscillator 2. The pulse width of this square wave can be modulated.

A ring modulator has two inputs. At his output you will find the sum and difference frequencies of the input signals.



The graphic shows the output signal of the ring modulator, appearing as the output signal of Oscillator 2. The amplitude (or the elongation, to be more exact) of the output of Oscillator 2 changes with the phase of Oscillator 1. Oscillator 2 is set to a higher frequency than Oscillator 1. As the frequency ratio is odd (irrational), the resulting waveform always changes over time. The resulting spectrum is inharmonic and you won't hear a clearly defined pitch.

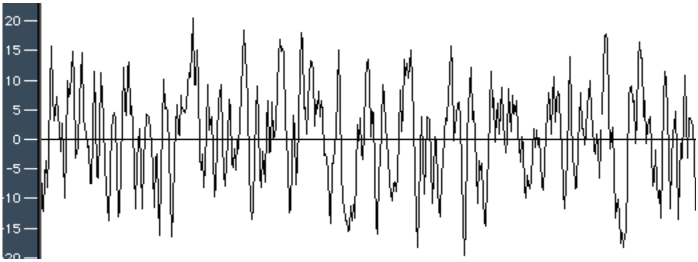
If you ring-modulate a sine oscillation of 200Hz with a sine oscillation of 500Hz, the output signal of the ring modulator will consist of a 700Hz and a 300Hz signal. Negative frequencies result in a change of the phase polarity of the output signals. With sawtooth and rectangular input signals, the output signal is much more complex, as these harmonically-rich waveforms produce a number of extra side bands.

Ring modulation is a powerful tool for inharmonic, metallic sounds as the spectra resulting from its use are inharmonic with almost every frequency ratio. The ring modulator was the tool of choice for bell-like sounds, and dates back to the early days of the synthesizer.

White and Coloured Noise (Oscillator 3 only)

Unlike Oscillator 2, Oscillator 3 is not capable of producing ring modulated signals nor sine waves. Its sonic palette however, is bolstered by the inclusion of a *Noise Generator*. By default, Oscillator 3's Noise Generator generates **White Noise**. White Noise is defined as a signal that consists of all frequencies (i. e. an infinite number) sounding simultaneously, at the same intensity, in a given frequency band. The width of the frequency band is measured in Hertz. Its name is analogous to "white light", which consists of a mixture of all optical wavelengths (all rainbow colors). Sonically, white noise falls between the sound of the consonant "F" and breaking waves (i. e.

“surf”). Synthesis of wind and seashore noises, or electronic snare drum sounds, requires the use of white noise.



Screenshot of a white noise sample, showing the output signal of Oscillator 3.

Oscillator 3 has more up its sleeve than the output of “neutral” sounding white noise—you also can make it hiss or rumble. Even better, you can modulate this sound-color in realtime without using the main filters of the ES2.

If the waveform of Oscillator 3 is modulated, (**Osc 3 Wave**) the color of the noise will change. It can be filtered by a dedicated high or low pass filter with a descending slope of 6dB/octave. At negative values, the sound becomes darker (red); The low pass filter can be tuned down to 18Hz with a setting of **–1**. When **Osc 3 Wave** is modulated positively, the noise becomes brighter (blue): At a value of **+1** for **Osc 3 Wave**, the high pass filter’s Cutoff Frequency is set to 18kHz. This filtering of the noise signal is independent of the main filters of the ES2, and can be automatically changed in realtime.

Oscillator Mix Field—The Triangle

By grabbing, and moving the cursor in the Triangle, you can crossfade beweeen the three Oscillators. This is pretty self-explanatory. Moving the cursor along one of the Triangle’s sides will crossfade between two Oscillators with the third Oscillator being muted.

The position of the cursor can be controlled via the vector envelope, just like the cursor position in the Track Pad (the Square), which we'll look at in *The Square* section, on page 81.

Note that the *vector envelope* features a *loop* function. This addition extends its usefulness, allowing you to view it as a luxurious pseudo-LFO with a programmable waveform. It can be used for altering the positioning of the Triangle and Squares' cursors. Read more about this in *Vector Mode* section, on page 81, and *The Vector Envelope* section, on page 82.

Triangle Values In Control View

Internally, the position of the cursor in the Triangle is described by two parameters (co-ordinates, actually) which are relevant when it comes to automating the Oscillator mix. These parameters, called **OscLevel X** and **OscLevel Y**—are visible in the Controls view. Don't confuse them with the X and Y positions of the *Square*.

If you intended to edit a sequence containing Oscillator mix automation data in the Hyper Editor, you would use the following MIDI controller values. Take a look at the information below to get a "feel" for how they operate.

In order to listen to Oscillator 1 only…

- choose **OscLevel X**= 1.0 (**MIDI**: **127**) and **OscLevel Y**= 1.154 (**MIDI**: **127**).

In order to listen to Oscillator 2 only…

- choose **OscLevel X**= 0.0 (**MIDI**: **0**) and **OscLevel Y**= 0.577 (**MIDI**: **64**).

In order to listen to Oscillator 3 only…

- choose **OscLevel X**= 1.0 (**MIDI**: **127**) and **OscLevel Y**= 0.0 (**MIDI**: **127**).

In order to listen to Oscillator 1 and 2 only…

- choose **OscLevel X**= 0.5 (**MIDI**: **64**) and **OscLevel Y**= 0.866 (**MIDI**: **95**).

In order to listen to Oscillator 1 and 3 only…

- choose **OscLevel X**= 1.0 (**MIDI**: **127**) and **OscLevel Y**= 0.577 (**MIDI**: **64**).

In order to listen to Oscillator 2 and 3 only…

- choose **OscLevel X**= 0.5 (**MIDI**: **64**) and **OscLevel Y**= 0.288 (**MIDI**: **32**).

All Oscillators have the same level at…

- **OscLevel X**= 0.667 (**MIDI**: **85**) and **OscLevel Y**= 0.577 (**MIDI**: **64**).

3.2 Filters

The ES2 features two dynamic *filters* which are equivalent to the "Voltage Controlled Filters" (VCF) found in the world of analog synthesizers. The two filters are *not* identical. *Filter 1* features several modes: Low Pass, High Pass, Band Pass, Band Rejection, Peak. *Filter 2* always functions as a low pass filter. Unlike Filter 1, however, Filter 2 offers variable slopes (measured in dB/octave).

i The **Filter** switch bypasses (switches off) the entire ***Filter*** section of the ES2. Disabling the Filters makes it easier to hear adjustments to other sound parameters, as the filters always heavily affect the sound. If the writing is green, the filters are engaged. It also reduces processor load.

"Series" and "Parallel" Filter Signal Flow

You can rotate the entire (circular) filter segment of the ES2 user interface. Press the button labelled **Parallel** or **Series** to do so. The label and position/direction of the filter controls clearly indicate the current signal flow.

Serially cabled Filters of the ES2.

In the position displayed above, the filters are cabled serially. This means that the signal of the *Oscillator Mix* section (Triangle) passes through the first filter, and then this filtered signal passes through Filter 2—if **Filter Blend** is set to **0** (middle position). See the *Filter Blend: Cross-Fading the Filters* section, on page 47 for a detailed description of this parameter.

The mono output signal of Filter 2 is then fed into the input of the dynamic stage (the equivalent of a “VCA” in an analog synthesizer), where it can be panned in the stereo spectrum, and then fed into the effects processor.

In the graphic below, the filters are cabled in parallel. If **Filter Blend** is set to **0**, you’ll hear a 50/50 mix of the source signal routed via Filter 1 and Filter 2, which is fed into the mono input of the dynamic stage. There it can be panned in the stereo spectrum, and then fed into the effects processor.

Parallel filter routing of the ES2.

Filter Blend: Cross-Fading the Filters

You can cross-fade the two filters. When wired in parallel, you'll find this quite "obvious" to look at and understand: If **Filter Blend** is set to its top position (value display in Controls View: –1), you'll only hear *Filter 1*. If **Filter Blend** is set to its lowest position (value display in Controls View: +1), you'll only hear *Filter 2*. In between these positions, the filters are cross-faded.

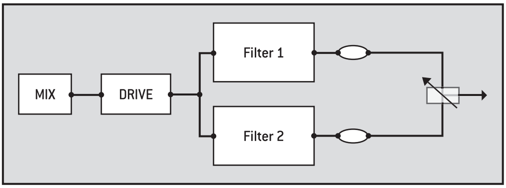
More often, you'll want to cable the filters serially. It should be noted that even in series mode, it's possible to cross-fade the filters! This is achieved through the use of controllable sidechains (bypassing lines). In this serial cabling scenario, the distortion circuits controlled by the **Drive** parameter, also need to be considered, as the distortion circuits are positioned before or in between the filters, dependent on **Filter Blend**. All fades occur flawlessly, in very fine steps.

The **Filter Blend** parameter can be modulated dynamically in the router!

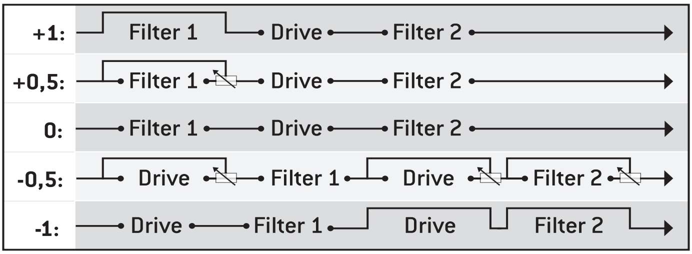
Filter Blend and Signal Flow

No matter whether parallel or series filter configurations are chosen, a **Filter Blend** setting of –1 always results in only *Filter 1* being audible. A **Filter Blend** setting of +1 will limit audibility to *Filter 2*. This is reflected in the user interface.

In conjunction with the overdrive/distortion circuit (**Drive**) and a series wiring configuration, the ES2's signal flow is anything but commonplace. The graphics illustrate the signal flow between the *Oscillator Mix* stage (the Triangle) and the dynamic stage, which is always controlled by ENV 3. The signal flow through the filters, the overdrives and the bypassing sidechains is dependent on the **Filter Blend setting**.



Filter Blend *in parallel Filter mode.*



Filter Blend *in Serial Filter mode. Between* **0** *and* **–1** *two distortion circuits are active—one "pre" each Filter.* **Filter Blend** *cross-fades up to three bypassing lines simultaneously.*

Filter Blend and Serial Filter Configuration Tips

1. With **positive** values for **Filter Blend**, Filter 1 is partially bypassed.
2. With **negative** values for **Filter Blend**, Filter 2 (the low pass) is partially bypassed.
3. At **zero** and **positive** values for **Filter Blend**, there is only one overdrive circuit for *both* filters.
4. With **negative** values, another overdrive circuit is introduced, distorting the output signal of the *Oscillator Mix* stage before it is fed into the first filter—i. e. **Filter Blend** = **–1.**

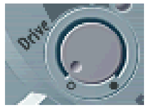
If **Drive** is set to **0**, no distortion occurs.

Filter Blend and Parallel Filter Configuration Tip

- The overdrive/distortion circuit is always wired *after* the *Oscillator Mix* stage and *before* the filters. The filters receive a mono input signal from the overdrive circuit's output. The outputs of both filters are mixed to mono via a **Filter Blend**.

If **Drive** is set to **0**, no distortion occurs.

Drive



The filters are equipped with separate overdrive modules. Overdrive intensity is defined by the **Drive** parameter. If the filters are connected in parallel, the overdrive is placed before the filters. If the filters are connected serially, the position of the overdrive circuits depend on the **Filter Blend** parameter—as described above.

Polyphonic Distortions In the Real World

The ES2 features a *Distortion* effect, equipped with a tone control, in the *Effects* section. Given the inclusion of this effect, you may be wondering what benefit the **Drive** function in the *Filter* section brings?

The distortion circuit in the *Effects* section affects the sum of the entire polyphonic synthesizer performance. Thus, more complex chords (other than major chords, parallel fifths and octaves) sound "rough", when using distortion. Every rock guitarist knows this. Due to these intermodulation distortions, distorted guitar playing is usually performed using few voices or parallel fifths and octaves.

In spite of this, the **Filter Drive** affects every voice individually —and when every voice in the ES2 is overdriven individually (like having six fuzz boxes for the guitar, one for each string), you can play the most complex harmonies over the entire keyboard range. They'll sound clean, without unwanted inter-modulation effects spoiling the sound.

Furthermore, "appropriate" **Drive** parameter settings lead to "character". To explain, the way analog filters behave when overdriven forms an essential part of the sonic character of a synthesizer. Each synthesizer model is an "individual" with regard to the way its filters behave when overdriven. The ES2 is extremely flexible in this respect, allowing for the most subtle "fuzz" through to the hardest distortions.

Finally, in series mode, the distortion always takes place *before* the low pass filter (Filter 2). As the low pass filter (Filter 2) can filter (cut) away the overtones introduced by the distortion, the *Drive* feature can be seen (and used) as another tool for deforming the Oscillator(s) waveforms.

To check out how the overdrive circuit between the filters works, program a sound as follows:

- Simple static waveform (a sawtooth),
- Filter set to **Serial** mode,
- **Filter Blend** set to **0** (center position),
- Set Filter 1 to **peak** **Filter Mode**
- Set a high **Resonance** value for Filter 1,
- Modulate **Cutoff Frequency 1** manually or in the *Router*,
- Set **Drive** to your taste,
- Filter away (cut) the high frequencies with Filter 2 to taste.

The sonic result resembles the effect of synchronized oscillators. With high resonance values, the sound tends to "scream". Modulate the **Resonance** of Filter 1, if you wish.

Filter Parameters

Cutoff and Resonance

With every low pass filter (in the ES2: **Lo** mode for Filter 1 and *all* of Filter 2's modes), all frequency portions *above* the **Cutoff**

Frequency are suppressed, or "cut off", hence the name. The **Cutoff Frequency** controls the brilliance of the signal. The higher the **Cutoff Frequency** is set, the higher the frequencies of signals that are allowed to "pass" through the low pass filter.

Resonance emphasizes the portions of the signal which surround the frequency defined by the **Cutoff Frequency** value. This emphasis can be set so intensively in Filter 2, that the filter begins to oscillate by itself. When driven to self-oscillation, the filter outputs a sine oscillation. The self-oscillation can be supported by the **Filter Reset** parameter. See *Filter Reset* section, on page 30 for details.

If you are new to synthesizers, experiment with a simple saw wave, using Oscillator 1, and Filter 2 (Low Pass Filter, **Filter Blend** = +1) on its own. Experiment with the **Cutoff Frequency** and **Resonance** parameters. You'll quickly learn how to emulate a number of recognizable sounds and will pick up the basic principles of subtractive synthesis intuitively.

The dynamic low pass filter is the most essential module in any subtractive synthesizer. This is why Filter 2 always operates in low pass mode.

As opposed to the filter and EQ effect plug-ins in Logic, the ES2's filters are dynamic, which means that the ***Cutoff Frequency*** parameter can be modulated extremely quickly and severely in real time—even on modulation signals in the audio frequency range.

The Chain Symbols

Playing the **Cutoff** and **Resonance** controls in realtime is one important key to expressive synthesizer sounds. So you will be pleased to know that you can control two filter parameters at once by clicking and moving on one of the three little chain symbols in the filter graphic.

- The chain between **Cut** and **Res** of Filter1 controls **Resonance** (horizontal mouse movements) and **Cutoff** (vertical mouse movements) of the first filter simultaneously.

- The chain between **Cut** and **Res** of Filter2 controls **Resonance** (horizontal mouse movements) and **Cutoff** (vertical mouse movements) of the second filter simultaneously.
- The chain between Filter1's **Cut** and Filter2's **Cut** controls **Cutoff** (vertical mouse movements) of the first filter, and **Cutoff** (horizontal mouse movements) of the second filter simultaneously.

Filter Slope

A Filter can not completely suppress the signal portion outside the frequency range defined by the **Cutoff Frequency** parameter. The "slope" of the filter curve expresses the amount of rejection applied by the filter beneath the Cutoff Frequency in dB per octave.

Filter 2 offers three different slopes: **12dB**, **18dB** and **24dB** per Octave. Put another way, the steeper the curve, the more severely the level of signals below the **Cutoff Frequency** are affected in each octave.

Fat

Increasing the **Resonance** value results in a rejection of bass (low frequency energy) when using low pass filters. The **Fat** switch compensates for this side-effect, giving a more "bassy" sound.

Filter Mode (Lo, Hi, Peak, BR, BP)

Filter 1 can operate in several modes, allowing the filtering away (cutting) and/or emphasis of specific frequency bands.

- A *Low Pass Filter* allows frequencies which fall below the Cutoff Frequency to "pass". As frequencies above the Cutoff Frequency are suppressed, it's also known as a High Cut Filter. When set to **Lo**, the Filter operates as a low pass filter. The slope of Filter 1 is 12dB/octave in **Lo** mode.
- A *High Pass Filter* allows frequencies above the Cutoff Frequency to "pass". As frequencies below the Cutoff Frequency are suppressed, it's also known as a Low Cut

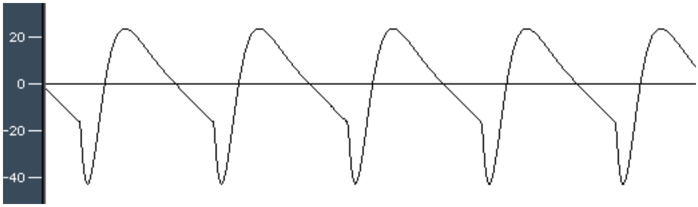
Filter. When set to **Hi**, the Filter operates as a high pass filter. The slope of Filter 1 is 12dB/octave in **Hi** mode.

- In **Peak** mode, Filter 1 works as a *Peak Type Filter*. This allows for an increase in level of a frequency band, the width of which is controlled by the **Resonance** parameter.
- The abbreviation **BR** stands for *Band Rejection*. In this mode, the frequency band (a range of adjacent frequencies) directly surrounding the Cutoff Frequency is rejected, whilst the frequencies outside this "band" can pass. The **Resonance** parameter controls the width of the rejected frequency band.
- The abbreviation **BP** stands for *Band Pass*. In this mode, only the frequency band directly surrounding the Cutoff Frequency can pass. All other frequencies are cut. The **Resonance** parameter controls the width of the frequency band that can pass. The band pass filter is a two-pole filter with a slope of 6dB/octave on each side of the band.

Impact of the Filters on the Waveform

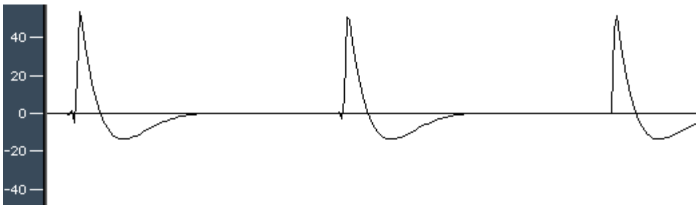
Below, you'll find a number of oscillograms of a sawtooth wave generated by Oscillator 1. These images illustrate the effect of the various modes of Filter 1. The Cutoff Frequency was set so that it is equal to the frequency of the first overtone (twice the frequency of Oscillator 1). The duration period and wavelength of this overtone (a. k. a. the second harmonic) is half as long as the duration period and wavelength of the first harmonic (i. e. the fundamental tone).

Low Pass Filter



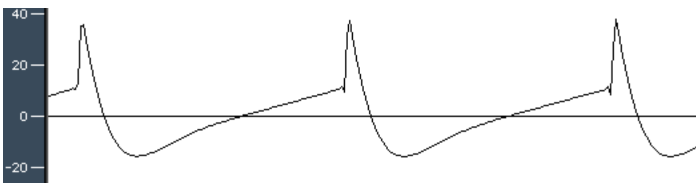
Low-pass-filtered sawtooth wave, Cutoff Frequency one octave above the frequency of the sawtooth. As the slope isn't infinite, harmonics are still visible. Note that the waveform is rounded.

High Pass Filter



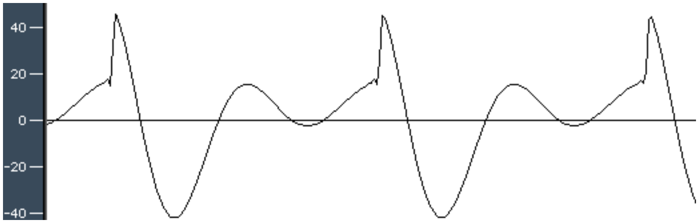
High-pass-filtered sawtooth wave, Cutoff Frequency one octave above the frequency of the sawtooth. The basic harmonic is rejected by about 12dB, as the slope is 12dB/Octave. Try to mentally add a sine curve with the basic wavelength to the graphic: The sum of both would result in a sawtooth wave.

Peak Type Filter



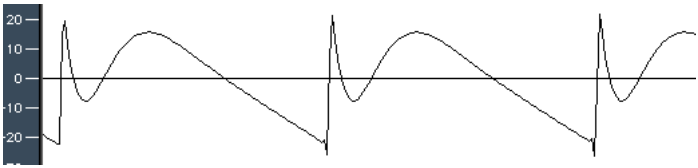
The second partial is not easy to see, even though all other partials are rejected.

Peak Type Filter with Resonance



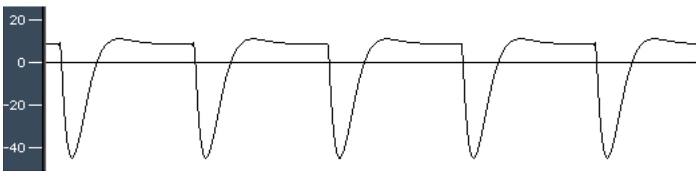
Here, the second partial is strongly emphasized.

Band Rejection



In this image, the second partial is rejected. The basic oscillation and all other harmonics are present. Sawtooth is still the input waveform.

Band Pass Filter



The band pass filter distorts the picture of the sawtooth wave.

Filter 2 FM

The Cutoff Frequency of Filter 2 can be modulated by the sine wave of Oscillator 1, which means that it can be modulated in the audio frequency range.

The effect of such filter modulations in the audio spectrum is unpredictable, but the results tend to remain harmonic as long as the modulation intensity doesn't get too high. **FM** defines the intensity of frequency modulation. This parameter can be modulated in realtime: In the *Router*, this modulation is abbreviated as **LPF FM**.

A clean sine wave, at the frequency of Oscillator 1, is always used as the modulation source.

Don't confuse this type of filter frequency modulation with the FM feature of Oscillator 1, which can be modulated by Oscillator 2, as described in the *Linear Frequency Modulation* section, on page 35.

If a frequency modulation of Oscillator 1 by Oscillator 2 is used, it does not influence the (sine wave) signal used to modulate the cutoff frequencies.

Filter 2 can be driven to self-oscillation. If you set a very high value for **Resonance**, it will produce a sine wave. This self-oscillating sine wave will distort at the maximum **Resonance value**. If you mute all oscillators, you'll only hear this sine oscillation. By modulating the Cutoff Frequency, you can produce effects similar to those produced by modulating the frequency of Oscillator 1 with Oscillator 2.

Handling Processing Power Economically

The ES2 has consequently been designed to make the most efficient use of the computer's processing power. Modules and functions, which are not in use, don't use processing power. This priniciple is maintained by all elements of the ES2. If, for instance, only one of the three oscillators is in use, and the others are muted, less processing power is required. Loso, if you do not modulate Digiwaves, or if you disengage the filters, processing power is being saved. When it comes to filtering, it's not obvious how processing power can be saved most efficiently. Therefore, here are some hints for you:

- If you can achieve the same low-pass-filtered sound with filter 1 as with filter 2, use filter one. Filter 1 uses less processing power, differs a little in sound, but does not sound worse at all.
- **Filter FM** needs additional processing power.
- A modulation of **Filter Blend** needs quite a bit additional processing power, as soon as it is engaged in the router.
- **Drive** requires additional processing power. This especially is the case when it comes filters wired in **series** and **Filter Blend** settings with two distortion circuits. See *Filter Blend and Signal Flow* section, on page 47 for details.

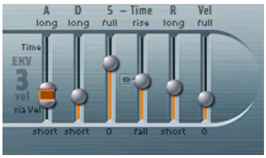
Remember that you'll never be forced to compromises in sound! You always can make use of Logic's bounce feature in order to convert a processor-intensive audio instrument track into an audio track, playing back a "bounced" audio file. To do so, route the audio instrument (the ES2) to an output object. Switch the ES2 track solo. Set the locators in the transport window. Press **Bounce** in the output object (in the audio mixer). Select **Bounce & Add**. After the bounce produre, drag the resulting file from the audio window into the arrange window, onto a stereophonic audio track. Save the ES2 setting. Mute the bounced audio instrument track. Don't delete anything. Save the song. If you want to change the notes, or tempo, recording level or sound, repeat the entire procedure.

3.3 Dynamic Stage (Amplifier)

The dynamic stage defines the level—which means the perceived volume—of the played note. The change in level over time is controlled by an *Envelope Generator*.

ENV 3 and the Dynamic Stage

ENV3 is always hard-wired to the dynamic stage—i. e. Envelope Generator 3 is always used for control over the level of the sound. For detailed explanations of the envelope parameters, see *The Envelopes (ENV1—ENV3)* section, on page 76.



ENV 3 always controls the level.

Router Modulation Target: *Amp*

The dynamic stage can be modulated by any modulation source in the *Router*. The modulation **target** is called **AMP** in the *Router*.

🛈 If you select *AMP* **as the target**, **LFO1** as the **Source**, and leave **via** set to **Off** in the *Router*, the level will change periodically, based on the current **Rate** of the LFO.—i. e. You'll hear a tremolo.

Sine Level

Sine Level allows the mixing of a sine wave (at the frequency of Oscillator 1) directly into the dynamic stage, independent of the filters. Even if you have filtered away the basic partial tone of Oscillator 1 with a high pass filter, you can "reconstitute" it through use of this parameter. Please note:

- In cases where Oscillator 1 is frequency modulated by Oscillator 2 (i. e. if you have turned up **FM** with the waveform selector), *only* the pure sine wave is mixed into the dynamic section, not the distorted FM waveform.

- Low frequency modulations of Oscillator 1's pitch, set in the *Router*, affect the sine wave frequency mixed in here.

Sine Level is well suited for adding warmth and a fat bass quality to the sound. Thin sounds can be "fattened up" with this function, given that Oscillator 1 actually plays the basic pitch.

3.4 The Router

The ES2 features a modulation matrix, called the *Router*. Any modulation **Source** can be connected to any modulation **Target**—much like an old-fashioned telephone exchange or a studio patchbay. The modulation intensity—how strong the **Target** is influenced by the **Source**—is set with the associated vertical slider.

If you hold ⇧ before clicking and moving the vertical slider, the modulation intensity can be fine-tuned.

To set the modulation intensity to zero, just click on the little zero symbol (the small circle) right beside **via**.

The intensity of the modulation can also be modulated: The **via** parameter determines yet another modulation source, which defines the amount or intensity of the modulation.

Ten such modulations of **Source**, **via** and **Target** can take place simultaneously, in addition to those which are hard-wired outside the *Router*.

Some modulations aren't possible, due to technical reasons. For example, the envelope times only can be modulated by parameters, the values of which are present with the note-on message. Therefore, there are situations, where the envelopes are not available as **Targets**. Furthermore, the LFO1 can't modulate its own frequency. Values which are not available are greyed out.

You may have to switch from the display of the Vector Envelope to the display of the Router.

After the **target**, **source** and **via** names you will find brackets containing the abreviations, with which the name will appear in the associated flip menu.

The Range of *Via*

The modulation intensity is set with the vertical slider. This is self-explanatory, as long as the **via** parameter is set to **off.** This ensures that the modulation intensity is constant, if not affected by any other controller (like the modulation wheel or aftertouch).

As soon as you select a value other than **off** for **via**, the slider is divided into two halves. The lower half defines the minimum intensity of the modulation, when the **via** controller is set to its minimum value. The upper half defines the maximum modulation intensity when the **via** modulator (the modulation wheel, in this case) is set to its maximum value. The area between the two slider halves defines the range that is controlled by the **via** controller.

You can grab this area between the two slider halves with the mouse and drag both halves simultaneously. If this area is to small to be grabbed with the mouse, just click in a free part of the slider way and move the mouse up or down to move the area.

In the example, the lower half of the slider knob defines the vibrato intensity when the modulation wheel is turned down, and the upper half defines the vibrato intensity that takes place when the modulation wheel is set to its maximum value.

A Modulation Example

Say you've chosen these settings:

- **Target** = **Pitch 123**
- **via** = **Wheel**

- **Source** = **LFO1**
- Modulation intensity = slider position, set as you like

In this configuration, the modulation **source**—**LFO1**—is used to modulate the frequency (pitch) of all three Oscillators (**Pitch 123**). (**Pitch 123**) is the modulation **target in this example**. You'll hear a vibrato (an undulation of the pitch) at the speed of LFO1's *Rate*. The *modulation intensity* is controlled by the (modulation) **wheel**, which is determined by the **via** parameter. This provides you with control over the depth of vibrato (pitch undulation) via the modulation wheel of your keyboard. This type of configuration is well-known from countless sound settings (patches).

It does not matter which of the 10 *Router Channels* you use.

You can select the same **target** in several *Router Channels*, in parallel. You can freely use the same **sources** as often as you like, and the same **via** controllers can be set in one or multiple *Router* channels.

Modulation Targets

The following *targets* are available for realtime modulation.

ℹ These modulation targets are also available for the X and Y axes of the X/Y modulator (the Square). See *The Square* section, on page 81.

Pitch 123

This **target** allows the parallel modulation of the frequencies (pitch) of all three Oscillators. If you select an LFO as the **Source**, this **target** leads to siren or vibrato sounds. Select one of the Envelope Generators with zero **attack**, short **decay**, zero **sustain** and short **release** as the **Source** for tom tom and kick drum sounds.

Pitch 1

This **target** allows the modulation of the frequency (pitch) of Oscillator 1. Slight envelope modulations can make the amount of detuning change over time, when Oscillator 1 is sounding in unison with another (unmodulated) Oscillator. This is useful for synthesizer brass sounds.

Pitch 2

This **target** allows the modulation of the frequency (pitch) of Oscillator 2.

Pitch 3

This **target** allows the modulation of the frequency (pitch) of Oscillator 3.

Detune

This **target** controls the amount of detuning between all three Oscillators.

Please note, that the sensivity of all the pitch modulation **targets** described above depends on the modulation intensity. This sensivity scaling allows for very delicate vibrati in the cent range (one Cent equals 1/100 semitone), as well as for huge pitch jumps in octave ranges.

- Modulation intensity from 0 to 8: steps are 1.25 cent.
- Modulation intensity from 8 to 20: steps are 3.33 cent.
- Modulation intensity from 20 to 28: steps are 6.25 cent.
- Modulation intensity from 28 to 36: steps are 12.5 Cent.
- Modulation intensity from 36 to 76: steps are 25 Cent.
- Modulation intensity from 76 to 100: steps are 100 Cent.

This leads to the following rules of thumb for modulation intensity settings.

- Modulation intensity of 8 equals a pitch shift of 10 cent.
- Modulation intensity of 20 equals a pitch shift of 50 cent, this is one quarter tone.
- Modulation intensity of 28 equals a pitch shift of 100 cent, this is one semi tone.
- Modulation intensity of 36 equals a pitch shift of 200 cent, this is two semi tone.
- Modulation intensity of 76 equals a pitch shift of 1,200 cent, this is one octave.

- Modulation intensity of 100 equals a pitch shift of 3,600 cent, this is three octaves.

Osc Waves

Dependent on the **Waveforms** set in the three Oscillators, this **target** can be used to modulate:

- the pulse width of rectangular and pulse waves,
- the amount of frequency modulation (Osc1 only),
- Noise color (Osc3 only)
- and the position of the DigiWaves.

ℹ Osc Waves affects all Oscillators together. The targets Osc1 Wave, Osc2 Wave and Osc3 Wave only affect the specified Oscillator. Check out the ensuing paragraphs to see what wave modulation does in the three Oscillators.

ℹ For further informations on the effects of these modulations, please read the *Pulse Width Modulation* section, on page 38. Also take a look at *Linear Frequency Modulation* section, on page 35, *White and Coloured Noise (Oscillator 3 only)* section, on page 42, and the *Digi-Waves* section, on page 34.

Osc1 Wave

Dependent on the **Waveform** selected, you can control the pulse width of rectangular and pulse waves of Oscillator 1, the amount of frequency modulation (with Oscillator 1 being the carrier and Oscillator 2 being the modulator), or the position of the DigiWave. So the pulse width of the rectangular and pulse waves is not restricted to two fixed values in Oscillator 1.

ℹ In classic FM synthesizers, the amount of FM is controlled in realtime by velocity sensitive Envelope Generators. Select one of the ENVs as the **source** for such sounds.

Osc2 Wave

As per Osc1 Wave, except that Oscillator 2 does *not* feature FM. Please note that pulse width modulation also works with the synchronized rectangular wave and with the ring modulated rectangular wave.

Osc3 Wave

Oscillator 3 is as per **Osc1 Wave** and **Osc2 Wave**, but it does not feature FM or ring modulation. Oscillator 3 features *Noise*, the color of which can be modulated with this parameter.

Sine Level (SineLevl)

This **target** allows modulation of the *sine wave* level of Oscillator 1, which can be mixed directly into the input of the dynamic stage—without being affected by the filters. The parameter defines the level of the first partial tone of Oscillator 1. See the *Sine Level* section, on page 58.

Osc Level Scale (OscLScle)

This **target** allows modulation of the levels of *all three* Oscillators simultaneously. A modulation value of **0** mutes all oscillators, while a value of **1** raises the gain of the entire mix by 12dB. The modulation is applied *before* the overdrive stage, allowing for dynamic distortions.

Osc 1 Level (Osc1Levl)

This **target** allows the modulation of Oscillator 1's level.

Osc 2 Level (Osc2Levl)

This **target** allows the modulation of Oscillator 2's level.

Osc 3 Level (Osc3Levl)

This **target** allows the modulation of Oscillator 3's level.

Cutoff1

This **target** allows the modulation of the **Cutoff Frequency** of Filter 1. See the *Cutoff and Resonance* section, on page 50.

Resonance 1 (Reso 1)

This **target** allows the modulation of the **Resonance** of Filter 1. See the *Cutoff and Resonance* section, on page 50.

Cutoff2

This **target** allows the modulation of the **Cutoff Frequency** of Filter 2.

Resonance 2 (Reso2)
This **target** allows the modulation of the **Resonance** of Filter 2.

Lowpass Filter Frequency Modulation (LPF FM)
A sine signal, at the same frequency as Oscillator 1, can modulate the **Cutoff Frequency** of Filter 2 (which always works as a low pass filter). This target allows the modulation of Filter 2's FM modulation intensity. This filter FM parameter is decribed in the *Filter 2 FM* section, on page 56.

Cutoff 1 und 2 (Cut 1+2)
With this target, you can modulate *both* filter's *Cutoff* frequencies in parallel, much like applying the same modulation to Cutoff 1 and Cutoff 2 in two *Router* channels.

Cutoff 1 normal und Cutoff 2 invers (Cut 1inv2)
With this target, you can simultaneously modulate the *Cutoff* frequencies of the first and second filters *inversely* (in opposite directions). Put another way, while the first filter's *Cutoff* frequency is rising, the *Cutoff* of the second filter will fall—and vice versa.

In cases where you have combined Filter 1, defined as a high pass filter, and Filter 2 (which always works in low pass mode) in *Serial* mode, both will serve as a band pass filter. Modulating the **target** **Cut1 inv 2** will result in a modulation of the band pass filter's bandwidth in this scenario.

Filter Blend (FltBlend)
With this target, you can modulate the **FilterBlend** (the crossfading of the two filters), as described in the *Filter Blend and Signal Flow* section, on page 47.

If **FilterBlend** is set as a **target** in one or several *Router* channels, the modulation data for *both* filters will be calculated even if the **FilterBlend** parameter is set to **–1.0** or **+1.0**. As such, we advise caution when choosing **FilterBlend** as a modulation target because it may increase the need for processing power.

Amp

With this **target**, you can modulate the dynamic stage, i. e.—the level (the voice's volume). If you select **Amp** as the **target** and modulate it with an **LFO** as the **Source**, the level will change periodically, and you will hear a tremolo.

Pan

With this **target**, you can modulate the panorama position of the sound in the stereo spectrum. Modulating **Pan** with an **LFO** will result in a stereo tremolo (*auto panning*).

In Unison Mode, the panorama positions of all voices are spread across the entire stereo spectrum. Nevertheless, pan can still be modulated, with positions being moved in parallel.

Scaled Modulations

All following modulation **targets** result in a "*scaled*" modulation, which means that the modulation value isn't simply *added* to the **target** parameter's value, but rather that the **target** parameter value will be *multiplied*. This works as follows: a modulation value of **0.0** results in no effect, while a modulation value of **+1.0** equals a multiplication with 10, and a modulation value of **–1.0** equals a multiplication with 0.04.

LFO1 Rate

With this **target**, you can modulate the frequency (speed, rate) of LFO1.

Say you've created a vibrato with another *Router* channel by modulating the **Target Pitch123** with **LFO1**. If desired, you can have LFO1's speed (the vibrato speed) automatically accelerated or slowed down. To do so, modulate the **target LFO1 Rate** with one of the Envelope Generators (**ENV**). Select **LFO2** as a **source** and reduce it's *Rate* in order to periodically accelerate and slow down the vibrato.

Envelope 2 Decay (Env2Dec)

With this **target**, you can modulate the *Decay* time of the second Envelope Generator.

In cases where you've selected **ENV2 Dec** as the **Target** and **Velocity** as the **Source**, the duration of the decaying note is dependent on how hard you strike the key. Selecting **Keyboard** as the **Source** will result in higher notes decaying more quickly (or slowly).

Envelope 2 All Times (Env2Time)

With this **target**, *all* of *ENV2's* time parameters are modulated: **Attack Time**, **Decay Time**, **Sustain Time** and **Release Time**.

Envelope 3 Decay (Env3Dec)

This **target** modulates the *Decay* time of the third Envelope Generator.

Envelope 3 All Times (Env3Time)

With this **target**, *all* of *Env3's* time parameters are modulated: **Attack Time**, **Decay Time**, **Sustain Time** and **Release Time**.

Glide

With this **target**, you can modulate the duration of the *Glide* (portamento) effect.

If you modulate **Glide** with **Velocity** selected as the **Source**, the velocity (how hard) at which you strike the key will define how long it takes for the played notes to "find their way" to the target pitch. See *Glide* section, on page 27.

Modulation Sources

Some modulation sources are *unipolar*, delivering values between **0** and **1**. Others are *bipolar*, and output values between **–1** and **+1**. The following modulation sources are available:

LFO1

… LFO1 is described in *The LFO's* section, on page 72.

LFO2

… LFO2 is described in *The LFO's* section, on page 72.

ENV1

… Envelope Generator 1 is described in *The Envelopes (ENV1—ENV3)* section, on page 76.

ENV2

… Envelope Generator 2 is described in *The Envelopes (ENV1—ENV3)* section, on page 76.

ENV3

… Envelope Generator 3 is described in *The Envelopes (ENV1—ENV3)* section, on page 76.

Please note , that Envelope Generator 3 always controls the level of the sound.

Pad-X, Pad-Y

This modulation **sources** allows you to define the axes of the Square for use with the selected modulation **target**. The cursor can be moved to any position in the Square, either manually or controlled by the vector envelope. See sections *The Square* section, on page 81 and *The Vector Envelope* section, on page 82.

Max

If you select **Max** as a **source**, the value of this **source** will permanently be set to **+1**. In conjunction with **via** this offers interesting options, as the possible values available for **via** control the modulation intensity.

Keyboard (Kybd)

This outputs the keyboard position (the MIDI note number). The center point is C3 (an output value of **0**). Five octaves below and above, an output value of **–1** or **+1**, respectively is sent.

This could be used to control the **Cutoff Frequencies** of the filters in parallel with the keyboard position—i. e. as you played up and down the keyboard, the *Cutoff Frequencies* would change. Modulate the **target** **Cut 1&2** with the **source** **Keyboard** to do so. At a modulation intensity of **0.5**, the **Cutoff Frequencies** scale proportionally with the pitch played on the keyboard.

Velocity (Velo)

The velocity sensitivity serves as modulation **source**, if *Velocity* is selected here.

Bender

The pitch bender serves as a bipolar modulation **source**, if **Bender is** selected. This is also true when the Oscillators' **Bend Range** parameter is set to **0**.

Modulation Wheel (ModWhl)

The modulation wheel serves as an unipolar modulation source, if **Wheel** is selected.

For most "standard" applications, you'll probably use the wheel as the *via* controller. Traditionally, it can be (and is) used for control over the intensity of periodic LFO modulations. Used here, it can be employed for direct, static modulations, such as controlling the Cutoff frequencies of the filters (**Target** = **Cut 1&2**).

The Least Significant Bit (LSB) Controller for the modulation wheel is recognized correctly, as well.

After Touch (Touch)

… stands for Channel Aftertouch.

If you set the **Target** to **Cut 1&2**, the *Cutoff Frequencies* will rise and fall dependent on how firmly you press a key on your touch-sensitive MIDI keyboard *after* the initial keystrike.

Modulation Wheel und After Touch (Whl+To)

The modulation wheel *and* channel aftertouch serve as modulation source.

Expression (Exp)

When **Expression** is selected, the expression pedal (MIDI Control Change Message #11) will serve as the modulation source.

Breath

The breath controller (MIDI control change message #2) serves as the modulation source.

MIDI Controller 16—19 (Ctrl16—19)

MIDI Control Change Messages 16—19 are also known as "General Purpose Slider 1/2/3/4".

The "Vector Stick" (Joystick) of the Korg Wavestation synthesizer generates Controllers 16 and 17, for example. If you use this instrument as your master keyboard, you can control any two ES2 parameters directly with its Joystick.

In the MIDI specification, for all controllers from 0 to 31, there also is a LSB-Controller defined (32 to 63). This "Least Significant Bit"-controller allows for a resolution of 14 bit instead of 7 bit. The ES2 recognizes these control change messages correctly, for instance the controllers for breath or expression.

Via—Controlling the Modulation Intensity

Some modulation sources are *unipolar*, delivering values between **0** and **1**. Others are *bipolar*, and output values between **–1** and **+1**. The following **sources** may be used to modulate the modulation intensity.

LFO1

The modulation undulates at the speed and waveform of LFO1, which controls the modulation intensity.

LFO2

The modulation undulates at the speed and waveform of LFO2, which controls the modulation intensity.

ENV1

ENV1 controls the modulation intensity.

ENV2

ENV2 controls the modulation intensity.

ENV3

ENV3 (the level envelope) controls the modulation intensity.

Pad-X, Pad-Y

Both axes of the Square (of the Vector Envelope) are available as **via sources** as well, enabling you to control modulation intensities with them.

Keyboard (Kybd)

This outputs the keyboard position (the MIDI note number). The center point is C3 (an output value of 0). Five octaves below and above, an output value of –1 or +1, respectively is sent.

If you select Pitch123 as the **target**, modulate it with the **source** LFO1, and select Keyboard as the **via** value, the vibrato depth will change, dependent on key position. Put another way, the vibrato depth will be different for notes higher or lower than the defined Keyboard position.

Velocity (Velo)

If you select Velocity as the **via** value, the modulation intensity will be velocity sensitive, i. e.—modulation will be more or less intense dependent on how quickly (hard) you strike the key.

Bender

The pitch bender controls the modulation intensity.

Modulation Wheel (ModWhl)

If you select Wheel as the **via** value, the modulation intensity will be controlled by your MIDI keyboard's modulation wheel.

The Least Significant Bit (LSB) Controller for the modulation wheel is recognized correctly, as well.

After Touch (Touch)

If you select Touch as the **via** value, the modulation intensity will be touch sensitive, i. e.—modulation will be more or less intense dependent on how firmly you press the key of your touch-sensitive MIDI keyboard *after* the initial keystrike (Channel Aftertouch, also known as pressure sensitivity).

Modulation Wheel und After Touch (Whl+To)

Both modulation wheel and aftertouch (channel pressure) control the modulation.

Expression (Exp)

If you set the **via** value to **Expression**, the intensity of the modulation is controlled by MIDI control change message #11—the expression pedal.

Breath

The breath controller (MIDI control change message #2) controls the modulation.

MIDI Controller 16—19 (Ctrl16—19)

MIDI Control Change Messages 16—19 are also known as “General Purpose Slider 1/2/3/4”.

ℹ The “Vector Stick” (Joystick) of the Korg Wavestation synthesizer generates Controllers 16 and 17, for example. If you use this instrument as your master keyboard, you can control modulation intensities directly with its Joystick.

ℹ In the MIDI specification, for all controllers from 0 to 31, there also is a LSB-Controller defined (32 to 63). This “Least Significant Bit”-controller allows for a resolution of 14 bit instead of 7 bit. The ES2 recognizes these control change messages correctly, for instance the controllers for breath or expression.

3.5 The LFO’s

LFO is the abbreviated form of *Low Frequency Oscillator*. In an analog synthesizer, LFO’s deliver modulation signals below the audio frequency range—i. e. in the bandwidth that falls between 0.1 and 20Hz, and sometimes as high as 50Hz. LFO’s serve as modulation sources for periodic, cyclic modulation effects. If you slightly modulate the pitch of an audio oscillator at a rate (speed, LFO frequency) of, say, 3—8Hz, you’ll hear a “vibrato”. If you modulate the *Cutoff Frequency* of a low pass filter, you’ll hear a “wah wah” effect, and modulating the dynamic stage results in a “tremolo”.

The ES2 features two LFO’s, the outputs of which are available as **sources** in the *Router*.

- LFO1 is polyphonic, which means that if used for any modulation of multiple voices, they will *not* be phase-locked. Furthermore, LFO1 is key-synced: Each time you hit a key, the LFO1 of this voice is starting from zero. To explain, when used on polyphonic input (a chord played on the keyboard) the modulation is independent for each voice (note). Where the pitch of one voice may rise, the pitch of another voice might fall and the pitch of a third voice may reach its minimum value.

- LFO2 is monophonic, which means, that the pitch of all voices will rise and fall synchronously, if you modulated the **target** Pitch123 with the **source** LFO2, for example.

Both LFO's feature a number of waveforms. LFO1 can fade in or out automatically, without the need to engage a separate Envelope Generator. The LFO parameters are detailed below:

EG (LFO1)

At its center position—which can be accessed by clicking the middle mark—the modulation intensity is static: i. e. it won't be faded in or out at all. At positive values, it is faded *in*. The higher the value, the longer the delay time is. At negative values, it is faded *out*. The lower (onscreen) the slider is positioned, the shorter the fade out time is.

The function is abbreviated as *EG* because the fading in or out is internally performed by an ultra-simple *E*nvelope *G*enerator.

Chaotic and fast modulations of the Oscillator(s) frequencies (**Pitch123**) by **LFO1** with a **delayed Sample&Hold** selected as the **waveform**, a high **Rate**, and short fade out, make the attack phase of the note sound "Rogue-ish"—and quite similar to the attack phase of brass instruments.

Most commonly, this is used for delayed vibrato—many instrumentalists and singers intonate longer notes this way. To set up a delayed vibrato: Place the slider at a position in the upper half (**Delay**) and modulate the **Target** **Pitch123** with the **Source** **LFO1**. Set a slight modulation intensity. Select a **Rate** of about **5Hz** and the **triangular** wave as the **LFO waveform of choice**.

Rate

This parameter defines the frequency or speed of the modulation. The value is displayed in Hertz (Hz) beneath the slider.

Wave

This is where you select the desired LFO waveform. Check out the waveforms while a modulation of **Pitch123** is engaged and running. You should find the symbols quite self-evident.

Triangular Wave

The triangular wave is well suited for vibrato effects.

Sawtooth Wave and Inverted Sawtooth

The sawtooth is well suited for helicopter and "space gun" sounds. Intense modulations of the Oscillator(s) frequencies with a negative (inverse) sawtooth wave leads to "bubbling and boiling, underwater" sounds. Intense sawtooth modulations of lowpass filters (such as Filter2) create rhythmic effects.

Rectangular Waves

The rectangular waves make the LFO periodically switch between two values. The upper rectangular wave switches between a positive value and zero. The lower wave switches between a positive and a negative value set to the same amount above/below zero.

An interesting effect you may wish to try out is achieved by modulating *Pitch123* with a suitable modulation intensity, that leads to an interval of a fifth. Choose the upper rectangular wave to do so.

Sample & Hold

The two lower waveform settings of the LFO's output *random* values. A random value is selected at regular intervals, defined by the LFO rate. The upper waveform delivers exact steps of random. At its lower setting, the random wave is smoothend, resulting in a "fluid" changing of values.

A modulation of **Pitch123** leads to the effect commonly referred to as a "random pitch pattern generator" or "sample and hold". Check out very high notes, at very high rates and high intensities—you'll recognize this well-known effect from hundreds of science fiction movies!

The term "*Sample & Hold"* (abbreviation—*S & H)* refers to the procedure of taking "samples" from a noise signal at regular intervals. The voltage values of these "samples" are then *"held"* until the next *sample* is taken. When converting analog audio signals into digital signals, a similar procedure takes place: Samples of the voltage of the analog audio signal are taken at the rate of the sampling frequency.

LFO2 Rate

The **LFO2 Rate** (frequency) control allows for the free (in the upper half of the slider range) or song-tempo synchronized (in the lower half of the slider range) running of LFO2. The rate is displayed in Hertz or rhythmic values, dependent on whether the song tempo synchronization is engaged or not. Rates range from speeds of 1/64-notes through to a periodic duration of 32 bars. Triolic and punctuated values are also possible. LFO2 is ideally suited for rhythmic effects which retain perfect synchronicity even during tempo changes to the song.

3.6 The Envelopes (ENV1—ENV3)

In addition to the complex Vector Envelope, described in *The Vector Envelope* section, on page 82, the ES2 also features three Envelope Generators per voice. On both the front panel and as a **source** in the *Router*, they are abbreviated as ENV1, ENV2 and ENV3, respectively.

ℹ The roots of the term *Envelope Generator* and its basic functionality are described in the *Envelopes* section, on page 103.

The feature sets of ENV2 and ENV3 are identical. ENV3 defines the changes in level over time for each note played. You can regard ENV3 as being "hard-wired" to the *Router's* **AMP m**odulation target.

The parameters of Envelope Generators 2 and 3 (ENV 2 and ENV 3) are identical—but ENV 3 is always used for control over level.

Unlike many other synthesizers, there is no hard-wired connection between any of the *Envelope Generators* and the **Cutoff Frequencies** of the filters in the ES2. Modulation of the **Cutoff Frequencies** *must* be set separately in the *Router*. In the default setting, this already is the case—in the router channel right under the filter (see graphic).

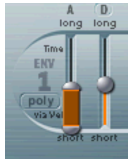
ℹ Set up a ***Router*** channel as follows, in order to establish this type of modulation: Set **Target** to **Cutoff1**, **Cutoff2** or **Cut1&2**, set **Source** to, say, **ENV2**. Once set as described, the slider of the ***Router*** channel will function as the filters **EG Depth** parameter.

ℹ Both **ENV2** and **ENV3** are velocity sensitive, so it is therefore unnecessary to set **via** to **Velo** in the ***Router*** channel: You can leave ***via*** switched **off**.

The Parameters of ENV1

At first glance, ENV1 appears to be rather poorly equipped. Its few parameters, however, are useful for a vast range of synthesizer functions.

Trigger Modes: Poly, Mono, Retrig



In **Poly** mode, the Envelope Generator behaves as you would expect any polyphonic synthesizers' to behave: Every voice has its own envelope.

In **Mono** and **Retrig** modes, a single Envelope Generator modulates all voices in parallel—i. e. identically.

- If ENV1 is set to **Mono**, all notes must be released before the envelope can be triggered again. If you play legato, or any key remains depressed, the envelope won't start its attack phase again.
- In **Retrig** mode, the envelope will be triggered by any key you strike, no matter whether other notes are sustained or not. Every sustained note is affected by the retriggered envelope.

The design of early analog polysynths led to polyphonic instruments where all voices passed through a single low pass filter. This design was primarily due to cost factors. The best known example of these polyphonic instruments were the Moog Polymoog, the Yamaha SK20 and Korg Poly 800. The sole low pass filter of such instruments is controlled by by a single Envelope Generator. To simulate this behaviour, use the modes **Mono** or **Retrigger**.

Say you've modulated the **target Cutoff2** with a percussive **source ENV1**, which is set to **Retrig**. If you play and sustain a bass note, this note will receive a percussive filter effect every time you hit another key. The newly struck key is also shaped by the same filter. Playing a sound set up in this way "feels" like playing a polyphonic synthesizer with *one* filter. This is despite the fact that the ES2's filters remain polyphonic and can be simultaneously modulated by different polyphonic sources.

If you want to simulate the Percussion of a Hammond Organ, you will also need the modes **Mono** or **Retrigger**.

Decay/Release

ENV1 can be set to act as an Envelope Generator with an **Attack Time** and **Decay Time** parameter *or* with an **Attack Time** and **Release Time** parameter.

Switching between both modes is achieved by clicking on the **D** or the **R** above the right slider of ENV1.

- In its **Attack/Decay** mode, the level will fall down to zero after the attack phase is over, no matter whether you sustain the note or not. It will decay at the same speed even if you release the key. The decay time is set with the **Decay Time** slider, abbreviated as **D**.
- In its **Attack/Release** mode, the envelope level remains at its maximum after the attack phase is over, and as long as the key is depressed. Following the release of the key, its level decreases at the speed set with the **R** slider—the abbreviation for **Release Time**.

Attack Time and Attack via Vel

The *Attack Time* slider is divided into two halves. The lower half defines the attack time when the keys are struck hard (at maximum velocity). The upper half defines the attack time at minimum velocity.

You can grab this area between the two slider halves with the mouse and drag both halves simultaneously. If this area is to small to be grabbed with the mouse, just click in a free part of the slider way and move the mouse up or down to move the area.

The Parameters of ENV2 and ENV3

The feature sets of ENV2 and ENV3 are identical, but it is always the task of ENV3 to define the *level* of each note, i. e.—to modulate the dynamic stage. ENV3 is available for simultaneous use as a **Source** in the *Router* as well. The **Decay Time** and (**Sustain) Time** parameters can also be used as modulation targets in the *Router*.

See *Envelopes* section, on page 103 for information on the basic functionality and meaning of Envelope Generators.

Attack Time

As per the *Attack* slider of ENV1, the *Attack Time* sliders of ENV2 and ENV3 are divided into two halves. The lower half defines the attack time when the keys are struck at maximum velocity. The upper half defines the attack time at minimum velocity.

You can grab this area between the two slider halves with the mouse and drag both halves simultaneously. If this area is to small to be grabbed with the mouse, just click in a free part of the slider way and move the mouse up or down to move the area.

Decay Time

The **Decay Time** parameter defines how long it takes for the level of a sustained note to fall to the *Sustain Level*, after the attack phase is over. If **Sustain Level** is set to its **maximum**, the *Decay* parameter has no effect. When the *Sustain Level* is set to its **minimum value**, *Decay* defines the duration or *fade-out-time* of the note.

The **Decay Time** parameter appears as a modulation **Target** in the *Router* for ENV2 and ENV3 individually (**ENV2Dec**, **ENV3Dec**).

On pianos and plucked string instruments, high notes decay faster than low notes. In order to simulate this effect, modulate the **Target Decay Time** with the **Kybd** in the *Router*. The *Router* channel slider should be set to a negative value.

Sustain and Sustain Time

When the **Sustain Time** parameter is set to its center value—which can be achieved by clicking the ∞ symbol—the **Sustain Level** behaves like the *Sustain* parameter of any synthesizer ADSR envelope. Now, the **Sustain Level**, abbreviated as **S**, defines the level that is sustained as long as the key is depressed, and following the completion of the **Attack Time** and **Decay Time** phases.

The **Sustain Time** slider defines the time it takes for the level to rise to its maximum—or to fall to zero—after the decay phase is over. Settings in the lower half of its range (**Fall**) determine the speed at which the level decays from **Sustain Level** to zero. The lower the slider, the faster the decay speed. Settings in the upper half of its range (**Rise**) determine the speed at which the level rises from **Sustain Level** to its maximum. The higher the slider, the faster the rise speed.

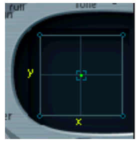
Release Time

As with any synthesizer ADSR envelope, the **Release Time** parameter (**R**) defines the time the level takes to decay to zero after the key is released.

Vel (Velocity Sensitivity)

The **Vel** parameter defines the velocity sensitivity of the entire envelope. If it is set to its **maximum**, the envelope will only output its maximum level when the keys are struck at maximum velocity.

3.7 The Square



The Square has two axes: The X and Y axes have positive and negative value ranges, or, in other words, they are bipolar. By touching and moving the cursor with the mouse, the values of both axes are continuously transmitted. Because you can modulate one freely selectable parameter with the X value, and another freely selectable parameter with the Y value, you can use the mouse like a Joystick.

Alternatively to this realtime control, the position of the cursor is modulated by the *Vector Envelope*—just like the mix between the three Oscillators in the *Triangle*. The *Loop* function of the Vector Envelope generator allows for cyclic movements. This opens a number of doors, allowing it to operate as a two-dimensional, luxurious pseudo-LFO with a programmable waveform. More on this is found in *The Vector Envelope* section, on page 82.

Vector Mode

Vector Mode allows to disable the control of the square cursor by the vector envelope. The same parameter defines, whether the triangle (the oscillator mixer) shall be controlled by the vector envelope.

- **Vector Mode off**—The vector envelope does not influence the triangle nor the square. It's simply switched off. Now you can set and control the cursors of the triangle and the square in real-time.
- **Vector Mode Mix**—The vector envelope controls the triangle (the oscillator mix), but not the square.
- **Vector Mode XY**—The vector envelope controls the square, but not the triangle.
- **Vector Mode Mix+XY**—The vector envelope controls both square and triangle.

Like all of the ES2's parameters, the movements of the cursors in the Triangle and Square can be recorded and automated by Logic. Tip: This automation data can be edited and looped in Logic. This is completely independent of the cyclic modulations of the Vector Envelope. Vector modulation of the Square should be disabled for this type of use (**Vector Mode** = **off**).

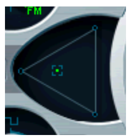
Vector Target—Modulation Destinations

The **Vector X Target** and **Vector Y Target** parameters determine the effect of cursor movements in the Square. The modulation targets are identical to those available in the *Router*, so we won't repeat ourselves here. Please see *You can select the same target in several Router Channels, in parallel. You can freely use the same sources as often as you like, and the same via controllers can be set in one or multiple Router channels.* section, on page 61 for descriptions. The position of the cursor in the *Square* is also available in the *Router*, as the **Source** and **Via** options: **Pad-X** and **Pad-Y.**

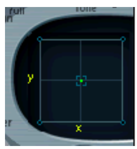
Vector Int—The Modulation Intensity

The maximum intensity, sensitivity *and* the polarity of the modulation is set with **Vector X Int** and **Vector Y Int**.

3.8 The Vector Envelope



The *Triangle* and the *Square* are the most special and unusual elements of the ES2's graphical user interface. Whilst the Triangle controls the mix of the three Oscillators, the X and Y axes of the Square can modulate any (modulation) target.



The *Vector Envelope* can control the movement of the cursors in the Triangle and the Square in realtime. Each voice is equipped with its own Vector Envelope, which is triggered from its start-point with every new keystrike (or with every MIDI note-on message, to be more precise).

The concept of the Vector Envelope, together with the Square and the Triangle, may seem strange at first glance, but believe us: Combine it with the other synthesis possibilities of the ES2, and you end up with sounds that are really unique and literally “moving”.

Envelope Points, Times and Loops

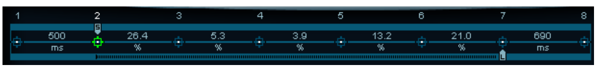
The Vector Envelope consists of up to 15 “points” on the time axis. Each point can control the position of both the Triangle and Square’s cursors.

The points are numbered sequentially. Point 1 is the starting point. In order to edit a point, simply select it—by clicking on it.

Sustain Point

Any point can be declared the *Sustain Point*. Given that the note played is sustained long enough and there’s no loop engaged, any envelope “movement” will stop when this sustain point is reached. It will be sustained until the key is released (until the MIDI note-off command).

To define a point as the *Sustain Point*, click on the turquoise strip above the desired point. The selected point will be indicated by an *S* between the point and its number, on the turquoise strip.



Loop Point

Any point can be declared loop point. Given that the note is sustained long enough, the envelope can be repeated in a loop.

The looped area is the time span between sustain point and loop point. In between, you can define several points which describe the movements of the square- and triangle cursors.

In order to define a point as loop point click on the turquoise strip below the desired point. A loop point is indicated by an *L* in the strip below.

Please note: In order to see and define the loop point, the loop has to be activated. See the *Loop* section, on page 87.

With the loop the Vector envelope works like a multi-dimensional and polyphonic LFO with programmable waveform.

Vector Envelope Times

With the exception of the first point, which is tied to the beginning of each played note, every point has a *time* parameter. This parameter defines the period of time required for the cursor to travel from the point which preceded it. The times are normally displayed in milliseconds (ms).

To adjust a time value, you can click directly on the numerical value and use your mouse as a slider.

Default Setting of the Vector Envelope

The default setting of the Vector Envelope consists of three points. Point 1 is the startpoint, point 2 is defined as the sustain point, and point three is the end point by default.

The impact of the Vector Envelope on the *Oscillator Mix* or on the Square is switched *off* by default. This allows the ES2 to behave as a synthesizer without a Vector Envelope generator. This "traditional" starting point is more convenient when creating patches from scratch.

There are two ways to switch off the Vector Envelope:

- You can switch **on** the **Solo Point** parameter (described on page 86). If it is **on**, only the triangle and square cursors positions of the currently selected point are active.
- Or you can disable the Vector Envelope altogether or only for the triangle or the square, as described in the *Vector Mode* section, on page 81.

Setting and Deleting Points

The more points you set, the more complex the Vector Envelope movements that can be designed. You can:

- Create a new point by ⇧-clicking between two existing points. The segment that previously existed between the two “old” points is divided at the mouse position. The sum of the two new segment times is equal to the time used by the “old” undivided segment. As such, the ensuing points retain their absolute time positions. In addition, the existing cursor positions in the *Triangle* and *Square* are “fixed”, ensuring that the creation of new points does not affect any previously defined movements.
- Delete points by clicking on them while holding ctrl (🍎)/alt (PC).

Setting Vector Envelope Times

By clicking a time value and moving the mouse, you will alter the envelope time—the time it takes for the Vector Envelope to travel from the point before this time value to the point after this time value. You have two ways of doing this.

- Simple vertical dragging of the time parameter results in reaching all following points later (or sooner, respectively) in time.
- Dragging with ctrl (🍎)/alt (PC) held, you will shorten or lengthen the time of the following point by the same amount. This ensures that the adjacent, and all following points, retain their absolute time positions.

Resetting the Values of a Point

How to get the default cursor positions in the Triangle and the Square:

- Clicking in the *Triangle* while holding down the ⌥ (🍎)/ctrl (PC) key sets all Oscillators to output the same level. The cursor is set to the middle of the Triangle.

- Clicking onto the *Square* while holding down the ⌥ ()/ctrl (PC) key sets the cursor to the center of the Square. Both axis values are set to zero.

Solo Point

With this function, you can basically switch off the entire Vector Envelope generator. If **Solo Point** is set to **on**, no dynamic modulations are applied by the Vector Envelope. In this scenario, the currently visible cursor positions of the *Triangle* and *Square* are permanently in effect. These cursor positions match the currently selected Vector Envelope point.

If you select another Vector Envelope point (by clicking on it), you will engage its Triangle and Square cursor positions immediately. If **Solo Point** is set to **on**, the newly selected point will become the solo point.

ℹ You can independently switch off the vector modulation of the *Square* by setting **Vector Mode** **off**, as outlined on page 81.

Envelope Modes Normal and Finish

If **Env Mode** is set to **normal**, the release phase (the phase after the sustain point) will begin as soon as you release the key (note off). The release phase will start from the vector envelope point at which you released the key.

- If the loop is switched off, and the vector envelope reaches the sustain point *S*, the sustain point *S* will be played for as long as you hold the key.
- If the loop is switched on (see *Loop* section, on page 87), and the loop point *L* is before the sustain point *S*, the loop will be played for as long as you hold a key.
- If the loop is switched on, and the loop point *L* is after the sustain point *S*, the loop will be played after the release of a key.

In **Env Mode** **finish**, the vector envelope will not go immediately into the release phase when you release the key. Instead, it will play all its points with all their times until the last point is reached, regardless if you hold the key or release it.

- If the loop is switched off, the sustain point *S* will be ignored. The vector envelope will end on its last point, regardless if you hold the key or release it.
- If the loop is switched on, the vector envelope will play its points until it reaches the loop and then play the loop for as long as the sound ends. It does not matter if the loop point *L* is before or after the sustain point *S*.
- If the loop is switched on, and **Loop Count** is set to other than **infinite**, the vector envelope will go on after the selected number of loop repeats. If **Loop Count** is set to **infinite**, the number of segments after the loop is irrelevant. See *Loop Count* section, on page 89.

Vector Envelope Loops

The Vector Envelope can—like any envelope—run in one shot (as long as the note is sustained). It can also run several times or in an infinite cycle, much like an LFO. You can achieve this through the use of *loops*.

ℹ The loop parameters might remind you of the loop parameters available for samples. Just to avoid any misunderstandings: The Vector Envelope only supplies control signals used for moving the cursor positions of the Triangle and Square. The audio of the ES2 is *not* looped at all.

Loop

The ES2 features these loop options:

Off

If **Loop** is switched **off**, the Vector Envelope runs in “one shot mode” from its beginning to its end—given that the note is held long enough. The other loop parameters are disabled.

Forward

When **Loop** is set to **Forward,** the Vector Envelope runs to the sustain point and begins to repeat the section between loop point and sustain point periodically, and always in a forward direction.

Backward

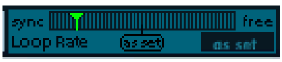
When **Loop** is set to **Backward,** the Vector Envelope runs to the sustain point and begins to repeat the section between sustain point and loop point periodically, always in a backward direction.

Alternate

When **Loop** is set to **Alternate,** the Vector Envelope runs to the sustain point and returns to the loop point and back to the sustain point periodically, alternating in both a backward and forward direction.

Loop Rate

Just as every LFO has its speed (or *Rate*) parameter, the loop can be set to cycle at a defined **Loop Rate**. And just like an LFO, the Vector Envelope **Loop Rate** can be synchronized to the song tempo automatically.



- If you switch the **Loop Rate** to **as set**, the duration of the loop cycle is equal to the sum of the times between the sustain and loop points. Click on the field labeled **as set** (below the **Rate** slider) to select.
- If you set the **Loop Rate** to one of the rhythmic values (**sync**, left half of the slider, **32 bars** up to **64th Triplet Note**), the **Loop Rate** fits to the song tempo.
- You also can set the **Loop Rate** in the small panel to the right half of the slider (**free**). The value indicates the number of cycles per second. Use the mouse as a slider to adjust.

If **Loop Rate** is not switched to **as set**, and the loop is activated (**Loop Mode Forward**, **Backward** or **Alternate**), the times of the points between the loop and sustain points as well as the value for **Loop Smooth** are indicated as a percentage of the loop duration, rather than in milliseconds.

Loop Smooth

When **Loop Mode** is set to **Forward** or **Backward**, there will ineveitably be a moment when a transition from the sustain point to the loop point has to be performed. In order to avoid abrupt cursor position changes, this transition can be smoothed through use of the **Loop Smooth** parameter.

- Set the value by touching and dragging it with the mouse.
- If **Loop Rate** is set to **Sync** or **Free**, the Loop Smoothing Time will be displayed as a percentage of the loop cycle duration.

- If **Loop Rate** is set to **as set**, the Loop Smoothing Time will be displayed in milliseconds (ms).

Loop Count

The loop cycle of the Vector Envelope doesn't have to be performed infinitely—you can have it cycle just a few times. Following the defined number of repetitions, the Vector Envelope will run from the sustain point onwards, just as with **Loop Mode off**.

- Use the mouse as a slider to set the **Loop Count value**. Possible values are **1** to **10** and **infinite**.

Time Scaling

You can stretch and compress the entire Vector Envelope. As an example, to double the Vector Envelope speed, it's not necessary to halve the time values of every point. All you need to do is set **Time Scaling** to **50%**.

- Adjust **Time Scaling** by using the mouse as a slider.

- The range for the **Time Scaling** parameter is from 10% to 1000%. It is scaled logarithmically.
- If the **Loop Rate** is **as set**, the scaling also affects the loop. If not (**Loop Rate** = **free** or **sync**), the setting will not be affected by **Time Scaling**.

Fix Timing—Normalizing Time Scaling and Loop Rate

By clicking **Fix Timing**, the **Time Scaling** value will be multiplied by all time parameters, and **Time Scaling** will be reset to **100%**. There will be no audible difference. This is simply a normalizing procedure, most similar to the normalizing function of the sequence playback parameters in Logic.

In cases where a loop which was synchronized to the song tempo had been engaged (**Loop Rate** = **sync**), pressing **Fix Timing** will also switch the **Loop Rate** to **as set**, thus preserving the absolute rate.

3.9 Effect Processor

The ES2 is equipped with an integrated effect processor. Any changes to this processor's effects settings are saved as an integral part of each sound program. The entire output of the dynamic stage is processed in "true stereo".

Despite the inclusion of this integrated effects processor, please feel free to process the ES2 with every other effects plug-in included with your Logic version. The sound and parameter set of the integrated effects unit is reminiscent of classic pedal effects, designed for the electric guitar. The use of guitar pedal effects on classic analog synthesizers was a standard practice amongst gigging musicians.

Distortion

At its **Soft** setting, the distortion circuit is somewhat like a tube overdrive, whilst **Hard** sounds like a fully-transistorized fuzz box. The **Distortion** control defines the amount of distortion, and **Tone** controls the treble portion of the output of the distortion process.

Chorus, Phaser, Flanger

These classic modulation effects and their parameters (**Intensity** and **Speed**) won't need any explanation. These sophisticated algorithms simulate the sound of analog effects of this kind, with one exception: They don't produce as much noise.

A ***Chorus*** effect is based on a delay line, the output of which is mixed with the original, dry signal. The short delay time is modulated periodically, resulting in pitch deviations occur. The modulated deviations, in conjunction with the original signal's pitch, produce the chorus effect.

The ***Flanger*** works in a fashion similar to that of a chorus, but with even shorter delay times, and the output signal being fed back into the input of the delay line. This feedback results in the creation of harmonic resonances which "wander" cyclically through the spectrum, giving the signal a metallic sound.

The ***Phaser*** is based on a mix of a delayed and an original signal. The delayed element is derived from a so-called all-pass filter, which applies a frequency-dependent delay to the signal. This is expresssed as a phase angle. The effect is based on a "comb filter", which is basically an array of inharmonic notches (rather than resonances, as with the flanger), which also "wander" through the frequency spectrum.

3.10 Random Sound Variations

The ES2 offers a unique feature which allows you to vary the sound parameters randomly. You can define the amount of random variation, and can restrict the variations to specific sonic elements. The random sound variation feature will inspire, aid (and occasionally amuse) you when creating new sounds.

Controls of the random sound variation feature

By pressing the **RND** button, the sound is altered randomly. The process is triggered by a single click and can be repeated as often as you like.

ℹ To avoid possible misunderstandings: This feature has nothing to do with random realtime modulations. The random feature changes the parameters randomly with each mouse click. Realtime random modulations are performed with the random waveforms of the LFO's and, for random pitch settings, by the ***Analog*** parameter.

RND Intensity

RND Int defines the amount of random parameter alteration. As you turn the slider to the right, you will increase the amount of random variation.

The random sound variation feature always alters the parameters as they are currently set, not based on the memorized *Setting* file. As such, clicking **RND** repeatedly will result in a sound which increasingly differs from the original setting. If want to check out several slight alterations of the current setting, you can reload the original *Setting*, after each random alteration.

RND Destination

Some aspects of your sound may already be ideal for the sound you had in mind. As such, it may not be desirable to alter them. Say your sound setting has a nice percussiveness, and you'd like to try a few sonic color variations while retaining this precussive "feel". To avoid the random variation of any attack times, you can restrict the variation to the Oscillator and Filter parameters, with the Envelope parameters excluded from the variation process. To do so, set the **RND Destination** to **Wav&Filt** (**Wave & Filter**).

Please note:

- The **Master Level**, **Filter Bypass** as well as the three **Oscillator On/Off** parameters and the display option **Vector/Router** won't ever be randomized.
- In case of randomizations of the vector envelope, **Solo Point** will always be set to **off**.

You can restrict the random sound variation to the parameter groups listed below:

All
All ES2 parameters, with the exception of the parameters mentioned above will be altered.

All without Router and Pitch (All w/o R+P)
All ES2 parameters, with the exception of all Router parameters and the basic pitch (**semitone** settings of the oscillators) will be altered. The oscillator fine tuning will vary nevertheless. This will supply more musically useful sounds.

All without Vector (All w/o Vec)
All ES2 parameters, with the exception of all Vector Envelope parameters will be altered. This maintains the rhythmic feel of a given setting.

Waves
Only the **Wave** and **DigiWave** parameters of the Oscillators will be altered. The other Oscillator parameters (tuning, mix and modulations in the *Router*) are excluded.

DigiWaves

Digiwaves will be selected in all oscillators. The **DigiWave** number of the Oscillators will be altered. The other Oscillator parameters (tuning, mix and modulations in the *Router*) are excluded.

Filters

The filter parameters are varied. The parameters included are: **Filter Structure** (series or parallel wiring), **Filter Blend**, **Filter Mode**, **Cutoff Frequency** and **Resonance** for Filter 1 and 2, **Fatness, Filter FM** of Filter 2.

ENVs

All envelope parameters of all three envelopes ENV1, ENV2 and ENV3 are varied. The Vector Envelope is *excluded*.

LFOs

All LFO parameters of all LFOs are varied.

Router

All Router parameters of all router channels are varied with all intensities, **Target**, **via** and **Source** parameters.

FX

All effects parameters are varied.

Vector

All Vector Envelope parameters are varied, including the XY routing of the square.

VectorMix

The oscillator mix levels (triangle cursor positions) of the Vector Envelope points are altered. The rhythm and tempo of the modulation (the time parameters of the points) will not be altered.

VectorXY

The square cursor positions of the Vector Envelope points are altered. The XY routing won't be altered. The rhythm and tempo of the modulation (the time parameters of the points) will not be altered.

VecTimes
Only the time parameters of the Vector Envelope points will be altered.

VecStruct
The Vector Envelope structure will be altered: All times, the sustain point, the number of points and all loop parameters.

It's recommended that you save any good sounds resulting from the *RND* process, as you work. Do this under a new name (**Setting** > **Save Setting as...**) in the plug-in window.

Chapter 4
Synthesizer Basics

Just in case you're not that familiar with synthesizers …

… we've included a few paragraphs to let you know what the ES2 is all about, how the sound generation process works, and what its special features are.

4.1 Analog and Subtractive

An analog synthesizer signal is an electrical signal, measured in volts. To give you a brief comparison with a technology you're probably familiar with, we'll look at speakers. The speaker "coils" move when the voltage—amplified by a power amplifier and output to the speaker—changes. When the voltage rises, the speaker coil moves forward. If the voltage falls, the speaker coil moves backwards.

In a digital synthesizer, the signal flow is digital. Binary *descriptions* of the signal (a string of zeros and ones …) are fed from one algorithm to another. This is an important distinction to make. It is *not* the signal itself that is fed from a vitual oscillator to a virtual filter a. s. o. A virtual analog synthesizer is a digital synthesizer which mimics the architecture, features and peculiarities of an analog synthesizer. It includes the "front panel" with all controls, which provides direct access to all sound generation parameters.

The ES2's virtual signal flow is as per that found in analog synthesizers. It includes some of the "desirable" idiosyncracies of particular analog circuits—i. e. in cases where they tend to sound nice, such as high oscillator levels overdriving the filter. The ES2 also features a graphical control surface on your computer screen. Its signal processing (those "virtual" oscillators etc.) is performed by the central processing unit (CPU) of your computer.

Undesirable analog synthesizer phenomena, such as the habit of going completely out of tune, are not simulated by the ES2. You can, however, set the voices of the ES2 to randomly detune, adding “life” to the synthesizer’s sound. Unlike its analog counterparts, the ES2 is also completely programmable (you can save sound settings), can be completely automated (you can record and playback fader movements), polyphonic (you can play up to 32 notes at the same time), multitimbral (you can play different sounds at the same time), and velocity sensitive.

These are important benefits, which overcome the limitations of old synthesizers. If you find it more inspirational to avoid the use of these features, you can always switch them off.

4.2 What Is Synthesis?

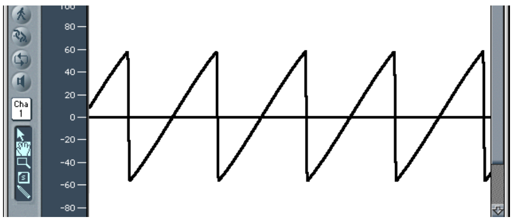
Before we start, “synthesis” in this context, is the (re)production of a sound which emulates, or “synthesizes” the sound of another instrument, a voice, helicopter, car, dog bark… in fact, any sound you can think of!

This “synthetic” reproduction of other sounds is what gives the synthesizer its name. Needless to say, synthesizers can also produce many sounds which would never occur in the “natural” world. This ability to generate sounds which cannot be created in any other way is what makes the synthesizer a unique musical tool. Its impact on modern music has been enormous, and will continue to be well into the future—although it is more likely to live on in “virtual” form, rather than as hardware.

4.3 Subtractive Synthesis

Subtractive synthesis is synthesis using filters. All analog and virtual analog synthesizers use subtractive synthesis to generate sound. In analog synthesizers, the audio signal of each voice is generated by the *Oscillator*. The Oscillator generates an alternating current, using a selection of waveforms which contain differing amounts of (more or fewer) harmonics. The fundamental (or root) frequency of the signal primarily determines the perceived pitch, ts waveform is responsible for the basic sound color, and the amplitude (level) determines the perceived volume.

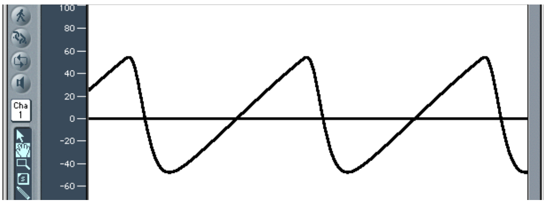
Cutoff, Resonance and Drive—illustrated with a sawtooth wave



This picture shows an overview of a sawtooth wave (a = 220Hz); The filter is open, with Cutoff set to its maximum, and with no resonance applied. The screenshot shows the output signal of an Emagic ES 1, routed to a monophonic Logic output object (object 1). The recording was performed with the Bounce function of this audio object, and is displayed in Logic's Sample Editor at a high zoom setting.

When Michelangelo was asked how he would manage to cut a lion out of a block of stone, he answered, "I just cut away everything that doesn't look like a lion". This, in essence, is how subtractive synthesis works: Just filter (cut away) those components of sound which should not sound—i. e. you subtract parts of the Oscillator signal's spectrum. After being filtered, a brilliant sounding sawtooth wave becomes a smooth, warm sound without sharp treble. Analog and virtual analog synthesizers are not the only devices that make use of subtractive synthesis techniques. Samplers and sample players also do so, but use modules which play back digital recordings (Samples) in place of Oscillators, which supply sawtooth and other waveforms.

The picture below shows a sawtooth wave with the filter half closed (24dB/Fat). The effect of the filter is somewhat like a graphic equalizer with a fader set to a given cutoff frequency (the highest frequency being fed through), pulled all the way down (full rejection), so that the highs are damped. With this setting, the edges of the sawtooth wave are rounded, making it resemble a sine wave.

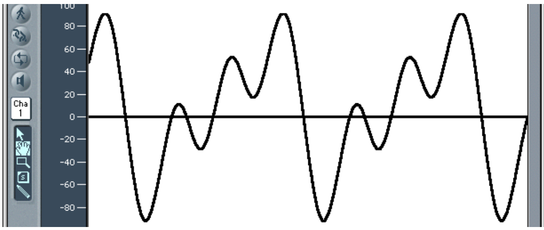


The wave length here is not really higher, but the zoom setting is higher.

Fourier Theorem and Harmonics

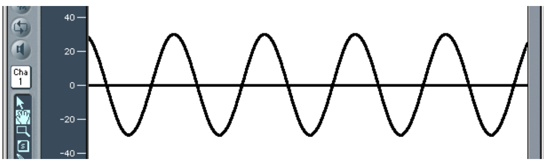
"Every periodic wave can be seen as the sum of sine waves with certain wave lengths and amplitudes, the wave lengths of which have harmonic relations (ratios of small numbers)". This is known as the Fourier theorem. Roughly translated into more musical terms, this means that any tone with a certain pitch can be regarded as a mix of sine partial tones. This is comprised of the basic fundamental tone and its harmonics (overtones). The basic oscillation (the first partial tone) is an "A" at 220Hz. The second partial has double the frequency (440Hz), the third one oscillates three times as fast (660Hz), the next ones 4 and 5 times as fast, and so on.

You can emphasize the partials around the cutoff frequency using high values for **resonance**. The picture below shows a sawtooth wave with a high resonance setting, and the cutoff frequency set to the frequency of the third partial (660Hz). This tone sounds a duodecima (an octave and a fifth) higher than the basic tone. It's apparent that exactly three cycles of the strongly emphasized overtone fit into one cycle of the basic wave:



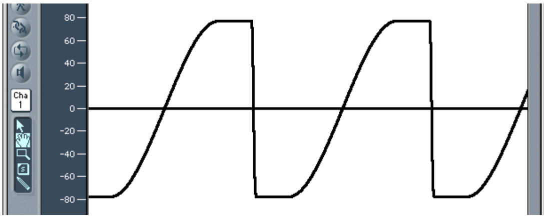
The effect of the resonating filter is comparable to a graphic equalizer with all faders higher than 660Hz pulled all the way down, but with only 660Hz (**Cutoff Frequency**) pushed to its maximum position (**resonance**). The faders for frequencies below 660Hz remain in the middle (0dB).

If you switch off the Oscillator signal, a maximum resonance setting results in the self-oscillation of the filter. It will then generate a sine wave.



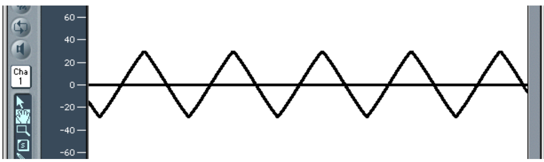
Drive

The ES2's **Drive** parameter offers a further opportunity to shape the waveform, and to change the sound. The picture shows an unfiltered sawtooth wave, with *Drive* set to about 40 %. You can see how the wave touches the floor and ceiling of the filter's dynamic range, thus overdriving it and distorting the waveform.



Other Oscillator Waveforms

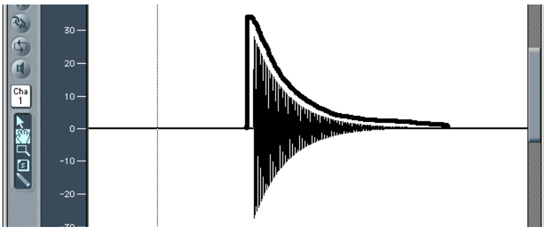
Waveforms (waves) are named "sawtooth", "square", "pulse" or "triangular" because of their shape when displayed as an oscillogram (as in Logic's Sample Editor). Select the "Tutorial Saw" sound setting and listen to different waveform settings. This is the triangular wave:



The triangular wave has few harmonics—which is evident by the fact that is shaped more like a sine than a sawtooth wave. This wave contains only odd harmonics—which means no octaves.

Envelopes

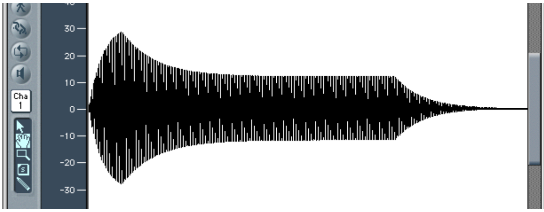
What does the term “envelope” mean in this context? In the image, you see an oscillogram of a percussive tone. It’s easy to see how the level rises immediately the top of its range, and how it decays. If you drew a line surrounding the upper half of the oscillogram, you could call it “the envelope” of the sound—a graphic diplaying the level as a function of time. It’s the job of the Envelope Generator to set the shape of the envelope.



The screenshot shows a recording of an ES1 sound created with this setting of the ADSR parameters (**attack time, decay time, sustain level** and **release time**): **attack** as short as possible, medium value vor **decay**, zero for **sustain**, medium value for **release**.

In order to achieve this envelope, the **Sustain Time** parameter must remain centered (indicated by the ∞ Symbol). The Envelope Generator responsible for the volume (level) of the ES2 is ENV3.

When you strike a key, the envelope travels from zero to maximum level in the **attack time**, falls from this maximum level to the **sustain level** in the **decay time**, and maintains the **sustain level** as long as you hold the key. When the key is released, the envelope falls from its **sustain level** to zero in the **release time**. The brass or string-like envelope of the following sound—the envelope itself is not shown in this graphic—has longer **attack** and **release times** and a higher **sustain level**.



The Envelope Generator can also control the rise and fall of the Cutoff Frequency. You can also use Envelope Generators to modulate other parameters. For these modulation purposes, the ES2 offers three Envelope Generators. In the *Router*, the mod envelope can be assigned to control several modulation destinations. In this context, modulation can be thought of as a remote control for a given parameter. There are more sources that can serve as a modulation source: e. g. the pitch (note number), the velocity sensitivity, or the modulation wheel. The Envelope Generator parameters are discussed in *The Envelopes (ENV1—ENV3)* section, on page 76.

Chapter 5

Tutorials

You will find the settings for these tutorials under the names of the sound developers in the settings menu (in the head of the ES2 plug-in window).

5.1 Soundworkshop: Emagic ES2

By Peter Krischker

Tutorial Setting: “Analog Saw Init”

Topics: Sound Design from Scratch, Filter Settings, DigiWaves

This is designed for use as a starting point when programming new sounds from scratch. Professional sound designers like to use such “scratch” settings when programming entirely new sounds, usually as follows: An un-filtered sawtooth wave sound without envelopes, modulations or any gimmicks. This type of setting is also well-suited to the purpose of getting to know a new synthesizer. It allows you to access all parameters without having to consider any pre-set values.

- Start with the filters, the heart of any subtractive synthesizer. Check out the 4 low pass filter types 12dB, 18dB, 24dB and **fat** (Filter 2) with different values for **Cut** (**Cutoff Frequency**) and **Res** (**Resonance**). Define Envelope 2 as filter envelope. This modulation wiring is pre-set in the router.
- Set **Filter Blend** to its left-most position, which will allow you to listen to Filter 1 in isolation. In many circumstances, you’ll probably prefer to use Filter 2, but Filter 1 has its advantages. In addition to the low pass filter with 12dB/octaves

slope (**Lo**), Filter 2 also offers: a high pass, peak, band pass (**BP**) and band rejection mode (**BR**). Filter 1's low pass sounds "softer" in comparison to Filter 2. It is best-suited to sounds where the filter effect is/should be less audible (Strings, FM-Sounds). Distorted TB-303 style sounds are more easily achieved with Filter 1.

- This setting is also ideal for checking out the oscillator waveforms. The analog waveforms can be set in the editor view. In order to select the DigiWaves, set **Osc 1 Wave** to **DigiWave**.

Tutorial Setting: "Analog Saw 3Osc"

Topics: Three detuned sawtooth oscillators, unison mode

Fat synthesizer sounds have always been popular, and are likely to remain so, given their use in modern trance and techno styles. This setting features three detuned oscillators, and sounds "fat" as it is. The following will introduce you to some additional tools to fatten the sound even more.

- Check out the 3-oscillator basic sound with different filter and envelope settings.
- Check out the chorus effect at different **intensities** and **speeds**.
- Engage **unison** mode and select a higher setting for **Analog**. As the sound is polyphonic, each note is doubled. The number of notes that can be played simultaneously will be reduced from 10 to 5. This will make the sound rich and broad. Combining **Unison** and higher values for **Analog** will spread the sound across stereo spectrum.

In many factory settings, the **Unison** mode is active. This demands a lot of processing power. If your computer isn't fast enough, you can switch off the **Unison** mode and insert an Ensemble effect in a bus, for use with several plug-ins. This will save lots of processing power. Another way to save CPU resources is to bounce several Audio Instrument tracks—which place high demands on the processor—to an audio track.

Tutorial Setting: "Analog Unison"

Topics: Extremely detuned monophonic analog sounds, effects

There's nothing fatter than this heavily detuned, un-filtered basic sound. As with the example above, three sawtooth oscillators are used, but they are detuned further. The combination of **Unison** and **Analog** (set to a high value) is essential again, but this time, monophonic mode is used to "stack" 10 voices. Without further effects, the result is an extremely fat lead sound, as used in countless dance and trance productions. With appropriate filter and envelope settings, trendy arpeggio and sequencer sounds can easily be set up.

- Set the **Cutoff Frequency** of Filter 2 to **0**. This will activate the preset filter envelope. Feel free to check out different envelope settings.
- Switch Osc 1 to sound one or two octaves lower.
- Increase **Drive** or **Distortion**.
- Set **Envelope 2** to be velocity sensitive. This allows for velocity sensitive filter modulations.
- Insert a stereo delay effect in the audio instrument channel strip of the ES2. In order to delay several Audio Instruments, you might prefer to insert the effect into a *Bus*, which is accessed via each channel's *Send*.

Logic incorporates reverb and delay effects which are essential for many synthesizer sounds. These aren't integrated into the ES2, ensuring that no processing power is unnecessarily wasted.

Tutorial Setting: "Analog Bass clean"

Topic: Clean bass settings with one oscillator only

Not every sound needs to comprise of several oscillators. There are numerous simple and effective sounds which make use of a single oscillator. This is especially true of synthesizer bass sounds, which can be created very quickly and easily through use of this basic setting.

The basic sound is a rectangular wave, transposed down by one octave. The sound is filtered by **Filter 2**. What's special about this sound is its combination of **legato** and **Glide** (portamento). As long as you play staccato, no glide effect will occur. If you play legato, the pitch will smoothly glide from one note to another. All keys must be released before striking a new key, in order to retrigger the envelopes.

- Check out different filter and envelope settings.
- Replace the rectangular wave with a sawtooth.
- Vary the **Glide** settings.

Editing works best while a bass line is running. Create a monophonic sequence, with most notes played staccato and some legato. This can provide some interesting results with very long **Glide** values.

Tutorial Setting: "Analog Bass distorted"

Topic: Distorted Analog Basses

Filter 1 is engaged, with high settings for **Drive** and **Distortion**. This filter is better suited to the creation of distorted analog sounds than Filter 2.

- Check out Filter 2 by setting **Filter Blend** to its right-most position. You'll hear that Filter 1 works better with distorted sounds.
- In order to control the filter modulation, move the green sliders of the first modulation channel in the router. This controls the modulation intensity.

Tutorial Setting: "FM Start"

Topic: FM Intensity and Frequency

These are the first steps to take when learning about linear Frequency Modulation (FM) synthesis. You'll first hear an unmodulated sine sound, generated by Oscillator 1. Oscillator 2 is switched on and set to produce a sine oscillation as well, but its level is set to **0**: Just push the cursor in the triangle in the uppermost corner.

In the ES2, Oscillator 1 is always the carrier, and Oscillator 2, the modulator. So Oscillator 2 modulates Oscillator 1.

- Adjust the intensity of the frequency modulation by slowly moving the **wave selector** from **Sine** to **FM**. You will hear a typical FM spectrum, with the carrier and modulator set to the same frequency.
- Alter the modulator frequency (Oscillator 2) by adjusting **Fine Tune** from **0 c** to **50 c**. You'll hear a very slow frequency modulation that can be compared to the effect of an LFO. The frequency modulation, however, takes place in the audio spectrum. It is adjusted in semitone steps by the frequency selector. Check out the entire range from **–36 s** to **+36 s** for Oscillator 2. You'll hear a broad spectrum of FM sounds. Some settings will remind you of classic FM synthesizer sounds.
- Select other waveforms for Oscillator 2. **Sine** is the classic, standard FM waveform, but other waveforms lead to interesting results as well, especially the **DigiWaves**.

- You will achieve further interesting results by altering the carrier (Oscillator 1) frequency. Check out the entire range: from –36 s to +36 s semitones here, as well. The odd intervals are especially fascinating. Note that the basic pitch changes when doing so.

Tutorial Setting: "FM Envelope"

Topic: Controling FM Intensity by an envelope and FM scaling

In this setting, you can control the FM intensity with an Envelope, generated by Envelope 2. The modulation target is the range which falls between **Sine** and **FM** in the Oscillator **wave** selector. The first *Router* channel is used for this. You can control a wider range through the use of additional modulations, which have been pre-prepared for you. All you need to do is set their values. As these modulations work without velocity sensitivity, you can set them in the editor view by moving both the lower and upper fader halves to their top-most positions.

- Set the second modulation channel to **1.0**. You'll hear how the modulation will now "wander" through a broader sound range.
- Set modulation channels 3 and 4 to a value of **1.0** as well, and listen to the increase in the sound range.
- Following such drastic augmentations to the modulation range, the sound will have become uneven. In the lower and middle ranges, it sounds nice, but in the treble range the FM intensity appears to be too severe. You can compensate for this effect by setting the target **Osc 1 Wave** to be modulated by the keyboard position **(kybd)** in modulation channels **5** and **6**. This results in a keyboard scaling of the FM intensity.
- As the sound range is so vast (due to the 4 modulations), two modulation channels are required to compensate for this. Set the lower slider halves to their lowest positions. Good keyboard scaling is essential for any FM sound.

Tutorial Setting: “FM Drive”

Topic: FM with Drive and Filter-FM

You can dramatically alter the character of FM sounds by applying **Drive** and **Filter FM**. The results are reminiscent of the feedback circuits of classic FM synthesizers.

- Check out different **Drive** and **Filter FM** settings.
- Lower the **Cutoff Frequency** of Filter 2 to **0**. Envelope 2 modulates Filter 2. This modulation routing is preset in the setting.

Tutorial Setting: “FM DigiWave”

Topic: FM with Digiwaves

In this example, a DigiWave is used as an FM modulator. This results in a bell-like spectra out of only two operators. Normally, this type of timbre could only be produced through the use of a larger number of sine oscillators.

In order to create a fatter, undulating and atmospheric quality to the sound, the polyphonic **unison** mode has been engaged. Filter and amplitude envelopes have been preset to shape the sound.

- Check out the variety of DigiWaves, as FM modulation sources.
- Check out different **Analog** parameter values.

Tutorial Setting: "FM Wavetable"

Topic: FM with Wavetables

You can program the most vivid FM sounds when the modulation source "morphs" between different Digiwaves. The morphing in this setting is controlled by **LFO 2** in this setting. The tempo of LFO 2 (and therefore the morph) depends on the sequencer tempo (here: **2 bars**).

- Set LFO 2 to different waveforms. **Lag S/H** (smooth random), in particular, should be fun.
- Check out different FM intensities and Oscillator frequencies.
- Alter the modulation intensity of the first modulation channel (**LFO2** modulates **Osc2 Wave**) and the LFO 2 **rate**.

Tutorial Setting: "FM Megafat"

Topic: Distorted FM in monophonic Unison

This sound is hard core, and is well-suited for distorted basses and guitar-like sounds. In its treble range, this sound gets rather "rude". This cannot be compensated for by scaling, but not every sound has to be "nice" over the entire keyboard range!

- Check out extreme detunings by adjusting the **Analog** parameter.
- Check out the **Flanger** with this sound.
- Engage the filter envelope by lowering the **Cutoff Frequency** of Filter 2 down to **0**.
- Add some **Glide** to lead sounds.
- As always, when it comes to FM: You can dramatically alter the sound by varying the frequencies of the oscillators. Make sure you check out the odd intervals, as well.

Tutorial Setting: “FM Out of Tune” and “FM Tuned”

Topic: FM with unusual spectra

If you’re unconcerned with the pitch of your sound, you can get the weirdest spectra out of odd frequency ratios (oscillator intervals).

This setting offers a bell-like sound, reminiscent of a ring modulator. It was achieved through a setting of **30 s 0 c**, with the modulator set to a value of **0 s 0 c**. Sounds like this were very commonly used in the electronic music of the eighties, and have undergone a resurgence in popularity in modern ambient and trance music styles.

You can further develop the sound by applying filtering, envelope modulations and effects. There is, however, one little problem—the sound is out of tune.

- Use Oscillator 3 as a reference for the tuning of the FM sound, by moving the cursor in the Triangle.
- You’ll notice that the sound is 5 semitones too high (or 7 semitones too low, respectively).
- Transpose both oscillators 1 and 2 five semitones (500 ct) lower. Transposing them upwards is not practical, as you’d have to select **37 s 0 c** for Oscillator 1, which maxes out at **36 s 0 c**.
- It’s important to maintain the frequency ratio (interval) between Oscillators 1 and 2. This means that Oscillator 1 will sound at **25 s 0 c** and Oscillator 2 at **–5 s 0 c**.

Tutorial Settings: “PWM Start”, “PWM Slow”, “PWM Fast” and “PWM Scaled”

Topic: Slow and fast pulse width modulations with Oscillator 2

Pulse Width Modulation (PWM) is one of the most essential features of any sophisticated analog synthesizer. Use this setting to manually control the pulse width of a rectangular wave, set via the **Wave** control

Avoid Drive and Distortion with PWM sounds.

- Select the “PWM Start” setting, and move the **Wave** control slowly back and forth between the rectangular and the pulse **wave** symbols. Both are green. What you will hear is a (manual) pulse width modulation.
- Select the “PWM Slow” setting. Here, LFO 1 controls the pulse width modulation source, not your manual movements. The result should be quite similar.
- Raise the LFO 1 rate from its pre-set value of **0.230** to **4.400**. The result is a classic, fast PWM.
- In this, and the next step, the PWM shall be set so that it sounds slower in the lower keyboard range, and faster in the upper range. This is desirable for many sounds, such as synthetic strings. First, reduce the **LFO 1 Rate** to **3,800**.
- Change the modulation intensity of the second router channel (**target** = **LFO1 Rate**, **Source** = **Kybd**) to **0.46**. This will alter the scaling of the PWM, making it sound faster in the treble range. You can also hear this type of effect in the “PWM Scaled” setting.

Tutorial Settings: "PWM 2 Osc" and "PWM Soft Strings"

Topics: Pulse width modulation with two oscillators, PWM Strings

In order to make the sound fatter, add Oscillator 3, which can also be modulated in pulse width. In fact, even the first oscillator can deliver PWM. In the "PWM 2 Osc" setting, both oscillators are detuned in a relatively strong way. Develop your own personalized PWM string sound, using this setting as your "base".

- Adjust the **Chorus intensity**. You'll probably choose higher values which make the sound rather broad.
- Program Envelope 3 according to your taste. You should, at the very least, raise the **attack** and **release** times. Define it to react to **velocity**, if you prefer. If you do not intend to solely use the sound as a simple pad, a shorter **Decay Time** and a lower **Sustain Level** of about **80** to **90%** may be more appropriate.
- Reduce the **Cutoff Frequency** and **Resonance** of Filter 1 to make the sound softer.
- Save the new setting.
- Compare the result with the "PWM 2 Osc" setting. You'll hear that the sound has undergone a remarkable evolution.
- Compare it to "PWM Soft Strings", which was created as described above. You'll probably notice a few similarities.

Tutorial Setting: “Ringmod Start”

Topic: Ring Modulation

A ring modulator takes its two input signals and outputs the sum and difference frequencies of them.

In the ES2, Oscillator 2 outputs a ring modulator, which is fed with a square wave of Oscillator 2 and the wave of Oscillator 1, when **Ring** is set as Oscillator 2’s waveform.

Odd intervals (frequency ratios) between the oscillators, in particular, result in bell-like spectra, much like those heard in the “RingMod Start” setting.

As discussed in the “FM Out of Tune” section, on page 113, the third oscillator can be used as a tuning reference, in order to maintain a kind of basic tuning. On occasion, you may find that it is nice to leave the sound out of tune, for use as a source of overtones and harmonics for another basic wave, supplied by Oscillator 3.

Try to program an atmospheric bell sound, using your own imagination. Some hints:

- Experiment with the various frequency ratios of Oscillators 1 and 2. You may want to use the **29 s 0 c/21 s 0 c** ratio, which doesn’t sound out of tune at all. Ring modulation is not only useful for bell-like sounds, It’s also good for a great variety of spectra which tend to sound pretty “weird” at lower frequency settings. Also try alterations to the fine tuning of the Oscillators.
- Check out an **Intensity** of **50%** and a **Rate**, set to around **2/3** of the maximum value, for the Chorus effect.
- Set the **Attack** and **Release Times** of Envelope 3 to taste.
- Check out **Drive** and **Filter FM**, if you like your sounds a little “out of control”.
- The rest is up to you!

Tutorial Setting: “Sync Start”

Topic: Oscillator Synchronisation

If you select the synced square and sawtooth waveforms for Oscillators 2 and 3, they will be synchronized with Oscillator 1. In the “Sync Start” setting, only Oscillator 2 is audible and Oscillator 3 is switched off.

“Typical” sync sounds feature dynamic frequency sweeps over wide frequency ranges. These frequency modulations (the “sweeps”) can be applied in various ways.

- Try the pre-programmed pitch modulation, assigned to the modulation wheel, first.
- In the second router channel, an envelope pitch modulation has been pre-programmed (**target** = **Pitch 2**, **Source** = **Env 1**). Setting the minimum value to **1.0** results in a typical sync envelope. Also check out shorter **Decay Times** for Envelope 1.
- In order to avoid a “sterile”, and lifeless sound after the decay phase of the envelope, you may want to modulate the oscillator frequency with an LFO as well. Use the third router channel and set the minimum modulation applied by **LFO 1** to about **0.50**.
- Check out the synchronized square wave in place of the synced sawtooth wave.

Pulse width modulation is also available via the synchronized square wave of Oscillators 2 and 3. A modulation of the wave parameters of these two Oscillators results in a PWM if the synced square is selected.

Tutorial Setting: “Vector Start” and “Vector Envelope”

Topic: First steps in Vector Synthesis

In this tutorial section you’ll find some useful hints for the programming of vector envelopes. In the “Vector Start” setting, the mix of the Oscillators is controlled by the vector envelope. Each Oscillator has been set to a different waveform.

- Switch from the **Router** view to **Vector** view.
- In its basic (default) setting, the vector envelope has 3 envelope points. Point 1 is the start point, Point 2 the sustain point, and Point 3 is the target in the release phase. By clicking the points, you can see that the mix is always set to **100%** for Oscillator 1, in the Triangle.
- Click Point 2, and move the Triangle cursor to Oscillator 2. You’ll hear a square wave, instead of Oscillator 1’s sawtooth.
- Engage the Vector Envelope by switching the **Solo Point** parameter **off**. As long as it is switched **on**, you will only hear the selected point, with no dynamic modulation. Having switched **Solo Point** **off**, you’ll hear the sound moving from saw to square, with every triggered note.
- Alter the pre-set time of **498 ms**, between points 1 and 2.
- While holding ⇧, click between points 1 and 2. This will create a new Point 2, and the point formerly known as “Point 2” will become Point 3. The total time span between Point 1 and Point 3 is divided into the times between Points 1 and 2, and 2 and 3. The division takes place at the click location. If you clicked at the exact mid-point, the new time spans are equal.
- Grab the newly created Point 2, and move its cursor in the Triangle to Oscillator 2.
- Grab Point 3, and move its cursor in the Triangle to Oscillator 3. Listen to the three oscillators morphing from

sawtooth to square to a triangular wave at the final sustain point.

- Grab Point 4 (the end point) and move its cursor in the Triangle to Oscillator 1 (if it is not already there). Listen to how the sound returns to Oscillator 1's sawtooth wave, following the release of the key.

Tutorial Settings: “Vector Envelope” and “Vector XY”

Topic: Vector Synthesis—XY Pad

This example starts where the first one left off. You have a simple Vector Envelope consisting of 4 points, which is set to modulate the oscillator mix (the Triangle).

In this example, the Vector Envelope will be used to control two additional parameters: The **Cutoff Frequency** of Filter 2 and **Panorama**. These are pre-set as the “**X**” and “**Y**” **Targets** in the Square. Both have a value of **0.50**.

- Switch **on Solo Point**, in order to more easily listen to the settings for the single points.
- Click Point 1. You will only hear Oscillator 1's sawtooth.
- Move the cursor in the Square to the hard left, which results in a low **Cutoff Frequency** for Oscillator 2.
- Click Point 2. You will only hear Oscillator 2's rectangular wave.
- Move the cursor in the Square all the way down, which results in the right-most **Panorama** position.
- Click Point 3. You will only hear Oscillator 3's triangular wave.
- Move the cursor in the Square all the way up, which results in the left-most **Panorama** position.

- Switch **on Solo Point**. The sound begins with a strongly filtered sawtooth wave and turns into an-unfiltered square wave. It initially sounds from the right, and then moves to the left while morphing into a triangular wave. After releasing the key, the saw sound will be heard.

This example sound isn't very dramatic or interesting, but we wanted it to be as clear as possible for the purposes of the tutorial.

Tutorial Settings: "Vector Loop"

Topic: Vector Synthesis Loops

This example is much more spectacular. The basic sound, without the Vector Envelope, consists of three elements:

- Oscillator 1 delivers a metallic FM spectrum, modulated by Oscillator 2's wavetable.
- Oscillator 2 outputs cross-faded DigiWaves (a wavetable), modulated by LFO 2.
- Oscillator 3 plays a PWM sound at the well-balanced, and keyboard-scaled, speed of LFO 1.

Unison and **Analog** make the sound fat and wide.

These heterogenic sound colors shall be used as sound sources for the vector loop.

A slow **forward** loop is pre-set. It moves from Oscillator 3 (PWM sound, Point 1) to Oscillator 1 (FM sound, Point 2), then to Oscillator 3 again (PWM, Point 3), then to Oscillator 2 (Wavetable, Point 4) and finally, it returns to Oscillator 3 (PWM, Point 5). Points 1 and 5 are identical, avoiding any transition from Point 5 to Point 1 in the **forward** loop. This "transition" could be smoothed out with *Loop Smooth*, but this would make the rhythmic design more difficult to program.

The distances between the points of the Vector Envelope have been set to be rhythmically exact. Given that **Loop Rate** has been engaged, the time values are not displayed in ms, but as percentages. There are four time values (each at 25%), which is a good basis for the transformation into note values.

- Switch off the Vector Envelope by setting **Solo Point** to **on**. This allows you to audition the individual points in isolation.
- Take the opportunity to alter the cursor positions in the Square according to your taste. As in the example above, the **X/Y** axes of the Square control the **Cutoff Frequency** of Filter 2, and the **Panorama** position. Adjustments to these make the sound more vivid.
- Activate the Vector Envelope by setting **Solo Point** to **off**. Check the result, and fine-tune the cursor positions in the Square.
- Alter the **Loop Rate** from the pre-set value of **0.09** up to **2.00**. You will hear a periodic modulation, much like that of an LFO. At this point, the modulation is not synchronized with the song tempo. To synchronize the loop speed with the song tempo, move the **Rate** cursor to the very left, and set a note or bar value.
- You can create faster rhythmic note values by clicking between two points and setting the new time values (resulting from the division which occurs) to, say, **12.5%**.

Tutorial Setting: “Vector Kick”

Topic: Bass Drum with self-oscillating filter and Vector Envelope

Electronic kick drum sounds are quite commonly created with modulated self-oscillating filters. This approach can also be taken with the ES2, particularly when the Vector Envelope is used for filter modulation. An advantage of the Vector Envelope, in comparison with conventional ADSR envelopes, is its ability to define/provide two independent decay phases. The distortion effect applies the right amount of “drive” without losing the original sonic character of the drum sound.

In order to make this setting really “punchy”, you must make sure to activate **Flt Reset**. Because all Oscillators are switched off in this setting, the filter needs some time to start oscillating. At the start of each note, **Flt Reset** sends a very short impulse into the filter to make it oscillating right from the start.

Through “tweaks” to the “Vector Kick” setting, you’ll probably be able to create any dancefloor kick drum sound your heart desires. These are the parameters which allow for the most efficient and significant variations:

- Filter2 slopes **12dB**, **18dB**, **24dB**,
- **Distortion Intensity**, **Soft/Hard**,
- Envelope 3’s **Decay Time** (**D**),
- **Vector Envelope Time 1 > 2** (Pre-set: **9.0ms**),
- **Vector Envelope Time 2 > 3** (Pre-set: **303ms**),
- **Vector Time Scaling**.

Tutorial Settings: "Vector Perc Synth" and "Vector Punch Bass"

Topic: Percussive Synthesizers and basses with two filter decay phases (Vector Envelope)

As with the "Vector Kick", this setting uses the Vector Envelope to control the Filter Cutoff Frequency (with two independently adjustable decay phases). This would not be possible with a conventional ADSR envelope generator. Try creating further percussive synthesizers and basses by varying these parameters:

- **Vector Envelope Time 1 > 2** (= Decay 1),
- **Vector Envelope Time 2 > 3** (= Decay 2),
- **Vector Time Scaling**,
- Cursor positions in the Square for points 1, 2 and 3 (= **Cutoff Frequency**),
- Choosing other waveforms.

5.2 Templates for Emagic ES2

by Hubertus Maaß

Welcome to my brief programming tour of the ES2!

While working on the factory preset programming for the ES2, a number of beta testers, sound programmers and other people involved in the project, indicated that it would be nice to start their programming work from templates, rather than entirely from scratch.

Needless to say, creating templates which covered all sound genres is something of a mission impossible. As you spend time familiarizing yourself with the ES2's architecture, you'll start to understand why …

Nevertheless, we kept this basic goal in mind, and included this programming tour for the ES2 as a part of the "toolbox" to help you learn and understand the ES2's architecture through experimentation. You'll find that this approach is fun. You'll also discover, as you're working through a number of simple operations, that results will come quickly when starting to create your personal sound library.

As you become more familiar with the ES2, and what its myriad of functions and parameters do, you can create your own templates for use as starting points when designing new sounds.

I wish you a great deal of fun as you begin to discover the potential of the ES2, and hope that you get more out of the templates than I could have imagined when preparing them for you.

Hubertus Maaß

-1- Clean Stratocaster ("Slap Strat")

This template was my first contribution to the ES2 forum. You may find that it offers a lot of "tweaking" potential.

My target was the sound of a Stratocaster, with the switch between bridge and middle pickup in the middle position (in phase). I tried to emulate the noisy "twang", typical of this sound's characteristics.

This might be a useful template to start work on emulations of fretted instruments, harpsichords, clavinets and so on.

Let me describe its architecture:

Osc 1 and 3 provide the basic wave combination within the *DigiWave* field. Changing the *DigiWave*s of both (in combination) delivers a huge number of basic variations—some also work pretty well for electric piano-type keyboard sounds.

Osc 2 adds harmonics with its synced waveform, so you should only vary its pitch or sync waveform. I kept the "noisiness" I was after in mind. There are a couple of values which can be changed here, which will give you a much stronger, more balanced signal.

I used an old trick (learned while working on wavetable synthesisers) which delivers a punchy attack that the use of a "naked" wave wouldn't deliver—even with the best and fastest filters available: You use an envelope (in this case No. 1) for a quick "push" of a wavetable's window (or all wavetables together, where it makes sense).

So I set up Envelope 1's *Decay* time for this short "push", moving the wave selectors for all Oscillators on the attack. (… actually it makes no sense on the synced sawtooth Oscillator, No. 2, but what an effort to exclude it from the selection!—it just works this way …)

So you can vary the "punchiness" of the content between:

1. envelope 1's contribution to overall attack noise, changing decay speed (a slow one gives you a peak, a long one gives you a growl, as it is reading a couple of waves from the wavetable).

2. modulation destination: you can always assign this to each of the Oscillators separately.

3. start point (you vary the wave window start with minimum/maximum control of EG1/Osc.waves modulation: negative values for a startwave before the selected wave, positive starts from a position behind the selected wave and rolls the table back) …

4. Feel free to try out a couple of experiments with this wavetable-driving trick. The growl effect works pretty well for brass sounds, and some organs absolutely shine with a little "click", courtesy of a wavetable "push".

Envelope 2, which controls the filter, provides a slight attack used for the slapped characteristics. Setting it to the fastest value eliminates the wah-like attack (and don't worry, there's enough "punch" left).

For playing purposes, you'll find that LFO 2 is used as a real-time source for vibrato. It is assigned to the mod wheel and pressure. Personally, I like to add a litle filter opening for all pitch modulations, introduced by the vibrato depth.

Don't concern yourself too much with the different settings for wheel and pressure—this is only how I felt most comfortable bringing in the controls. Feel free to change them!

I've set up velocity to be very responsive because I don't touch keys in the manner of a piano player's weighted-action-punch. As such, I ask that you play this patch softly, or you may find that the "slap" tends to sweep a little. Alternately, you can adjust the sensitivity of the filter modulation's velocity value to match your personal touch.

Also feel free to increase the *Voices* to maximum; I thought six strings would be enough for a guitar, but for held or sustained notes, a few extra voices may come in handy.

-2- The big twirl, basically ("Wheelrocker")

This quite ordinary organ patch doesn't hold any deep, high-end sound design secrets: it is just a combination of three Oscillators with their wave levels mixed together. I'm positive that you'll easily find a different combination which more closely matches your vision of what an organ sound is like. Check out the *DigiWaves*.

I'd like you to focus your attention on the mod wheel's response: please hold a chord, and bring the wheel in by moving it slowly upwards, until you reach the top (maximum).

What I intended to program with this mod wheel modulation was a simulation of an accelerating rotor speaker (or "Leslie").

My modulation routings do the following jobs:

- Modulation 1 (Cutoff 1) assigns envelope 2 to Filter 1 (the only one I use for this patch), and produces a little organ key "click" with the envelope. I also opened the filter (with

Keyboard as via) when playing the higher ranges of the keyboard (with the maximum value).

- Modulation 2 and 3 (Pitch 2/Pitch 3) bring in LFO 1 vibrato, and both Oscillators are modulated out of phase.
- Modulation 5 reduces the overall volume—according to my personal taste, the organ's level shouldn't increase too drastically when all modulations are moved to their respective maximums.
- Modulations 6 and 7 (Pitch 2/Pitch 3) detune Oscillators 2 and 3 against each other, within symmetrical values (to avoid the sound getting out of tune, overall). Again, both work out of phase with modulations 2 and 3; Oscillator 1 remains at a stable pitch.
- Modulation 8 brings in LFO1 as a modulator for panorama movement—this patch changes from mono to stereo. If you would prefer a full stereo sound, with a slowly rotating Leslie in its idle position, just set an amount equal to the desired minimum value, thereby achieving a permanent, slow rotation. Another modification you may wish to try is a higher value, resulting in more extreme channel separation.
- Modulation 9 speeds up LFO 2's modulation frequency.
- Modulation 10: for increasing the intensity of the "big twirl" I added a little Cutoff to Filter 1.

Please feel free to find your own values for customization of this setup. While doing so, keep in mind the fact that there are two modulation "couples", which should only be changed symmetrically (Mod. 2 and 3 work as a pair of "twins", and also Mod. 6 and 7). So, if you change Pitch 2's maximum to a lower minus value, remember to set Pitch 3's maximum value to the same positive amount (same goes for modulation pair 6 and 7).

You can also bring in LFO 2 to increase the pitch diffusion against LFO 1's pitch and pan movements. Just exchange it for LFO 1 on modulation 2 and 3—but note that there will be no modulation source for the Leslie acceleration—so you'll have to use it in a static way, just fading it in. Alternately, you'll have to sacrifice one of the other modulations in favor of a second twirl.

For another stereo modification of the idle sound, you can use the patch in *Unison* mode with a slight detune (please adjust the "analog" parameter for this).

-3- Something Horny ("Crescendo Brass")

First of all—the tasks of the Oscillators:

- Oscillator 1 provides the basic brass wave—"sawtooth".
- Oscillator 2 provides a—not that brassy—"pulse" wave, which brings in the "ensemble". It is pulse-width modulated by LFO 1 (Modulation 4).

Please note that for any modifications, the following *critical* point should be taken into account. There are four (4) parameters, which behave in an entirely different fashion, once one of them is changed. As such, all four must be changed when making adjustments:

1. You may adjust the initial pulse width of Oscillator 2's wave parameter. I selected a sort of "fat" position, close to the ideal square wave because I wanted to program a full, voluminous synth-brass sound.
2. Modulation 4 adjusts the modulation intensity, which means: how far does the range differ from "fat" to "narrow", when being pulsewidth modulated? Set with the *Minimum* parameter.
3. The rate of LFO1 directly controls the speed of the movement of the pulse width modulation. For this patch, I used both LFO's, to achieve a stronger diffusion effect at different modulation speeds. As a bit of general advice, I suggest that you use LFO1 for all permanent, automatic modulations because you are able to delay its "job" with its EG parameter. You may use LFO 2 for all realtime modula-

tions, which you intend to access via modwheel, pressure or other controls while playing.

4. Additionally, I set up a keyboard assignment as the Modulation 4 source because all pitch or pulse-width modulations tend to cause a stronger detuning in the lower ranges, while the middle and upper keyzones feature the desired diffusion effect. When using this parameter, you should initially adjust the lower ranges until an acceptable amount of detuning resulting from the modulation is reached. Once set, check whether or not the modulations in the upper zones work to your satisfaction. Adjust the relationship between intensity (Max) and scaling (Min) values.

Oscillator 3 generates a *DigiWave*, which I considered "brassy" enough within the overall wave mix. As an alternative to the *DigiWave*, I could have used another modulated pulse wave to support the ensemble, or another sawtooth wave to achieve more "fatness" when detuning it with Oscillator 1's sawtooth wave.

What I was after, however, was to have a little bit of growl, achieved through a short wavetable "push", as described for the Stratocaster patch, on page 124. This configuration is set up in Modulation 3 (Oscillator 3 Wave moved by Envelope 1's Decay).

Other controls

Envelope 1 also effects the pitch of Oscillator 2 against Oscillator 3. This results in the clash of both pitches against each other and also against the stable pitch of Oscillator 1, in the attack phase of the sound.

The filter envelope's design closes with a short stab in the attack phase and then opens again for a slower crescendo phase.

I've assigned another realtime crescendo to the mod wheel, which also brings in an overall pitch modulation, controlled by LFO 2.

In addition to all of this, I programmed a sort of "contradictory" realtime modulation (by pressure) which closes the filters. This allows you to play with an additional decrescendo, remotely controlled by touch. Try to get a "feel" for the patch's response. You'll find that it offers quite a few controls for "expression": velocity, pressure after note on, and pressure in advance. Listen to what happens when pressing with the left hand before hitting a new chord with the right hand, and allowing the swell come in.

-4- … making our job redundant …? ("MW-Pad-Creator 3")

I wanted to create a patch which is able to create patches by itself—actually I'm still working on the question of how I can get it to save its own results…

The Basics

Again, I use Oscillator 2 for a pulse width modulation to create a strong ensemble component (please refer to *-3- Something Horny ("Crescendo Brass")* section, on page 128, for further information).

Oscillators 1 and 3 are set to an initial start wave combination within their respective *DigiWave* tables. You can modify these and start with a different combination of *DigiWaves* from the outset.

On Modulation 3, I assigned a wavetable "drive" to all three Oscillators via the mod wheel. What this actually means is that you scroll through Oscillator 1 and 3's wavetables, and change Oscillator 2's pulse width by moving the mod wheel.

Try a careful, very slow movement of the modwheel, and you'll hear drastic changes within the wave configuration. Each incremental position of the wheel offers a different complete digital pad sound. No frantic movements, please, otherwise this will sound like an AM-radio.

Another potential modification procedure is hidden in the modulation intensity of Oscillator 1, 2 and 3 wave's parameter. As already mentioned with the Stratocaster patch, the value of this intensity parameter assigns step width and direction through the wavetables. You may try modifications to the amount and positive or negative values.

I also discovered an interesting side-effect of FM assignment to Filter 2 (Modulation 4/Lowpass Filter FM). Moving to higher positions of the mod wheel, I increased the frequency modulation on the filter, causing all cyclical "beats" (vibrating pitches, detunes, pulsewidth) to be emphasized. This also results in a rough and hissing touch to the overall sound character.

FM offers vast scope for experimentation, and you can decide between:

- An initial FM, using Filter 2's FM parameter, which you can "redraw" (set a negative modulation amount for Modulation 4's maximum) by moving the mod wheel to its top position.
- Or you may have permanent FM (and another modulation setup, saved for a different assignment). You can also switch off FM, if you consider its effect too "dirty" sounding.

Realtime control is via pressure for a vibrato (Modulation 10), and also for a slight opening of the Cutoff to emphasize the modulation (Modulation 9).

-5- Another approach to "Crybaby" ("Wheelsyncer")

Never obsolete—and undergoing a renaissance in new popular electronic music: Sync Sounds

The technical aspects of forcing an Oscillator to sync are described in the *Sync (Oscillator Synchronization)* section, from page 39 onwards. Here's the practical side of the playground.

Wheelsyncer is a single-oscillator lead sound, all others are switched off.

Although Oscillator 2 is the only one actively making any sound, it is directly dependent on Oscillator 1.

If you change Oscillator 1's pitch or tuning, the overall pitch of the sound will go out of tune, or will be transposed.

The pitch of Oscillator 2 provides the tone-colour (or the harmonics) for the sync sound. Pitch changes are controlled by modulation 7's setup, where I assigned Oscillator 2's pitch to the mod wheel.

If you move the wheel, you can scroll through the spectrum of harmonics that I've programmed for realtime changes. Any modification here starts with the pitch of Oscillator 2 itself, which I've set to three semitones below the overall pitch. Feel free to start with a different pitch for Oscillator 2; it won't effect the patch's tuning.

The next modification may be modulation 7's intensity (or the interval). I've selected the maximum value—maybe this is too extreme for your needs, so feel free to reduce it.

Another modification lies in the tone colour of the lead sound itself. I've switched Oscillator 1 off, as I'm already satisfied with the result. If you switch it on, it will offer you the entire range of Oscillator 1's waveforms; from *DigiWaves*, through the standard synth stuff, up to a sine wave, which can be further modulated by FM.

All realtime controls are via the mod wheel: It is used for opening the filter on modulation 6, a panning movement on modulation 8, and acceleration of panning movement on modulation 9. If you have deeper modulation interests, please refer to *-2- The big twirl, basically ("Wheelrocker")* section, on page 126, where I used a similar setup for the Leslie simulation.

Chapter 6
A Brief History of the Synthesizer

by Jan-Hinnerk Helms

6.1 Precursors to the Synthesizer

As the synthesizer evolved, crafty designers produced a vast number of wildly different instruments. In this summary, we're limited to showcasing just a few of the most significant developments in synthesizer history.

It may surprise you to learn that it all began in the twilight years of the 19th century: In 1896/1897, an American inventor named Thaddeus Cahill applied for a patent to protect the principle behind an instrument known as the *Telharmonium* or *Dynamophone*. Weighing in at a staggering 200 tons, this mammoth electronic instrument was driven by twelve steam-powered electromagnetic generators. This behemoth was played in realtime using velocity-sensitive keys and, amazingly, was able to generate several different sounds simultaneously. The Telharmonium was presented to the public at large in a series of "concerts" held in 1906. Christened "Telharmony", this music was piped into the public telephone network because no public address systems were available at the time.

In 1919, Russian inventor Leon Theremin took a markedly different approach. Named after the man who masterminded it, the monophonic *Theremin* was played without actually touching the instrument. It gauged the proximity of the player's hands waved in an electrostatic field between two antennae, and used this information to generate sound. This unorthodox technique made the Theremin enormously difficult to play. Its eerie, spine-tingling (but almost unvarying) timbre made it a favorite on countless horror flick soundtracks. Incidentally, R. A. Moog, whose synthesizers would later garner worldwide fame, began to build Theremins at the tender age of 19.

In Europe, the Frenchman Maurice Martenot devised the monophonic *Ondes Martenot* in 1928. The sound generation method of this instrument was akin to that of the Theremin but, in its earliest incarnation, it was played by pulling a wire back and forth.

In Berlin during the '30s, Friedrich Trautwein and Oskar Sala worked on the *Trautonium*, an instrument that was played by pressing a steel wire onto a bar. Dependent on the player's preference, it enabled infinitely variable pitches, much like a fretless stringed instrument, or incremental pitches similar to that of a keyboard instrument. Sala continued to develop the instrument throughout his life, an effort culminating in the two-voice *Mixturtrautonium* in 1952. He scored numerous industrial films as well as the entire soundtrack of Alfred Hitchcock's masterpiece "The Birds" with this instrument. Although the movie does not feature a conventional musical soundtrack, all birdcalls and the sound of beating wings heard in the movie were generated on the Mixturtrautonium.

In Canada, Hugh Le Caine began to develop his *Electronic Sackbut* in 1945. The design of this monophonic instrument resembled that of a synthesizer, but it featured an enormously expressive keyboard, which responded not only to key velocity and pressure, but also to lateral motion.

The instruments discussed thus far were all designed to be played in realtime. Relatively early on however, people began to develop instruments that combined electronic sound generators and sequencers. The first instrument of this kind was presented by the French duo Edouard Coupleux and Joseph Givelet in 1929—the inspirationally named *Automatically Operating Musical Instrument of the Electric Oscillation Type.* This hybrid married electronic sound generation to a mechanically punched tape control. A mouthful in anyone's book, its unofficial name was shortened to *Coupleux-Givelet Synthesizer* by its builders; this was, incidentally, the first time a musical instrument was called a "synthesizer".

The term was officially introduced in 1956 with the debut of the *RCA Electronic Music Synthesizer Mark I*, developed by American engineers Harry F. Olson and Herbert Belar. Its dual-voice sound generation system consisted of twelve tuning forks which were stimulated electro-magnetically. For its time, the instrument offered relatively sophisticated signal processing options. The output signal of the sound generator could be monitored by loudspeakers and, amazingly, recorded directly onto two records! A single motor powered both turntables and the control unit of the Mark 1. The synthesizer was controlled by information punched onto a roll of paper tape, which actually enabled continuous automation of pitch, volume, timbre and envelopes. It was as complicated as it sounds—handling was anything but a dream, and spontaneous playing was impossible.

6.2 The First Voltage-controlled Synthesizers

With the exception of the Telharmonium, which was conceived prior to the advent of the thermionic valve, these precursors to the modern-day synthesizer were all based on tube circuitry. This made these instruments relatively unwieldy and certainly volatile. Once the transistor saw the light of day in 1947/48, more rugged, smaller, and thus portable, instruments were soon to come.

At the end of 1963, American innovator R. A. Moog met the composer Herbert Deutsch, who inspired Moog to combine a voltage-controlled oscillator and amplifier module with a keyboard in 1964—the first prototype of a voltage-controlled synthesizer. This collaboration with Deutsch prompted Moog to extend his range of modules and combine modules to create entire systems. It wasn't until 1967, that Moog actually called his diverse mix-and-match systems "synthesizers".

Moog's achievements spread by word of mouth, and Moog, always keen to elicit the feedback of his customers, continued to add further modules to his offering. A milestone contributing immeasurably to the breakthrough of Moog's instruments was Wendy Carlos' LP release "Switched-On Bach", in 1968. The record featured Moog's modular synthesizers and multitrack recording. The recording's success introduced the synthesizer to a wider audience and made the name Moog synonymous with the synthesizer. Hoping to capitalize on the new sounds of this instrument and match Carlos' commercial success, numerous studios, producers and musicians acquired Moog modular synthesizers. In 1969, as many as 42 employees produced two to three complete modular systems every week at Moog's production facility.

Working independently, an engineer named Donald Buchla had conceived and implemented the concept for a modular, voltage-controlled synthesizer. This coincided with Moog's version. Buchla also developed his first instruments in close co-operation with users. The inspiration for his first synthesizer originated with composers Morton Subotnik and Ramon Sender, of the San Francisco Tape Music Center. Although he began working on this instrument in 1963, it didn't make its public debut until 1966. By design, Buchla's instruments catered primarily to academia and avant-garde musicians, so they never garnered the public attention and acclaim of Moog's instruments.

6.3 Compact and Cheap

These early voltage-controlled synthesizers were modular: One (or several) chassis housed the power supply and the actual modules. The inputs and outputs of the modules had to be interconnected via a confusing tangle of patch cords before the synthesizer would do so much as actually make a sound. Establishing these connections properly was an art unto itself, and dialing in useful settings on the modules required a good deal of expertise.

Moog realized that these modular synthesizers were too complex for the average musician and were likely to bomb if sold on the traditional music market. More significantly though, they were too expensive. In 1969, in collaboration with engineers Jim Scott, Bill Hemsath and Chad Hunt, Moog began designing a compact, portable and affordable synthesizer that was easy to use. After building three prototypes, the *Minimoog Model D* was finally presented to the public in the summer of 1970.

Unlike previous modular synthesizers, it was neither necessary nor possible for players to connect the diverse modules of the Minimoog as they saw fit. All of the modules' connecting circuitry was hard-wired at the factory. The type and number of modules was also fixed. This simplified manufacturing considerably, and cut costs dramatically. Hard on the heels of a major marketing campaign, the Minimoog became a huge success. Without altering its functional design, some 13,000 Minimoogs were sold worldwide, right up to 1981.

6.4 Storability and Polyphony

Customers weren't entirely satisfied, however. Although musicians no longer had to contend with countless cords in order to play a synthesizer, they still had to deal with loads of knobs and switches before they could do something as "basic" as switch from one sound to another. Moreover, keyboardists were bored with playing monophonic melody lines on synthesizers—they wanted to be able to play chords. Although dual-voice keyboards that connected to two monophonic synthesizers were available as early as 1970, customers wanted more.

Attempting to satisfy these demands, two schools of thought emerged in synthesizer design: One approach called for an independent, monophonic synthesizer to be assigned to every key on the keyboard. To this end, designers married the design principles of electronic organs to synthesizer technology. Although this breed of synth was indeed fully polyphonic—all notes of the keyboard could be heard simultaneously—it wasn't nearly as versatile in its control options as a "pure" synthesizer. The first fully polyphonic synthesizer to feature this type of design was the *Moog Polymoog*, released in 1975. Developed primarily by David Luce, it featured a keyboard with 71 weighted, velocity-sensitive keys.

In the second approach to polyphonic sound generation, a synthesizer was assigned to a key only when the key was pressed—in effect, semi-polyphony. As early as 1973, the American company E-MU Systems introduced the *Modular Keyboard System Series 4050*, a digital keyboard that could be connected to up to ten monophonic synthesizers and thus enabled ten-voice polyphony. Nice try, but with the benefit of hindsight, the problems are glaringly obvious—very few people owned ten synthesizers, and the amount of time and effort involved in dialing in the settings for a new sound were an overwhelming deterrent. Memory for synthesizer settings was still waiting to be developed.

The same prerequisite—digital engineering—that was needed for semi-polyphonic keyboards, eventually led to synthesizers that allowed sounds to be stored with relative ease. Without the benefit of digital technology, early attempts at storing sounds produced unfeasible and complicated solutions; i. e.—a synthesizer with analog programmability required a dedicated row featuring all of the instrument's control elements, for every "memory" slot! In this case, a selector switch accessed one of the many identical control panels and connected it to the sound generator.

The first synthesizer featuring storage slots implemented in this manner was the GX1, which *Yamaha* released in 1975. The control elements for the system's storage slots were far from user-friendly. They were miniaturized to such an extent that they could only be adjusted using complicated tools—so-called programmers, comparators and tiny precision screwdrivers.

It was not until 1978 that the problem was resolved satisfactorily. With their introduction of the five-voice polyphonic *Prophet-5*, an American company, Sequential Circuits, treated the world to the first synthesizer with a global storage facility: All settings for each of its five onboard monophonic synthesizers were storable in memory slots, 40 in the debut model. Moreover, all five synthesizers shared a single user interface, which simplified matters considerably. In spite of its initially steep price, this instrument proved extremely popular and approximately 8,000 were built, up until 1985. In addition to its digitally implemented polyphony and memory, the success of the Prophet-5 is attributable to the outstanding quality of its analog sound generation system.

6.5 Digital Synthesizers

Even modern digital synthesizers featuring variable polyphony, memory, and completely digital sound generation systems are designed along this "semi-polyphonic" approach. The number of voices that these instruments are able to generate, however, is no longer dependent on the number of "built-in" monophonic synthesizers. Rather, polyphony depends entirely on the performance capability of the computers that power them.

The breathtaking developments in the digital world are best illustrated by the following example: The first program that emulated sound generation entirely by means of a computer was *Music I*, authored by the American programmer Max Mathew. Invented in 1957, it ran on a university mainframe, an exorbitantly expensive IBM 704. Unimpressively, its sole claim to fame was that it could compute a Triangle wave, although doing it in realtime was beyond its abilities.

This lack of capacity for realtime performance is the reason why early digital technology was used solely for control (and storage) purposes in commercially available synthesizers. Digital control circuitry debuted in 1971 in the form of the digital sequencer housed in the colossal *Synthi 100* modular synthesizer—in all other respects an analog synthesizer—from English company *EMS*. Priced out of the reach of all but the most well financed musicians, the Synthi 100 provided a sequencer featuring a whopping total of 256 events.

Ever-increasing performance made it possible to integrate digital technology into parts of the sound generation process itself. The monophonic *Harmonic Synthesizer*, manufactured by *Rocky Mountain Instruments* (RMI) was the first commercially available instrument to do so. This synthesizer had two digital Oscillators combined with analog filters and amplifier circuits.

The *Synclavier*, introduced in 1976 by *New England Digital Corporation* (*NED*), was the first synthesizer with completely digital sound generation. Instruments like the Synclavier were based on specialized processors, which had to be developed by the manufacturers themselves. This development cost made the Synclavier a “big ticket” item—an investment few could afford.

An alternative solution was the use of versatile processors made by third-party computer processor manufacturers. These processors, especially designed for multiplication and accumulation operations, which occur in demanding audio processing tasks, are called DSPs (Digital Signal Processors). *Peavey’s DPM-3*, released in 1990, was the first commercially available synthesizer completely based on standard DSPs. The instrument was 16-note polyphonic and based mainly on three Motorola 56001 DSPs. It featured an integrated sequencer and sample-based subtractive synthesis with factory preset and user definable samples.

Another solution was to design synthesizers as an additional unit for a computer, rather than as a stand-alone unit. The growing popularity of personal computers from the early 1980s made this option commercially viable: *Passport Soundchaser* and the *Syntauri alphaSyntauri* were the first examples of this concept. Both systems consisted of a processor card with a standard musical keyboard attached to it. The processor cards fit into the card slots of an Apple II computer, which was very popular in the United States at the time. The synthesizers were programmed by the use of the Apple keyboard and monitor. They where polyphonic, had programmable waveforms, envelopes and sequencers. Today’s soundcards, introduced in countless numbers since 1989, follow this concept.

Exploiting the ever-increasing processing power of today’s computers, the next step of the synthesizer evolution is the software synthesizer, which runs as an application on a host computer. The sound card is only needed for its A/D and D/A converters. The actual process of sound generation with effects, recording and sequencing is performed by the versatile central processing unit of your computer—as with Logic and your new ES2.

A

B

C

D

E

F

G

H

L

M

N

O