Hordijk G2 Pages

Introduction

Sound synthesis and sound design

Music has brought pleasure and entertainment to mankind throughout the whole of history. Each person is by nature equipped with one of the most elaborate and emotional musical instruments; the human voice. Whenever people feel good music seems to fit the occasion, and it is considered quite natural to hum or sing a song. Musical instruments have brought their own moods to music and at the current moment in human evolution there is an enormous variety of musical instruments available. The twentieth century has seen the development of a range of new and exciting electronic musical instruments. These electronic instruments are very flexible, they can produce a wide range of timbres and can be amplified to whatever loudness level sounds best for the occasion. Most of these electronic instruments are played by a keyboard, but in essence the keyboard can be replaced by any electromechanical device that is able to transform a movement caused by a human interaction into an electrical signal that can drive the sound generating core of the electronic instrument.

All sorts of technical and scientific developments have helped to create electronic instruments and the human interface to play them. Still, music is an art and not really a hard science, although music and sound have for a long time been subject to various scientific research. An important realization is that science can not really explain why much music is such a pleasure to listen to and such a joy to make. Which is not a bad thing, as probably no one is waiting for science to take the fun out of music by applying formalized rules and templates on what is also subject to 'feel'. So, although this book covers techniques that lean heavily on scientific research, the application of these techniques will in general be aimed at creating fun. There are a lot of professionals working with sound and even more people that make music for their personal enjoyment. Mastery of sound synthesis is valuable to all of them. Still, it won't be easy to please everyone with one single book, as some people will be more interested in how things work and others might want practical examples that just work. The aim of this book is that it can at least be used as a practical guide in workshops and courses in electronic music, covering some essential basics that are needed to operate the equipment used in sound synthesis in a way that makes some sense. Additionally it can be used to explore techniques to find out how they can help in the development of one's own musical style.

Sound synthesis is the art of creating sounds by using suitable electronic means, using either analog or digital electronic devices. Sound design is the art of creating particular sounds using sound synthesis techniques. The definition of sound design as used here might be confusing to some, as the name sound design is also used in the discipline in industrial design that occupies itself with how mass produced objects should sound. Examples are how the sound of cars or ladyshaves are 'designed' to sound pleasing while in use. Which of course has nothing to do at all with music or sound synthesizers. This book puts the emphasis on the various synthesis techniques for musical purposes and how to setup sound synthesizers to create a large range of characteristic musical sounds. The art of musical sound design is left to the artist.

Psychoacoustics

Most scientific research has been concentrated on what is named psychoacoustics, which is basically the research on how all sorts of sonic phenomena are perceived by the human mind. It should never be forgotten that the human mind is the final link in any audio chain. Meaning that the most important property of any artificial sound is 'how it sounds', no matter how complex or simple it is to create that artificial sound. This 'how it sounds' is basically equivalent to how the sound is actually perceived in the human mind. The ultimate mastery of sound synthesis is to be able to create sounds that sound good to the ear. Those sounds don't necessarily have to be made with complex techniques or equipment that is difficult to understand, the basic idea is that when it sounds good it simply sounds good. And if it doesn't there is still some work to be done. Anyway, whatever makes a sound sound good to the ear is valid.

From a psychological point of view sound is a manifestation in the human awareness. This means that when a sound is heard it is exclusively the perception itself that manifests in the human mind. All that is involved in making music will eventually induce this perception and the nature of the perception will fill part of the human awareness. What happens in the brain is not really part of the synthesis process itself, but the synthesis process should take into account that the human brain acts like a filter that molds the perception into a form that depends on the condition of the human mind. E.g. one must be in the mood for music to enjoy it fully. Matters like personal taste, fatigue, the social surroundings, etc., will all influence the enjoyment of music. Another and more general factor is how the brain itself processes the incoming auditory information on a 'raw data' level. The original function of hearing is not to enjoy music but to gather information from the immediate surroundings. Sounds will draw the attention to things happening around us, enabling the human mind to e.g. detect danger. This process works on a half-concious level, meaning that the attention is drawn before the mind can start to think about it. This mechanism has been useful in prehistoric times to warn for immediate dangers like hungry ferocious animals sneaking up from behind. In modern times it is still functional, e.g. when driving a car all sorts of sounds enter the mind at a half-conscious level and cause immediate reaction to avoid dangerous situations. In detecting danger through hearing the sense of space and distance is very important. A soft rustling sound that is very close can mean a more immediate danger as a low roaring sound heard at a long distance. However, another type of soft rustling sound might actually give a comfortable feel. So, a very important property of a particular sound is how it focuses the attention and what sort of sense it will in general introduce in the human mind, again taking into account the state and surroundings a person is in. As this process of focusing happens before one can even think about it, it can be stated that each sound itself has a property that defines how it will by default focus the attention. The wondrous thing about the human mind is that it can focus on so many different sounds and immediately give them some meaning in a vast range of settings.

Happy accidents

There is still a lot of unexplored territory in sound synthesis, as there is such a broad range of flexible sound synthesis techniques available. Creating artificial sounds by electronic means often leads to unexpected results. Some results sound very good and others very bad, while many will be somewhere in between. Happy accidents in sound synthesis are quire rewarding, as they can be immediately explored musically and lead to new forms or compositions. It is not a bad thing to be inspired by some weird sound and try to weave a musical pattern around it. In fact, this is a valid musical improvisation technique. To be able to reproduce the happy accident later it is quite important to be able to detect when such an accident happens and to quickly grasp the nature of the accident. This requires experience, when starting to use synthesis techniques happy accidents will often happen but be quickly gone and leave one wondering why it did sound so good and how that came about. When experience starts to give more grip on what is happening the nature of happy accidents gets understood more quickly and eventually become a new technique that can be used at will. This gives a lot of fun, so much that experimentation and electronic improvisation can become quite addictive. Still, music is often a mix of many different and sometimes delicate sounds and it is always important to judge a sound on how it works out in a musical arrangement.

Technology and sound design

Research on the various technical ways that specific sounds can be generated and processed by electronic means, sometimes referred to as sonology, has provided the musician and composer with many new musically useful techniques and helped to develop new electronic musical instruments that are now taken for granted in today's music. These electronic instruments employing sound synthesis techniques have become known as sound synthesizers or synths. Sometimes the instrument exists as computer software only, in which case the instrument is named a softsynth. Application of sound synthesis techniques to create sounds for musical purposes has become known as sound design, which is a form of art where musical sounds are created and built from the ground up, sounds with the purpose of being used in some musical way. Sound design covers the whole process of creating the sounds to play with or to use in compositions, design refers to the creative process as a whole and synthesis refers to the more technical side of the creative process. Let's take as an example the design of a hornlike sound to be played on an electronic keyboard. To create such a sound, the sound designing artist can choose from several available tools and techniques. What makes sound design an art is that the ear is always the final judge, although a lot of knowledge can be used to initially set up the sound. The last tweaks on the sound must be done by ear and not according to scientific rules. In the end the only rule that applies is if it sounds good to the ear and the sound has the right feel.

The name synthesizer refers to several classes of electronic musical instruments, classes that can be based on totally different technical concepts. The popular notion of a synthesizer is that of a musical instrument with lots of flickering lights, knobs and buttons. This romantic image is perhaps caused by the association with the imagery of science fiction in the fifties and sixties of the twentieth century. There is also some vague notion of 'the typical synthesizer sound', but on closer inspection this type of sound might as well have been made by an electric guitar or an acoustic recording immersed in an array of spatial sound effects. In fact, there is no such thing as 'the typical synthesizer sound', sound synthesizers can produce such a huge number of totally different sounds that not one of them can distinctly characterise 'the sound of the synthesizer'.

Types of synthesizers

As said, in this book sound synthesis literally means the process of creating musical sounds using a dedicated sound synthesizer, provided this synthesizer has all the necessary tools to offer dynamic and detailed control of the created sounds. The most flexible type of synthesizer to use for this purpose is definitely the modular synthesizer. Today's modular synthesizers appear in three instances, the traditional analog modular, the digital modular based on DSP techniques and the modular softsynth running as a software-only application on a personal computer. The last two instances are commonly referred to as virtual modular synthesizers, as they emulate to some extend the traditional analog modular synthesizer. All three instances have their little sonical advantages and disadvantages, but the synthesis techniques themselves are basically the same on all three. Analog modular synthesizers are really a collection of small and independently working devices, named modules, housed in one single cabinet. These modules can be freely reconfigured and reconnected to suit any musical need. This freedom offers endless sonic possibilities, some of the produced sounds are great while others might sound like nothing at all. There is a similarity to the palette of a painter, although there might be paint in many colours on the palette, that doesn't yet say anything about the final painting. The art of painting is how to paint a picture with the available paint by mixing the right colours from the basic colours on the palette. The technique of painting is obviously a part of the art of painting, but for a person looking at the finished picture, the palette and brushes the painter has used are in general totally irrelevant. Still, for the painter these are quite essential, simply as they define what the painter can and can not do. It is exactly the same with a musician using a modular synthesizer, the artist has to learn to interprete and use the possibilities of the instrument to be able to put it to a musical use. Additionally, a sound that sounds very bad in one musical context can sound great in another musical context.

All techniques discussed later in this book will to some extend be possible on the earlier mentioned three instances of the modular synthesizer, provided the necessary modules are present in the system. Most digital modular systems have the advantage that if an extra module is needed it can be instantly created as a new instance in the software. In contrast, on the analog modular it is necessary to go to the shop and buy the extra module. Still, the feel of working with an analog modular is still highly valued and many musicians are still willing to pay vast sums of money for a traditional analog modular system.

The fun with any modular synthesizer is that everything is allowed, there are no rules of what or what not to do with a sound synthesizer. Instead, there is the complete freedom to connect the modules in whatever way one feels like. Experimenting with less obvious connections is definitely part of the fun. The range of possible sounds is endless, there will always be new sounds left to be discovered and musically explored.

Short history of electronic musical instruments

Nineteenth century

Before a new technique is developed it is necessary that the underlying physical principles are discovered and examined first. The nineteenth century was a time where there was the social freedom to question the nature of natural phenomena, including the physical nature of sound. E.g. the first attempts to understand why equally pitched sounds can sound completely different took place in the nineteenth century. In 1822 the scientist Jean Baptiste Joseph Fourier published a study about how wave phenomena like soundwaves can be mathematically described and analysed by series of harmonically related sine and cosine functions. This mathematical method will become known as the Fourier Transformation. The method is used in 1863 by Hermann Ludwig Ferdinand von Helmholtz in his research on sound and acoustics. Helmholtz proves with an experiment that all pitched sounds are made up of a number of sinewaves with certain pitch relations, named harmonics. The Helmholtz experiment can isolate a single harmonic sinewave by a simple device that will become known as the Helmholtz resonator, in its most simple form a hollow glass ball with a little hole. The air in the ball's cavity can resonate at a certain pitch, the pitch depending on the dimensions of the ball. Helmholtz' study shows that the resonator can convert the kinetic energy of the vibrating air into warmth. When a harmonic component in a sound is equal to the resonant frequency of the resonator, the resonator will damp the loudness level of that harmonic component by converting the sound energy of the harmonic into warmth in the cavity of the ball, which causes the temperature of the ball to be increased. Helmholz noticed that this experiment also resulted in a change in timbre of the sound. So, this experiment also proved that the timbre of a sound depends on the relationship between loudness levels of the harmonic components that are present in the sound. Using modern digital measuring devices the loudness levels of these harmonic components can be calculated by taking a sample of one cycle of the waveform and then apply the Fourier transformation on the sample.This principle is the foundation for a technique named additive synthesis, a method where any conceivable sound can be synthesized by separately generating all the necessary harmonic components and mixing them together in certain volume ratios. Another popular technique that relies heavily on the Fourier transformation is convolution. This convolution technique makes it possible to superimpose characteristics of one sound on another sound. Convolution needs to do an enormous amount of calculations, but by using the Fourier math the amount of necessary calculations can be dramatically reduced. It is interesting to note that techniques like convolution, that have only become practical because of the advent of fast computers, do many times have their roots a long, long time ago.

First half of the twentieth century

Musical instruments reflect to a certain extend the technological level of the culture using the instrument. Up to the beginning of the twentieth century it is mainly the materials wood, metal, ivory, leather, ceramics, etc., that are used to build musical instruments. It is no surprise that when electronics becomes a common technology in the twentieth century it is used extensively in new types of musical instruments. The development of electronic musical instruments walks along with the refinements of electronic technology, spanning a period of over a hundred years. In the year 1906 Lee DeForest invents the triode vacuumtube, which he names the Audion. This device is capable of amplifying electrical signals, enabling the design of 'active' electronic devices like the audio amplifier and the radio. The oscillator circuits and filters that are used in radio technology inspire the russian inventor Lev Thermen in the early twenties to invent a completely new type of musical instrument, the Theremin. The instrument is fully electronic, without any mechanical parts used to generate sound. The Theremin is played by moving the hands towards two antenna's. One antenna controls the pitch while the other controls the volume of the sound. The way pitches are generated is based on what is named the superheterodyne principle, a technique where two radio frequencies are mixed, resulting in signals that contain the difference and sum of the original frequencies. Thermen chooses the radio frequencies in such a way that the resulting difference frequency is within the human hearing range. Detuning one of the original frequencies by waving a hand near an antenna results in a gliding pitch change. The Theremin is a very difficult instrument to master, only few musicians dare to play it. One of the mysterious aspects of the instrument is that during play it is not touched by the musician, which at the time added much to its futuristic image. In the year 1929 the american inventor Laurens Hammond starts to develop an organ based on tonewheels. The very stable electromotor he invented earlier is used to rotate the tonewheels in a precisely controlled manner. The use of tonewheels had already been used by e.g. Thaddeus Cahill in his Telharmonium, built around the year 1900. But the Telharmonium was gigantic in size, as it was constructed of big electricity generators that occupied a complete building. Hammond used vacuum tubes as amplifiers, enabling him to build his organ in a much more manageable size. After the Hammond tonewheel organ is brought to the market in 1935 it immediately starts to play an important role in popular music. The big difference with the Theremin is that the Hammond organ can be readily played by anyone knowing how to play a piano or organ keyboard, so there is an immediate market of the instrument. The tonewheels generate sinewaves that are mixed in certain ratios, making the Hammond organ an example of an electronic musical instrument based on the principles of additive synthesis. As the pitches which the organ can produce depend on both mechanical and electronic devices, this class of instruments is named an electromechanical instrument. Later on, in the year 1939 Hammond develops the Novachord, a version of his organ where the tonewheels are replaced by electrical circuits, making it a completely electronic organ.

The first half of the twentieth century sees the invention of more electronic and electromechanic musical instruments, some with more appeal to the public than others. The main property that all these instruments have in common is that they all are to be played live by a musician.

The fifties and sixties

Around the year 1950 the taperecorder becomes available. And although the taperecorder is not perceived as a musical instrument, its invention soon turns out to be a very important event in the history of music, as the taperecorder offers the ability to manipulate recordings in a way that was unconceivable before. Tapes can easily be played speeded up, speeded down or played in reverse. These manipulations change the original timbre of the recorded sounds in a dramatic way. New pitches can be created by changing the playback speed and subsequent recording on a second taperecorder. Manipulations by changing playback speed had already been done before by using wire recorders and gramophones, e.g. by Edgar Varèse, but using a taperecorder turned out to be more practical. However, the real new thing the taperecorder offered was the possibility to splice the tape in parts and assemble these parts in a different order. With this splicing technique a composer is able to assemble a melodic composition from snippets of sounds by splicing the tape, making overdubs and rerecording at different speeds. This made the taperecorder immediately the central component in the recording studio.

Also new was that the whole setup in the recording studio became like one, new instrument for composers, offering them a totally new concept for composing. In contrast, before 1950 virtually all music is composed to be played live by musicians. Recordings on gramophone had to be done in one single take for the whole orchestra at once. After 1950 recordings on tape can be made in different places at different moments in time and be manipulated and assembled later in the studio. Many composers readily understood the new possibilities and started to experiment with this new medium. This resulted in new musical genres like tape compositions and electronic music.

The recorded source material to be manipulated can be recordings of literally anything. Like the sounds everyday objects make when hit, bowed, scratched, crushed, crashed, etc. Another source of sounds are electronic laboratory instruments normally used for measurements in electronic circuits, like tone generators, noise generators and audio filters. When material is rerecorded on a second taperecorder the sound can be manipulated during the transfer. Manipulations like audio filtering, distortion, amplitude modulation and the addition of echo or reverberation, can drastically change the colour of the timbre and add spatial characteristics to the sounds. These manipulations were named treatments and would soon become more and more important in the composing process. Although a treatment is the actual manipulation done to a sound, the 'box' that did the manipulation was referred to as treatment as well.

The typical fifties experimental recording studio consists of a big table with two or more taperecorders and a tape splicing device. Microphones are present to do acoustic recordings. Next to these are a mixing desk and a collection of tonegenerators and treatments. Equipment didn't necessarily have to be in the same room. Like in the beginning of the fifties, when the WDR broadcasting company in Cologne, Germany, where the composer Stockhausen did a lot of his work, didn't have tonegenerators in their studio for electronic music. In fact, an engineer had to go to the laboratory department on another floor and route the output of a tonegenerator to the audio cabling system that ran through the building. In the recording studio the signal could be picked up from this cabling system. Then, the taperecorder operator and the tonegenerator operator had to communicate instructions to each other over an internal telephone line. It was tricky to interconnect the tonegenerators and treatments, as these were often not designed for this use. Signal levels could differ considerably resulting in excessive noise, unwanted distortions or even the premature death of a piece of equipment. In the WDR studio it wasn't even allowed for a composer to directly operate the equipment, perhaps because the management feared for liability issues in case of premature death of the composer by electrocution. Instead a special composer's assistant was appointed to operate the equipment under the direction of the composer.

It soon became clear that there was a need for standardization of signal levels. Another need that arose was to be able to remotely control the functions on the taperecorders and treatments. At the time using electrical voltages seemed the best way to do this, resulting in what will become known as voltage control. The first use of voltage control is to toggle relays that start and stop the taperecorders or distribute the audio signals. The actual value of the controlling voltage didn't matter as long as it had enough power to toggle the relay. Some time later voltages are used to directly control the loudness contour of the audio by using lamps and light sensitive resistors or photo cells. In this case the actual voltage level does matter, as it directly represents the actual volume. In the late fifties there is the notion that every function like pitch or timbre could be made to depend on the actual level of a voltage, meaning that any musical property can be expressed by a voltage of a certain level. Additional advantage of voltage control is that the controlling voltages can be processed before being used. Many treatments that could be used on audio signals could be used on the controlling signals as well. Other possible treatments on control voltages are similar to the processes found on the analog computers of that time, like adding, subtracting, offsetting and multiplying of voltage levels. Voltage control turned out to be very useful in especially serialist composing techniques. In the period between 1960 and 1965 transistors start to replace vacuumtubes in electronic circuitry. The transistor makes voltage control much easier to implement and by the year 1965 voltage controlled equipment seems as much part of the electronic studio as the taperecorder.

The modular synthesizer

There is a clear link between the collection of equipment surrounding the taperecorders in the early experimental electronic studios and the first sound synthesizers. Around 1965 the equipment is redesigned to be assembled into singular standardized systems, with as much functions controlled by voltage levels as is technically feasible. Influential electronics designers and manufacturers in this period are Don Buchla and Robert Moog. The Moog systems become known to the public as synthesizers. Although Buchla initially opposes the name synthesizer, he names his system the Buchla Box, the word synthesizer soon becomes the brand name for Moog and Buchla systems and similar systems from other manufacturers.

Splicing tape is a tedious process and there was a clear need for a technique that could replace parts of the tapesplicing process. This leads to the development of a device named a sequencer. This is a box that can generate a short sequence of individually programmable voltage values. The time that a voltage is available is named a step and can have a fixed or variable length in time. After programming the voltage values the sequence can be started by hand to ‘step’ through the sequence, or it can be set to loop the sequence forever. The voltage values can represent a note sequence, e.g. short arpeggio’s or programmed melodies, or any other musical events that can be controlled through a control voltage. Raymond Scott, a composer and inventor from New York, had already built a huge sequencing machine in the fifties, which he named the UJT-Relay sequencer. It used oscillators driven by relays to play preprogrammed melodies and rhythms. Scott was a commercial composer and he kept his invention secret for many years, hoping that his machine would give him an advantage over other commercial composers.

Don Buchla’s Music Box, which he developed for the San Fransisco Tape Centre in 1965, included a voltage sequencer module, with the purpose of replacing some steps in the tape splicing process. Bob Moog didn’t incorporate a sequencer in his system until some time later, though Moog was one of the very few people who had seen the sequencer built by Raymond Scott. The story goes that as being a friend of Scott and out of respect for his friend, Moog omitted a sequencer in the first systems he built.

The first generation of synthesizers are referred to as being modular synthesizers because all sound generators and treatments are available as independently working modules. Both Moog and Buchla could assemble a system including any number of different modules to the needs (and budgets) of the customers. The modules in the system can be interconnected with cables that were named patchcords. Using patchcords allows for free routing of signals through the available modules. It also allows for feedback and the use of audio signals as controlling signals. With some fifteen different types of modules in a typical system an enormous range of sounds and sound effects can be made by different interconnections and knobsettings. The total of connections and knobsettings is named a patch. It became custom to draw schematics of the module interconnections and knobsettings to be able to reproduce the same patch later. These schematics were named patchsheets.

Moog used a method of voltage control where the relation between the voltage level and the function is exponential, normalising an increase of exactly 1 Volt to a raise in pitch of exactly one octave, the Volt/Oct norm. This system makes it easy to play the pitch of the sound generators in equally tempered scales. All inputs and outputs of modules use the range of this Volt/Oct normalization, so all signals can be used as controlling signals as well. Controlling a function on a module by a varying signal generated by another module is named modulating. It adds tremendously to the sonic power of synthesizers, enabling the generation of totally new and as yet unheard sounds. Buchla on the other hand used a method of voltage control where the relation between the voltage level and the function is linear, named the Volt/Hertz norm. This system makes it difficult to play the pitch in equally tempered scales, but makes it much easier to play in just tuned scales and make tunings based on harmonic relations that work very well with certain more advanced synthesis techniques. This made the Buchla system more oriented towards sound designers for sound effects and advertisements and the more experimentally minded composers. But no matter the normalization used, voltage control makes it possible to control the synthesizer by literally anything that can produce voltages. This is important to realize as it means that the musician's interface is in essence not a part of the synthesizer itself, the synthesizer can be connected to a vast range of musician's interfaces or elctronic or electromechanic sensors. It also allows the synthesizer to be played by other machines, as long as they can produce the necessary controlling voltages in a sensible voltage range. So, the synthesizer can also be played by another synthesizer. This means that a modular synthesizer is in essence an open-ended system with unlimited expansion possibilities. A modular synthesizer also allows for feedback, where the output of a module is used to operate upon its own input, creating a recursive operation upon itself. Proper feedback of processed control voltages allows the synthesizer to compose by itself. To do so the composer 'feeds the synthesizer a set of rules' to which the machine has to adhere, and then lets the synthesizer run by itself. These rules can e.g. be implied in the way feedback is applied.

In the second half of the sixties some performing musicians express their wish to be able to play the synthesizer live. For Bob Moog this is a commercial market he couldn't ignore, so the organ keyboard is adapted in a way that it can generate the necessary control signals to enable the synthesizer to be played live. More experimental interfaces are developed, like e.g. the ribbon controller, but the keyboard will prove to be the most successful commercially.

The prepatched synthesizer

The modular synthesizer is in essence a studio instrument and developed as a composers tool. It is hard to use on the road, as it is bulky and very sensitive to changes in temperature. The first modular systems didn't have temperature compensation and needed constant retuning while performing. Repatching to get a different sound is tedious work and very difficult during a live performance. Around 1969 a smaller and portable type of synthesizer appears, the prepatched synthesizer, which is much more a musician oriented performance instrument. It became clear that a certain type of patch was used many times by keyboardists and these smaller synthesizers had this patch hardwired internally, hence the naming prepatched. This reduced the need for patching cables as different sounds could easily be created by only throwing a couple of switches and tweaking the knobs. Three instruments from different manufacturers appeared almost at the same time around 1969, the Minimoog by Moog, the ARP2600 by ARP and the british VCS3 by EMS. The Minimoog is completely hardwired internally. The ARP2600 is still partially modular as patchcords could be used to override the internal interconnections. The VCS3 has no internal hardwiring but instead uses a small pin matrix to make the connections between the small set of modules it houses, so in fact it is still a true modular synthesizer.

These three instruments mark the beginning of a new generation of synthesizers. Very important to the musician is that these synthesizers are in essence monophonic. This might appear a limitation, but it in fact it enables keyboard players to play the same type of solo's like saxophonists and guitarists play, and so get a bit more in the spotlight on stage. Synthesizers like the Minimoog have added play controllers like pitchbenders and modulation wheels that let the musician bend and modulate notes in ways that allow for very expressive soloing. Another feature is that the sound of these synthesizers has enough power to stand out against other heavily amplified instruments in the typical electric bands of the seventies. These features quickly makes this generation of synthesizers very popular amongst keyboard players and the prepatched synthesizer becomes one of the basic instruments in the electric popband. Manufacture of modular systems is soon ceased in favour of manufacture of these portable prepatched synthesizers. Still, the much greater flexibility of modular synthesizers compared to prepatched synthesizers is up to this day highly valued. Using a modular synthesizer these days, no matter if it is analog or digital, is still considered playing topleague in sound synthesis.

The polysynth and preset synthesizers

Around 1978 the prepatched synthesizer becomes polyphonic, the polysynth. In the first half of the eighties digital techniques and mass production make the polysynth a fully matured, reliable and wellrespected musical instrument. The new chip technology enables the manufacture of complete analog modules into single chips and these match enough to be used in a polyphonic system, where each voice has to match the other voices exactly. Two chip manufacturers supply the synthesizer industry with these chips, Solid State Music and Curtis Electromusic Specialties. Some of their chips, prefixed by the codes SSM or CEM, are still manufactured and available up to today. Wellknown polysynths around 1980 are the six voice polyphonic Memorymoog and the five voice polyphonic ProphetV. The Prophet V is built by Sequential Circuits, the company of synthesizer designer Dave Smith.

Digital technology is needed to control a polyphonic system. Digital chips are used to scan the keyboard for chords and to distribute the correct control voltages for a particular key to the modules. There is a crucial difference between the architecture of a polysynth and the monophonic prepatched synthesizer, which by this time gets named as the monosynth. While on a monosynth the knobs connect directly to the sound generating and modifying circuits, in the polysynth a little computerchip known as a microcontroller is put between the knobs and the sound circuitry. This microcontroller has the intelligence programmed into it on how to measure the control voltages or sources and process them digitally into new values that are distributed to their respective destinations. The source values and their destinations are in fact the patch, and in this way control the final sound. These values and destinations can be stored together in a preset memory connected to the microcontroller and can be recalled as a single entity, named a preset. Recalling a preset takes only a few milliseconds, fast enough to be done while playing. This is an enormous improvement over the patching of cables by hand on a sixties modular synthesizer. On the polysynth of the early eighties digital technology is used only to process the control signals. The microcontroller does not yet do digital soundgeneration or processing of audio signals, sound synthesis itself is still done by using analog electronics.

The multitimbral synthesizer and MIDI

Synthesizers can be used to play different instruments in an arrangement. To do this live several synthesizers are needed, each one set to the sound of one of the instruments in the arrangement. In the first half of the eighties the polyphonic preset synthesizer is adapted in a way that each voice can play a different instrumental sound. By splitting the keyboard in sections, and assigning each section to a different sounding voice, it is possible to use the instrument in a multitimbral way. It is also possible to stack different sounds upon each other, resulting in very thick symphonic textures. However, there is still only a limited number of voices available on the polysynth, typically four to eight voices, and with this technique one runs easily out of voices. Connection of polyphonic synthesizers to each other by means of control voltages and patchcords is in practice too complicated to be feasible. For this reason Sequential Circuits developed a digital means of connecting synthesizers to be able to have one synthesizer play several others. More manufacturers, like the Japanese instrument building company Roland, see the sense of this idea and after adding some minor modifications they together decide to promote this digital connection as an industry standard, to be used on every new synthesizer. The connection is baptised MIDI, an acronym for Musical Instrument Digital Interface.

MIDI is both a hardware and a software specification. The hardware is simple, very similar to the way printers and telephone modems are connected to computers. But the power is in the software. Through MIDI a synthesizer can send a set of commands to another synthesizer, e.g. a command to play a certain note. This set of commands is named the MIDI Protocol. Each command is assigned to a MIDI channel of which there are sixteen. A synthesizer can be set to react to commands in one specific channel only, or to act on commands received in any of the sixteen channels.

In the MIDI software specification symbols are assigned to possible musical events, the symbol being represented by a short digital code. The specification defines how values can be added to the symbol to send well-formed commands. Technically the command symbol is expressed as a hexadecimal digit. There is a symbol for the pressing of a key, together with a channel number, a value denoting which key is actually pressed and a value denoting the velocity of the keypress. This symbol is paired with another symbol that stands for the release of a key, again with a channel number, a value to identify which key is released and the velocity at which it is released. The number of the channel in which the command should act is embedded with the command symbol in the first part of the command. There are seven commands that can act in a single channel; NoteOn, NoteOff, PolyphonicKeyPressure, ControllerChange, ProgramChange, ChannelPressure and PitchwheelChange. Next to these commands there are commands that act globally and are not tight to a single channel. These global commands have to do with timing information and start/stop control for automated devices like sequencers and recording devices. Additionally there is a SysEx command that can encapsulate manufacturer and model-specific information of variable length. The SysEx command enables the transfer of presets to another synthesizer or to a computer. SysEx also enables the transfer of short samples of sound. It is up to the manufacturer to decide on which commands a specific synthesizer model should react and which commands it can send to other devices. This is written down in a table named the MIDI specification sheet that should be somewhere in the back of the manual of every MIDI equipped synthesizer.

When several MIDI equipped synthesizers are used, only one of them requires a keyboard, as all the other synths can be played by that keyboard through the MIDI cabling. This leads to a new type of synthesizer named a MIDI expander. It is a synthesizer without the keyboard but with a MIDI connection to be able to play it from another synth. Omitting the keyboard makes expanders less expensive than their keyboard equipped versions. Most expanders are rackmountable and the name racksynthesizer becomes another common name for the expander. After the year 1985 many new synthesizers come in both a keyboard and the cheaper rack version.

Another important feature of MIDI is that the MIDI commands themselves can be recorded on a computer. The introduction of MIDI is around the same time that personal computers become available. Around the year 1984 the computer manufacturer Atari does a very clever move by including MIDI interface connectors on their new ST series of budget computers. Very soon software programs appear for the Atari computer that allow to record, edit and playback the MIDI information. This means that an arrangement can be played live into the computer, the software recording only the play information. After recording the play information the arrangement can be heavily edited and new parts added or deleted at will. Intermediate states of the arrangement can be conveniently saved on a floppy disk to be recalled at a later time. Many popular music software programs in use today saw their first implementations on the Atari ST computers.

To control synthesizers that did not yet have MIDI, like the older analog modular synthesizers, devices named MIDI to CV converters are developed. Such devices are capable of converting the incoming stream of commands into one or more control voltages and gate signals that can directly control the analog modules.

Digital sound synthesis techniques

The first steps in this field were done in 1957 by Max Matthews at Bell Labs in the United States. Mathews had written the program Music I as a 'socially desirable' side project next to his official job at Bell Labs. The first rendering of a 17 second long audio file using Music I is said to be the first computer generated sound. Mathews kept on developing his Music software through different versions over many years, having a decisive influence on what is now known as computer music. In the early sixties many universities and research institutes that had access to computers started to experiment with calculating soundwaves directly by computer programs. The technique of generating and manipulating soundwaves in the digital domain is based on the principle of chopping the soundwave in a sequence of very small timeslices, named samples. Every sample becomes in fact a single value that represents the average mean of the sound signal during the short period the sample is pending. The device that can slice and measure the timeslices is named an analog to digital or AD converter. When the rate of slicing is about two and a half times the highest pitch perceivable by the human ear, the sequence of samples is perceived as a continuous audio signal, in the same way as in a movie twentyfive still pictures a second appear to project a fluid motion to the human eye. This means that in practice the sound signal must be sampled at least between fourtythousand and fiftythousand times a second. The number of measurements per second is named the samplerate of the digitized sound.

Another requirement is a high enough accuracy for the measurement of the mean value of the signal during a single sample period. This accuracy must be somewhere around the noisefloor of the signal to be sampled. The noisefloor is the point where a signal is so low in level that it starts to become indistinguisable from the natural noise present in the analog parts of the signal chain. The accuray or resolution of digital numbers is represented as the number of bits used to represent the value, the more bits the higher the accuracy, and if the values represented by the bits are fixed point or floating point values. In any case, the measurement has to span the whole dynamic range of the signal. In practice the dynamic range is the space between the loudest level that can be recorded without distortion and the noisefloor. In the case of fixed point values there is a simple relation between the amount of bits in the digital number representing the value and the dynamic range of the signal; each extra bit will increase the dynamic range by 6 dB. For a professional taperecorder the dynamic range is about 60 dB, which means that at least ten bits of resolution would be needed to represent this range. But there is a bit more to it than this simple assumption, recording tape can be overdriven, causing the tape to be saturated. This tape saturation is not really problematic when it happens now and then. In fact, a little tape saturation effect is said to sound good. But when a signal is digitised with an AD converter and there is a peak in the signal that exceeds the measurement range, then there will be an effect named clipping. Clipping sounds awful and must be avoided at all costs during a recording. To reduce the chances of clipping some extra headroom is needed, requiring some extra bits. These days it is common to use 24 bit converters for professional level audio recording, not only to reduce noise as 24 bit is well below the noisefloor of the human ear, but specifically for offering more headroom during the recording and mixing. For the final mixed recording an average resolution of at least 14 to 15 bits is needed, as the digitization process itself adds its own sort of digital noise, adding to the noisefloor. This has become the standard for a Compact Disk with its sample rate of 44.1 kHz and an average resolution of around 15 bits. To go back from the digital numbers to an analog audio signal that can be fed to a loudspeaker a device named a digital to analog or DA converter is used. To take an analogy with a tape recorder, the AD converter is functionally similar to the recording head and the DA converter to the playback head, the recording tape being some appropriate type of memory device in the computer or some type of mass memory storage like a harddisk, a CD, a DVD, a flash-memory card, an optical disk, etc.

The whole idea of digital sound synthesis is to have the computer calculate the list of values or samples that together in one long row represent the sound signal. The calculations are in general rather simple, but they have to be repeated for each single sample, still requiring a very powerful computer. In the sixties computers were definitely not yet up to the task to make digital recordings with a high enough sample rate, simply as the memory was rather slow and way too expensive to be waisted on a snippet of ordinary sound. However, the method of generating sound was feasible by having the little programs run maybe fivethousand times a second and recording the DAconverted results on a taperecorder running at a relatively low speed. After the recording the tape is played back at a speed some eight times faster to produce the required quality. Rerecorded on another tape would create the master tape for a record or to be played during a presentation, radio broadcast or concert.

Digital signal processors

After the first silicon chips came available in the sixties chip technology has developed in an incredible speed. Around the start of the eighties the VLSI or 'very large scale integration' technique is available for mass production of digital chips, enabling manufacture of chips with millions of transistors on an area the size of a poststamp. In the early eighties a special type of very powerful computerchip is developed, optimized to do repeated calculations like those used in sound synthesis and sound modification. This type of chip is named a Digital Signal Processor or DSP. The initial reason why synthesizer manufacturers are interested in this technology is because analog oscillators are hopelessly temperature sensitive, making their pitches drift constantly. The temperature compensation techniques needed in especially polysynths put quite a burden on their manufacture. A DSP can be programmed to emulate an oscillator without the dreaded temperature drifts, finally enabling the use of promising synthesis techniques which need rockstable oscillators, like the linear FM technique. The first commercially available synthesizer based on a DSP chip is the Yamaha DX7, its synthesis based on the linear FM technique, already researched in the late sixties by John Chowning. The sixteen voice polyphonic and MIDI equipped DX7 became immensely popular overnight, though it was a drag to program useful sounds oneself. But it came with a big factory preset library on board with reasonably convincing electric piano, organ and brass sounds. One of the main reasons why it became such a popular instrument was its relatively light weight; it was sÛ easy to take it to a gig and provide the average keyboard musician with the most common 'bread'n butter' sounds. Being able to produce relatively light weight instruments is definitely a big advantage of using DSP chips. At the moment almost every new synthesizer uses a DSP somewhere in its internals, either for sound synthesis or to add effects like chorus, echo and reverberation.

The sampler

Another development in the early eighties extends directly on the taperecorder and the tape manipulation techniques developed in the fifties. This development goes back to the late sixties when an instrument named the Mellotron is developed and marketed. The Mellotron houses a mechanism of small tapes and playback heads, each one dedicated to a key of the small organ-type keyboard. On each tape is a fixed recording of some sound at a certain pitch, and if the corresponding key is pressed the sound is played back. After a key is released its corresponding tape is quickly rewound. The Mellotron came with factory recorded tapes with a choice of orchestral ensembles, string sections, brass sections, silver flutes and the like. By using a Mellotron a recording studio didn't have to hire an orchestra for budget recordings, saving immensely in time and money. The Mellotron also became popular with the symphonic and psychedelic rockbands at the end of the sixties. On request the factory could fit the Mellotron with custom recordings. Much of the sound effects of the popular British television series Dr. Who were put in a Mellotron, so they could be easily reproduced on demand.

The big disadvantage of the Mellotron is that it is a mechanical device. Both the tapes and mechanics wear quickly over time, needing expensive servicing. Taking the instrument on a tour wasn't very healthy either. Around 1980 digital techniques offer a solution and a new type of instrument is developed, named a sampler. The basic idea of the sampler is in fact not much different to that of the Mellotron, the tape being simply replaced by digital memorychips. The playback heads are replaced by a DSP chip that reads digitized sounds from the digital memory and routes them to a DA converter. An interesting feature is that all digitized sounds can share the same memory, and the DSP can play a single digitized sound polyphonically at different pitches. In the beginning period of samplers two instruments are starring the stage, the Fairlight CMI and the NED Synclavier. Both are in essence quite traditional computer systems completed with dedicated hardware for AD and DA conversion for recording and playback of audio. Both have a organ-type keyboard to play notes, but control and programming sounds is done by means of a video monitor and an ASCII keyboard. Both came in a big 19î system rack, with the typical late seventies computerlook. Noisy fans and eight inch diskettes made the scene complete. The big advantage over the Mellotron was that different sounds could be quickly and conveniently loaded from a computer disk, while replacing sounds on the Mellotron was a complicated and time consuming mechanical operation. Sounds could be sampled instantly on the spot and trimmed and saved for later use. But there was more, sounds could be manipulated by the system processor and a copy of the manipulated sound saved as a new, independent sound. It was even possible to generate sounds by sound synthesis programs run on the system processor and again save the results for later use. The big disadvantage of both systems was their cost, a very substantial sum of money had to change hands before a musician could name him/herself the proud owner of a Fairlight or Synclavier system.

The first serious competition is the Emulator. With the appearance of the average polysynth and a pricetag that, although still pretty stiff, is way below the Fairlight/Synclavier pricetag, it does pave the way for the average professional keyboard musician to explore sampling. But the breakthrough for the sampler is in 1986 when Akai releases the S900 sampler, a rack device with a very reasonable pricetag, but with outstanding specifications for that time. Around 1986 sampling had much appeal, maybe as before the S900 it was so far beyond the budget of most musicians. This quickly made the S900 immensely popular, similar to the success of the Yamaha DX7.

The treatments offered by a modular synthesizer, like filterings and distortions, can be applied to any sound source, no matter if it an internal sound source, another synthesizer, a sound recording or a sampler. In practice modular synthesizers and samplers turned out to be ideal companions, as the modular synthesizer can be used as the source to be sampled, and at play time the modular can be used to manupulate samples made earlier. The combination of sampler and modular synthesizer is in fact very similar to the electronic studio setup that was developed in the fifties of the last century. The sampler being akin to the taperecorder and the modular synthesizer to the equipment that surrounded the taperecorder.

A variation on the sampler is the sample-based drumcomputer. Shortly after the release of the Emulator Emu uses the Emulator technique to design the Drumulator. At about the same time Roger Linn releases the Linndrum. Both instruments use recordings of real drums, making the sound very convincing. Recording real drums in a recording studio is a tedious process, first of all the drums have to be mic'ed up and then the right mixing balance and sound has to be found. This takes lots of expensive studiotime. The drumcomputer in contrast can be plugged in directly and it is go. Active drumpads were developed that could be plugged in at the back of a drumcomputer or through a 'drum to MIDI' converter box. These drumpads allow drummers to play the drumcomputer like a familiar drumkit. Another feature of the drumcomputer is that patterns can be recorded or preprogrammed as MIDI information and arranged in songs. The Linndrum concept is later bought by Akai and is still available in their popular MPC range of products.

Sampling and drum programming has had an enormous influence and initiated new styles of music. Still, the techniques employed with a sampler are in essence the same as the tape techniques developed in the fifties and sixties electronic studio's, though the genres of music they are used for are now quite different. However, using a sampler is much more convenient and straightforward than using a taperecorder, accounting for the sampler's immense and ever growing popularity. Today's laptop computers gradually overtake the territory samplers have claimed for some two decades, but this is only because a laptop with the appropriate software is itself also a sampler, it does in essence the same thing with the same technology. Still, reliable hardware samplers like the Akai MPC range offer extra play controllers like drumpads and knobs to instantly tweak the sounds, making them ideal to use during live performance. And the samplers' musician friendly and dedicated operating systems will undoubtedly keep them from extinction for some time to come.

Digital effect units

Many treatments are based on manipulations of time delays or time displacements. Well known effects are the creation of echo and reverberation. Techniques that use a cyclic digital memory and a DSP to read and write signals from and to this memory allow the creation of high quality and natural sounding time displacement treatments. Echo, reverberation and related effects are popular with all musicians, so they appear in separate boxes that can be used by synthesizer players, guitar players, vocalists, etc. These days most synthesizers have an effect unit built in, although these are generally not of the same quality as the high end studio devices.

Next to effects based on time displacements, digital effect units can offer filterings like multiband filters, equalizers and vocoders. Some can also manipulate the dynamics of audio signals like applying compression, and others can even correct pitches of vocals.

The cheaper effects units may have some hundred or more types of effects, but only one effect can be used at the same time and most of its parameters are fixed. The more expensive units developed for professional studio use can have several different effects working together and the parameters can be finetuned. On high end equipment effects can be freely chained in any order, much like the patching on a modular synthesizer.

Synthesizers and digital effect units are commonly used together, the synthesizer creating the timbres and the effect unit creating the spatial effect by placing the sound in an acoustic space of a certain characteristic. The main difference in manipulation is that a synthesizer works on the level of individual notes and single voices, while the effect unit works on the total of the sound.

Basic principles of sound synthesis

The three parameters of sound

The character of a sound is controlled by the three distinct properties pitch, loudness and timbre. These are named the three basic parameters of a sound. All three are dynamic in nature, changing and developing gradually over the time the sound is heard. So, a distinct sound is characterized by how pitch, loudness and timbre each develop over time. The musician or composer controls how these developments will be by either dynamically and expressively playing the parameters or describing their temporal developments in a score on paper, a computer file or even a computer program.

Whenever a sound is heard there will always be sensations of pitch, loudness and timbre. Additionally a sound has a certain starting point and a certain end point in time, formally the time between two adjacent periods of zero loudness, giving a certain duration to the sound. Some composers name sound duration a fourth parameter of sound. But as the sound duration is already implicit in the description of how the loudness of the sound develops over time, this parameter can be discarded when the developments of the three basic parameters are described well enough. Of course this is of much more concern to composers, who have to somehow describe sounds in a score, than to a musician who simply wants to play the sound.

A musical sound (which is just any sound that is used in a piece of music) doesn't necessarily need to have the distinct single pitch of a single piano or organ note. There can be more pitched components in a sound, like in a chord. Additionally, these pitched components don't necessarily have to have a harmonic relationship, just think of the 'enharmonic' sounds of certain drums and percussive instruments. In this class of sounds there still can be one pitched component that is perceived as the dominant pitch, enabling the sound to be tuned to other sounds. An example of such a sound is the sound of a timpani drum.

Another class of sounds is named the pitchless sounds, like the sound of falling rain or ocean waves. In fact pitchless sounds are an assembly of many pitched components, but there are so many components that the human ear cannot perceive their distinct pitches any more. The components melt into one single 'pitchless' sensation. And although there is no sense of a definite pitch in pitchless sounds, there can be a strong sense of very characteristic timbres. Noise is the textbook example of a pitchless sound. In nature noise is created by processes that are in essence chaotic, meaning that there is a definite order in the process, but the order is too complex for the human mind, it cannot detect the order and will even have a hard time to concentrate on one of the individual components. Think of a waterfall and how the turbulence of the falling water creates the sound. It is impossible to follow all the terbulent water movements by eye or ear, but there is a definite physical reason why the water at a certain moment falls and sounds like it does.

In noise there are so much pitched components present that each of them literally drowns in the overall sound. Still, this noise might be produced by a well defined and in essence simple and orderly executed process. Noise might appear random, but that doesn't mean it is really random, in fact there might be all sorts of statistical properties in the noise. Later on it will be shown how these statistical properties can affect the many possible timbres of noise based sounds.

An interesting moment is when something that is perceived as an orderly process develops towards the state where it is perceived as chaotic. In the moment of the transition from order to chaos the human mind can still concentrate on a single component of the sound, but the mind will start trying to skip from component to component. An interesting example can be heard once a year when visiting the Frankfurter Musikmesse, where one exposition hall is exclusively dedicated to the grand piano. There might be a hundred or so piano's on display, ready to be played by potential buyers. On the busiest moments at the fair all piano's are being played. Of course this creates a huge and totally chaotic sound, aptly named a cacophony. By taking a strategic position somewhere in the hall and listening to the overall sound there is these short moments one suddenly recognizes a bit of Beethoven in the cacophony, immediately dissolving into some ragtime and then dissolving into cacophony again. It is virtually impossible to catch and hold on to the moment when something is recognized in the cacophony.

When a sound is heard it will always give a distinct sensation of timbre. Timbre plays an important role in recognizing the sound. The synthesizer is specifically designed to be able to generate a vast range of timbres. Timbre as a phenomenon is created by a collection of partials, similar to how molecules are created by a collection of atoms. In the nineteenth century the physicist Helmholz has proved that a singular pitched sound has a series of possible partials. If these partials are harmonically related they are named harmonics or overtones. All natural sounds have some or more partials. Only by electronic means can a sound be created that consists of only one single partial, the one that is named the fundamental. The waveform that creates this sound is named a sinewave. As this sound has no extra partials to give it a timbre, it can be said that the sound of a sinewave has no timbre, similar to saying that distilled water has no taste.

Working on the timbre of a sound is the most laborious part of sound design. Human hearing is incredibly sensitive to the most subtle changes in timbre. Additionally there is the tendency to adhere some association or sense of meaning to the intonation of sounds. The same sentence of spoken words can change from a question to a command by only changing the intonation, e.g. by slightly changing the pitch development in the words. In certain circumstances timbral effects are used to work on the human emotion. Examples are religious music, shamanistic incantations, and the like. Psycho-acoustics might also play an important role, especially when a sense of spaciousness is required. Another important aspect of timbre is legibility, or how easy it is to isolate the sound in between other sounds, in order for the mind to recognize it and give it some meaning. Some aspects in timbre have the ability to mask away aspects in other sounds, reducing their legibility. This is of great importance during the mastering process of a music recording when the mastertape is made which will be used as the source for submitting the music to vinyl or a CD. In the mastermix it might turn out that instruments conflict with each other, reducing each others legibility or presence. The regular approach is to use compressors and equalization functions on the mixing desk to improve the mix. However, it is common sense to think things out before initial recordings are being made, so these conflicts in legibility occur to a much lesser extend. A good orchestration or arrangement for a piece of music can emphasize the melodic or timbral structures by a well balanced choice of sounds that do not mask each other away, but instead tend to emphasize each other musically.

Loudness

Loudness is how an individual perceives the volume of a sound at a certain sound pressure level or SPL. This perception can differ from person to person, as not everybody has the same sensitivity for different registers in the audio range. Also, a sound might be so low in volume that the ear doesn't perceive it any more, while a measurement device would still prove it present. The point where the volume is so low that the ear ceases to hear the sound is named the threshold of audibility. This threshold differs for person to person and for different pitches. In general the threshold for the higher pitches is raised when a person is getting older, until finally deafness for this pitch range occurs. Note that when the threshold is raised it means that a louder volume is needed for the sound to be heard. Like the threshold of audibility is the lower limit of the hearing range, there is also sort of an upper limit to this range. When the sound pressure exceeds this limit a severe pain in the ears is felt. Increments in volume are no longer heard above this level, as it is overruled by the pain sensation getting worse. Excessive sound levels can permanently damage one's ears, resulting in deafness. Very loud volume levels that do not yet induce pain in the ears will also cause deafness over time. So, great care must be taken with loud volume levels, damage to the ears caused by loud levels is permanent and a great disaster for a musician, compare it to when a painter gets blind. It is very wise to protect one's ears during performing live to avoid possible future hearing damage. Headphones can also produce a lot of sound pressure on the ear, which may result in ear damage as well. Never take any risk with loud volumes. It is better to keep a comfortable volume level, not too soft but also not too loud. At a comfortable volume level one can also hear the most detail in sounds. In general a good volume level for the studio is when the sound can be easily heard and only a slight concentration is needed to hear a specific detail in the sound. When the volume level is either too low or too loud more concentration is needed. Find a balanced loudness level that feels comfortable and doesn't tire after some hours.

Handling volume levels requires experience, as several psycho acoustic principles are involved and it takes getting used to hearing these principles at work. When designing sounds that need to be played expressively it is important to keep these principles in mind. Psycho acoustics is a science by itself and it goes beyond this book to delve into the details. Still, some basic examples will be given in the following paragraph, as these examples are many times the key to a good sound.

The human mind has the capacity to mute sounds from the surroundings. These ambient sounds are registered by the ear, but they don't enter into the awareness. This has to do with how the mind concentrates on what it is doing. Sudden changes in volume in these sounds will attract the attention. Another effect is that when the volume reaches a certain loudness level people are forced to listen to it. But this doesn't mean that they listen attentively. Softer loudness levels are better in keeping the focused on the sound, as a little more effort is required to listen to the sound. It is advisable to keep the volume at a reasonable level in a small or medium sized listening room or when performing at a small club.

There has been research on how loud certain frequencies have to be in order to be perceived at a uniform loudness level by the mind. So, how the SPL for a given frequency range has to change to be perceived equally loud as another frequency range.This can be plotted in a curve that is named an equal loudness curve. This curve is different from person to person, but when a lot of individual curves are averaged a curve results that is named the Fletcher-Munson curve. This curve reveals that in general the human ear is the most sensitive to volume changes in the 2kHz to 5kHz region, while being only moderately sensitive to volume changes in the bass region. This curve can have an impact on a mix, as when the mix is made at a relatively low loudness level, the 2kHz to 5kHz region might sound too loud when the recorded mix is played back at a much higher loudness level, making the overall sound a bit squelchy. 33-band graphic equalizers can be used to compensate for this effect when such a mix is played for an audience in a hall. Another issue that the Fletcher-Munson indicates is that the ear is not sensitive for expressive loudness changes in bass sounds. In many instances the average volume level of bass sounds is kept constant in time by using heavy compression of the average level on the bass sounds in a mix. Another issue is that when a keyboard sound is designed to be sensitive to the velocity at which the keys are struck, it is an oversimplification to have this velocity value just control the overall volume for that key. Instead it will sound more natural when mainly the 2kHz to 5kHz region is affected, while the region below 400Hz and above 6kHz are affected only slightly or not at all by the velocity value. This can be accomplished by e.g. letting the velocity value affect the curve of an equalizer module.

Dynamic range

The difference between the softest and the loudest perceivable volume levels is named the dynamic range of the ear. The softest level is the treshold of hearing while the loudest level is the treshold of pain when the sound level becomes unbearable. The dynamic range for the human ear is remarkably large, about one in a billion. This range can be set out on a base 10 logarithmic scale, resulting in 12 subdivisions expressed as twelve Bell. Each Bell is divided in ten deciBell, decibel or dB. Consequently it follows that the dynamic range for the ear of the average human being is about 120dB. When the volume is raised by about ten dB the perceived loudness is doubled. This fact is quite subjective, as perception itself can only be measured what persons subjected to a test report to have witnessed. When amplification of a signal is concerned a raise in level by 6 dB is equal to an amplification of exactly two times.

Amplitude

When the volume knob on an amplifier is fully closed there will be no sound in the room, but there may very well be a signal at a certain level present on the input of the amplifier. As loudness is a subjective value that also changes from person to person, it cannot be used as a parameter to express the level of the electric signal at the input of the amplifier. Instead amplitude is used to express a signal level. Electrical audio signals have an electric polarity that alternates between positive and negative voltage levels at audio frequency rates. Amplitude is in practice the amount of voltage swing between the positive and negative peak levels in the electrical signal. There are two common ways to plot the amplitude as a curve over time, one method uses the absolute values of the peak values in the swing and connects a line between these peaks, the other method takes the average signal power in a certain time frame. In a synthesizer both ways of looking at amplitude are used. Using the absolute peak values is important to prevent sounds from exceeding the maximum limits the circuitry can handle, which could result in clipping of the tops of the signal peaks. This is especially important with digital equipment, where clipping is instantly and can sound pretty severe. In contrast, analog equipment has in general a range where the signal gradually saturates before it clips and the audible effect of clipping is less severe than with digital equipment, though the momentary distortion is still very audible. Working with the average power value instead of the peak values is useful when balancing the signal levels of two or more sound sources against each other in a mix.

The loudness contour and amplitude envelope

The curve that connects the peaks of the absolute values of the alternating signal is named the amplitude envelope and it describes exactly the loudness contour or how the loudness of the sound develops over time. When looking at a single, isolated sound, like a single beat on a drum, this sound will have both a distinct start point and a distinct end point in time. At the start point the amplitude is zero but will rise very quickly to a certain level. Then the amplitude will decay slowly until it reaches zero again. This can be plotted in a curve, where the elapsed time since the starting point is plotted on the horizontal axis and on the vertical axis the amplitude at a certain point in time is shown. Such a plot is simply referred to as the envelope of a sound. To get a bit more grip on this envelope the curve is subdivided in those sections where the amplitude value either increases or decreases. These sections are generally named by using single alphabetic characters.

The first part of the amplitude envelope of the earlier mentioned drum sound is named the attack phase and is denoted with the character A. In a drum sound the attack phase will be relatively short. Immediately after the amplitude envelope has reached it's highest level the amplitude will start to decay. This section is named the decay phase, denoted with the character D. This type of envelope with only an attack and a decay phase is named an AD envelope. Many instruments that are struck like drums or plucked like a harp exhibit this type of envelope.

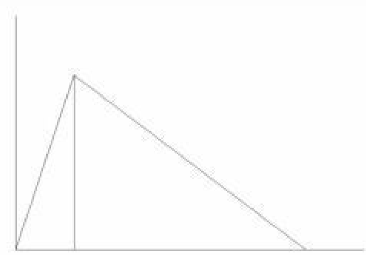


Figure 1 - AD envelope

To describe an AD envelope it is enough to describe either the angles of the attack and decay slopes or how long the attack and decay phases last. Using time values to describe the attack and decay durations is more convenient and the method used on many different brands of synthesizers. So, an AD envelope of a percussive sound can be sufficiently described by saying that it has an attack time of e.g. 5 milliseconds and a decay time of 1500 milliseconds.

When a note is played on a wind instrument, the amplitude will raise fairly quickly, be stable while the note is sustained and then quickly decay after playing is stopped. There is an extra section between the attack and decay phase. This stable phase is named the hold phase, denoted with the character H. This type of amplitude envelope is named an AHD envelope. The AHD envelope is most common with wind instruments, bowed string instruments and pipe organs. Note that the length of the hold phase depends totally on how long the note is held by the musician, in theory it could hold forever.

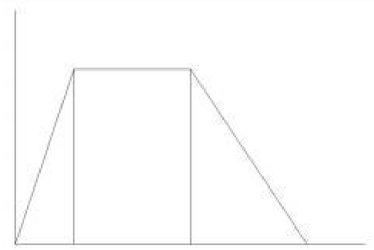


Figure 2 - AHD envelope

With instruments like the piano there are in fact two envelopes that work together to create the final envelope of the sound. The first envelope is defined by the hammer striking the strings and the following vibration of the strings. The second envelope is defined by the interaction between the strings and the sound board and resonance box of the piano. The hammering action has a short attack and a relatively long decay phase and so follows an AD envelope. During this AD envelope the kinetic energy of the vibrating strings is transferred to the sound board and resonance box where this energy builds up strong resonances. The amplitude development of these resonances follows roughly an AHD envelope, the sonic energy lingering in the sound board and resonance box during the hold phase, only starting to decay when the strings are damped when the key is released. The sustain level during the hold phase is lower than the peak level of the AD envelope of the hammering action, as the kinetic energy of the string vibrations also leaks away into the air.

When these two envelopes are joined in one graph it shows an envelope with four phases. In the first phase, when the hammer hits the strings, the overall amplitude will raise quickly and is again named the attack phase or A. Then the amplitude of the hammering action will decay while building up the resonances in the sound board, until it more or less equals the sustain level of the AHD envelope of the vibrating strings/sound board/resonance box combination. This is the decay phase or D. Then the vibrating strings/sound board/resonance box combination will sustain the sound, this phase is named the sustain and denoted with the character S. Finally, when the strings are damped on the release of the key the sound decays quickly, this phase is named the release phase denoted by the character R. This type of envelope is named an ADSR envelope.

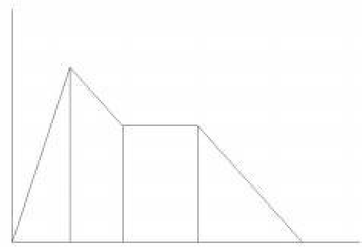


Figure 3 - ADSR envelope

The advantage of an ADSR envelope over an AD envelope is that it allows for the intentional dampening of the sound on a moment chosen by the musician, giving simple and instant control over the note length. The musical difference between the ADSR and the AHD envelope is that the amplitude during the hold phase of the AHD envelope is equal to the maximum amplitude that was reached at the end of the attack phase. In contrast, the sustain level of an ADSR envelope can be significantly lower than the peak of the attack. The ADSR envelope is in fact designed to mimic the mechanics that happen in instruments with a sound board and/or a resonance box. Such an instrument can be seen as having a resonating body and an excitation function, like the piano strings/hammer combination. The excitation function fills up with energy on the moment the sound starts and this energy is then transferred to the resonating body. When a hammering or plucking action is used to initially generate the energy, there is almost instantly a lot of energy available. Then this energy will flow slowly from the excitation function to the resonating body, building up and sustaining the resonance. Right after the attack phase a lot of the released energy will be used to quickly build up the resonance. The decay phase is actually the time needed to build up this resonance. After the resonance is built up only moderate amounts of energy are needed to sustain the resonance, causing only a minor decay in the amplitude level. When the excitation function is stopped, e.g. by dampening the strings in the piano, there is no more energy flow from the excitation function into the resonant body and the resonance will die out rather quickly. This means that the release time is actually the natural reverberation time of the resonant body.

The AD, AHD and ADSR envelopes are well suited to emulate the envelopes of real world percussive instruments, blown and bowed instruments or struck and plucked instruments. But there are many more sounds that have a much more complex amplitude envelope development, a clear example being human speech. To emulate complex amplitude envelopes multi stage envelopes are used. In a multi stage envelope there are several segments that can be increasing, decreasing or stable in level. Two methods are used to describe such an envelope. The first method records the actual amplitude level when the curve changes direction and the time when such a change takes place. The second method records the final amplitude level of a segment and the angle of increase or decrease of the segment, named the rate. When the curve reaches the final level of the current segment it starts to increase or decrease with a new rate to the final level of the next segment. Multi stage envelopes can theoretically have any number of segments, but on most synthesizers they tend to be limited to five or six stages.

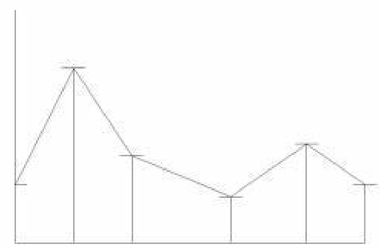


Figure 4 - Multi stage envelope

A modular synthesizer will have modules that can generate an electrical control voltage signal that will exactly follow one of the described envelope curves. Such a module is named an envelope generator. An envelope generator module will have an input that can receive a trigger signal that will start the curve at the beginning of the attack phase. This trigger signal marks the start point of a sound in time. When the trigger input of such an envelope generator is connected to a keyboard key trigger signal, a switch or a drum pad the musician can instantly start the envelope. But the trigger signal used to start the envelope can also originate from a module that can generate a train of trigger pulses in some rhythm or from a computer, or any other device that can generate a compatible trigger signal. By itself an envelope generator will do nothing, it always needs a trigger signal as a command to start the envelope. And when the envelope has fully decayed it will meekly wait doing nothing, until another trigger command is given.

Pitch and frequency

On a instrument each note has a distinct pitch. The pitch depends on how many vibrations per second are present in the played note. The number of vibrations per second is named the frequency. In other words, frequency is how many occurrences of repeating vibrations or cycles of a certain waveform happen during a second of time. Frequency is expressed in Hertz or Hz. These days it is custom to tune instruments to the note A that has a frequency of 440 Hz, meaning that this note makes the air pressure vibrate at a rate of 440 times a second. The lowest number of air pressure vibrations the average human ear can pick up has a frequency of about 20 Hz. The highest number can be as high as 20.000 vibrations per second, a frequency of 20 kHz (kilo Hertz).

Like electrical devices internally deal with amplitude they also deal with frequency, while pitch deals more with how the human mind perceives frequencies. Notably frequency differences between the notes on musical scales. A note with a frequency that is twice as high as the frequency of another note will sound twice as high. E.g. a note at 440 Hz will sound twice as high as a note at 220 Hz. This is interval is named an octave. A note with a frequency which is three times as high, so at 660 Hz, will not sound two octaves, but only 'one octave and a fifth' higher. The note that will sound two octaves higher as the 220 Hz note will in fact have a frequency of 880 Hz, meaning that this frequency, which is perceived 'three times as high' as the 220 Hz frequency note, has an actual frequency value that is four times higher. And a note that sounds four times higher will have an actual frequency of 1760 Hz, which is eight times higher. The explanation is that a note that sounds exactly one octave higher must have it's actual frequency value doubled. Look at the frequency values for all A notes over an eight octaves range:

55 Hz - 110 Hz - 220 Hz - 440 Hz - 880 Hz - 1760 Hz - 3520 Hz - 7040 Hz

This shows that the relation between the notes on the well tempered twelve note musical scale and their actual frequency values is exponential. This is very important to realize, as it might lead to confusion. On modular synthesizers pitches can be controlled either through their corresponding notes on the exponential musical scale, or through the exact frequency values on a linear frequency scale. Only few modular systems offer both methods. For musicians wanting to play in the western well tempered twelve note scale the exponential method is the most convenient, as it translates directly to the black and white keys on a keyboard. This method is also named the Volt/Octave norm. But for sound synthesis the linear method has some very useful features. Meaning that for the more experimental composers and sound designer artists this linear method, also named the Volt/Hertz norm, might have interesting advantages.

Monophony and polyphony

Some musical instruments can only produce one note at a time, these instruments are named monophonic instruments. Examples are the silver flute, the trumpet, etc. Other instruments allow for many notes to be played at the same time, like the piano, the organ, etc. These are the polyphonic instruments. Polyphonic instruments can play both single notes and chords. A chord is a layering of several notes in a certain musically pleasing relation. One of the pitches in the chord can appear to dominate over the others. This note is named the key or root of the chord. The other pitches have relatively easy frequency ratios with this root pitch. These frequency ratio's might be 3:2, 4:3, 5:3, 5:4, etc.

In the more experimental electronic music genres chords with more complex and exotic ratio's than those used in traditional western music are often used to create rich sounding sonic textures. To avoid confusion with the traditional chords and their traditional names it is better to use the name composite sounds for these sonic textures. These textures are often used in experimental electronic music, soundscapes, drone music, film music, etc. In these musical genres it might be the changes in timbres that define the development of the composition. Melody, harmony and rhythm are made subordinate to these timbral developments. E.g. rhythm might be created by rhythmic changes in timbre. Composers have the freedom to work out their own personal system of composing and sound synthesis can be an important part of that system. There is a choice of synthesis systems and material can be intuitively assembled 'by ear'. Without doubt it takes a lot of experience with sound synthesis to make such efforts musically worthwhile.

Traditional music notation is not very useful for the compositions that involves the notation of the development of the sonic developments in the sound synthesis. Pitches and tuning can be freely defined and are difficult to express in traditional notation. many contemporary composers have experimented with new ways of notation and the resulting scores sometimes look more like paintings than like a traditional score.

The frequencies of some of the partials in a single pitched sound might coincide with the intervals found in the traditional chords. E.g. an interval of an octave and a fifth is related with the third harmonic, making a fifth also related to that third harmonic, as it happens that the second harmonic of a fifth will coincide with the third harmonic of the root. But a monophonic sound can have something like up to a hundred harmonics present within the hearing range. And there are many possible relations that can not be simply expressed by chord intervals. To define the relation between a root frequency and a second frequency the frequency ratio is used. This ratio can be expressed in a fractional number containing a numerator and a denominator, notated like n:d. When the ratio is 3:2 then the second frequency is 3/2 or 1.5 times higher. This system was already used in ancient cultures to define musical scales, an well known example is the Pythagorean scale.

The harmonics of a fundamental frequency always have a ratio of n:1, where n can be any positive integer number. Partials do not necessarily need to have a simple ratio to the fundamental, many drum sounds are good examples. These are sounds that can have non harmonic partials present, which still melt nicely into the overall drum sound.

Chords sound best in just tuning, in just tuning exact and simple ratio's are used to define the scale. Many synthesizers offer the possibility to use both well tempered scales with a user definable amount of notes in an octave and a number of just tuning scales.

On a traditional keyboard the keys for the notes in a chord need to be played at once. Modular synthesizers offer features to 'preprogram' chords and composite sounds under single keys. Defining different composite sounds which are tuned to exact ratio's under different keys, allows for the play of complete soundscapes in just tuning.

When the amount of partials is increased and several non harmonic partials are added the sense of pitch might be lost. Formally the sound becomes noise, but noise can have an infinite amount of different timbres. And sounds that are definitely noise can generate a sense of pitch, like the whistling of the wind.

A sound can have a single pitch, be with or without harmonic or non harmonic partials, be the layering of some pitches like a chord, a composite sound, a complete soundscape and finally up to completely pitchless like the sound of ocean waves. While the sound sounds the pitch or pitches can glide, vibrate or jump. This is named the pitch envelope. It is important to have very precise control over the pitch envelope, as unlike the amplitude envelope the pitch envelope doesn't follow simple graphs. It is best to bend the pitch by hand, to give the sound the right intonation. A device named a pitch bender allows for expressive manual control. Most common pitch benders are the pitchbend wheel, the pitch stick and the ribbon controller.

Timbre

Timbre is the sonic quality of a sound that defines the distinct character of this particular sound and makes it recognizable amongst other sounds. When a trumpet player and a violin player play the same note with exactly the same loudness contour and pitch bend, the difference in timbre is still clear and hardly anyone will have a problem in recognizing the sound of the trumpet from the sound of the violin. But there is more than recognition to a timbre, there are additional musical properties to the timbre of a particular sound. These properties are often very subjective. Vague names are used to classify their sonic effect, like a timbre can be damp or bright, muddy or squelchy, woody or metallic, singular voiced or chorused, thin or fat, massive and impressive, soft or aggressive, warm or cold, deep and spaced or right into the face, etc. But before these kinds of subjective qualifications can be dealt with there must be an understanding on how the basic timbre of a sound comes about. In sound synthesis there are a number of different techniques to create certain timbres. The simplest technique is to make a digital recording of a sound of a particular instrument, commonly named a sound sample. The sound sample can be played back at a different pitch and one of the first things one notices is that the timbre changes in an unnatural way when the sample is played back only just a few notes higher or lower. And when the detuning is more than an octave the sound is hardly recognizable any more. This means that there is no simple relation between the pitch and loudness contour and the timbre of the sound of an acoustic instrument. In general the overall loudness contour is the same for each pitch, although initial segments of the loudness contour, like the initial attack and decay phase, might be shorter for higher pitches. When playing different notes on an acoustic instrument much more complex things seem to happen. For one there are some fixed frequency ranges that seem to be present in all notes and the relative strength of these ranges seem remain pretty constant no matter how much the pitch changes. Instead these frequency bands seem to be much more influenced by how loud the note is played, a good example is a muted trumpet. Additionally the playing style of the instrument can change the timbre in sometimes dramatic ways. This means that timbre can not be captured with one single parameter, like the frequency parameter or the amplitude parameter. In fact, there are many parameters that define the timbre of a sound. So, while a sound still has the three basic parameters loudness, pitch and timbre, the loudness on a certain moment can be defined by only one amplitude value, the pitch can be defined by one or more values, e.g. for a chord there might be three frequency values, while for timbre there might be a whole array of values needed to describe the sound. So, what was named up to now a basic parameter of sound is not simply one single value, but in practice a collection of values, used together to define a generalized parameter like ‘a trumpet sound’.

The exciter/resonator model.

To gain some more insight it often pays to simplify the situation into a simple model. A very useful model for acoustic instruments is the exciter/resonator model. In this model the instrument is roughly divided into two parts and the interaction of these two parts with each other is responsible for the resulting timbre of the instrument. This model is able to describe in a simplified way what happens in most acoustic instruments. A very good example is an acoustic guitar, where a string is used to make the body of the guitar vibrate. The string acts as the exciter and the guitar body resonance box as the resonator. The sound of only the string itself is not loud enough to be musically useful and the resonance box is used to amplify the sound. Additionally, the resonance box shapes the timbre of the sound. This model immediately explains why a sampled sound starts to sound unnatural when detuned to a new pitch, as the resonant guitar body does not change for a new pitch. So, the timbre for each note in a real world instrument is defined by how the resonant body or resonator interacts with an excitation at a certain pitch.

The VCO-VCF-VCA model synthesizer

The traditional analog synthesizer tries to simulate this exciter/resonator model by using two separate modules that act as an excitation function and a resonator. For the excitation function an electronic sound source, named an oscillator, is used. The oscillator module is similar to the strings, reeds, etc., of acoustic instruments In its effect an oscillator provides a train of steadily repeating pulses on its output, the number of pulses per second defining the frequency. A single pulse is named a cycle and the cycle can have various forms, named the waveform. The resonating body is simulated by the use of various types of resonating filters. The sonic energy in the signal from the oscillator cannot leak away in the air in the form of sound or warmth like in an acoustic instrument, instead the flow of sonic energy is continuous when the oscillator is connected directly to the filter. As a result a synthesizer can create steady pitches with resonance effects that can sound forever. In order to create natural swells and decays an extra set of controllable amplifiers must be used to control the overall volume development of the sound. These amplifiers can be controlled by devices that generate a control signal which follows the envelope curves as described in a previous chapter. When designing sounds it is useful not only to think in electrical signals that flow from module to module, but also in terms of sonic energy that excites another module, where the energy is 'transformed' into timbre. Like how the sonic energy from the oscillator is actually exciting the filter in a similar way as a guitar string is exciting the body of the guitar.

When the exciter/resonator model is patched on a modular synthesizer, there are three modules chained in a serial way, meaning that their respective outputs will go into the input of the next module in the chain. The first module is the oscillator and its output goes into the input of the second module, the filter. Then the output of the filter goes into a third module, a controllable amplifier which is responsible for the volume envelope. The general notion is that in this model the oscillator module defines the pitch parameter, the filter defines the timbre parameter and the controllable amplifier defines the amplitude parameter. This is almost true, as the timbre parameter is actually defined by how the filter reacts on the oscillator, as in fact the timbre is created by the cooperation between the oscillator and the filter. Different waveforms for the cycles of the oscillator will excite the same filter in different ways, creating different sonic effects. So instead, one can think in terms of how the exciter/oscillator is exciting the resonator/filter and the stream of continuous sound this process creates is controlled in amplitude by the controllable amplifier. Later on in this book the advantage of thinking in this more correct way will become clear when looking at the synthesis of certain sounds in more practical detail.

The three modules, oscillator, filter and controllable amplifier, each get their own separate control signals to be able to dynamically shape the sound. A module can receive more than one control signal, e.g. the oscillator can receive a control signal defining the pitch of the note it has to play, but additionally receive an extra, slowly varying, control signal to give a vibrato effect to the pitch. On the analog systems of the past, where the control signals were actually voltage levels, the modules were named Voltage Controlled Oscillator, Voltage Controlled Filter and Voltage Controlled Amplifier, abbreviated to VCO, VCF and VCA. Which is why this model is still referred to as the VCO-VCF-VCA model, although digital system do not work with discrete voltage levels anymore.

Picture of the schematic!!!

Playing style

The basic VCO-VCF-VCA patch has the advantage that it can mimic the dynamics that happen in an acoustic instrument through the control inputs on the modules. But it is in fact very hard to convincingly imitate an existing acoustic instrument with the model. In general the synthesizer is not really very interesting to imitate existing instruments, instead it is mostly used to create totally new musical sounds, that can be played with the same sort of dynamics and sonic characteristics of a certain acoustic instrument. Playing style is very important here, e.g. when a synthesized sound that very vaguely reminds of a flute is played with a flute-like playing style, the human mind will have the impression of a flute, though maybe a cheap flute. But when a very close imitation of a flute sound is synthesized and played in a polyphonic way like an organ is played, it will sound much more like an organ that like a flute. It is very important to realize that playing style is as important as synthesizing a certain timbre to create the effect of a certain existing instrument.

Sound imitation

In the music industry there is a commercial need for convincing electronic imitations of real world acoustic musical instruments. When in a recording studio a string section has to be recorded, it is much cheaper to use an electronic instrument than to hire a couple of musicians for a few days. Since the early seventies studio’s tried to use VCO-VCF-VCA model synthesizers to replace real musicians. This led to a common but false believe that the main purpose of these synthesizers is to imitate existing instruments. In fact, imitation is their weakest point. It is a much healthier approach to see a synthesizer as an instrument by itself, with its own musical right of existence and use it as such. In the eighties samplers replaced the original VCO-VCF-VCA model synthesizers in the studio, as when using the right set of samples, samplers are much more convincing in imitating acoustic instruments. Just think about digital piano’s, these are in fact preprogrammed samplers with in general several samples for every single key. For recording purposes these digital piano’s do perform very well. Still, samplers lack the kind of dynamic timbral control that the VCO-VCF-VCA model synthesizers have. So, when it is about imitating acoustic instruments, samplers have the realism in the timbre, but lack the dynamics. In contrast, the VCO-VCF-VCA model has the dynamics, but in general lacks realism in the timbre of imitated acoustic instruments.

The waveshaping model

To overcome the limitations of both the sampler and the VCO-VCF-VCA model, there have been attempts to use methods that try to directly synthesize the audio signal of the timbre without using resonant filters. In these techniques only oscillators are used, but special types with a dynamically controllable variable waveform. While the sound develops, the waveform is dynamically reshaped in a way that the resulting timbre follows the timbral development of the instrument to be imitated as close as possible. This technique is named waveshaping. Waveshaping takes a basic waveform and then distorts this waveform by a distortion function. There are two subclasses of waveshaping techniques. Techniques in the first class distort the amplitude of the waveform at audio rates, techniques in the other class distort the frequency of the waveform, also at audio rate. To understand the difference and realize why there are only two subclasses, note that any momentary waveform can be drawn as a two dimensional graph on a piece of paper. When doing so it becomes instantly clear that there can be a distortion in the vertical direction, which is the amplitude value, or a distortion in the horizontal direction, which is the time axis. And time of course relates to frequency. Distortions in these two possible directions are named amplitude modulation in the audio range or AM and frequency modulation in the audio range or FM. A variation on FM is where it is not the actual frequency parameter that is heavily modulated with an audio rate signal, but instead the phase of the waveform is modulated. This is properly named phase modulation or PM. PM is a 'digital only' technique and offers a small advantage over FM as it allows for feedback modulation or self modulation of the waveform oscillator without altering the pitch of the signal. For the rest everything that applies to FM also applies to PM. When AM, FM or PM techniques are used in a synthesizer the basis is in general a digital sinewave oscillator. Some types of waveshaping synthesizers, like the Yamaha DX-type synthesizers, use only the phase modulation technique and are commonly (but wrongly) named FM synthesizers. On the better traditional analog modular synthesizers both AM and FM is possible, but the frequency stability of the analog oscillators is not enough to precisely use the technique to do convincing imitations. A digital modular synthesizer like the Nord Modular G2 is able to do AM, FM and PM waveshaping and also combine these techniques together to create expressive timbral developments in a straightforward and intuitive way. Additionally, on the G2 resonating filters can be inserted in the modulation path to patch models with very powerful sonic characteristics.

One drawback of the waveshaping techniques is that, when imitation is the goal, complex mathematics is involved in calculating the exact depths of modulation to create the wanted timbral developments of certain sounds. With the AM technique this involves Chebyshev functions and the FM technique involves Bessel functions. Interestingly the commercially most successful synthesizer up to now, the Yamaha DX7, uses the PM technique. Still, programming sounds on the DX7 was very difficult, only very few people had any notion on how to go about. Luckily the DX7 came with a useful set of factory preset sounds, and the commercial success of the DX7 has more to do with the fact that it came at the right moment in time and with the right amount of polyphony and the right size, weight and price. The complex DX synthesis method with its six phase modulatable sinewave oscillators per voice was happily ignored by the average musician.

One important notion is that waveshaping using AM and FM techniques relate to each other in a very close way. One might be tempted to think that amplitude modulation cannot change the pitch, but in fact amplitude modulation of certain waveforms by other certain waveforms can actually create a fixed detuning of the pitch. Just like frequency modulation can keep the pitch fixed while only affecting the timbre of the sound. So, both are really related. But it goes way beyond the purpose of this book to go deep into the basically mathematical subject of proving this relation in a scientific way. Later on in this book there will be some simple practical recipes to use these properties of AM, FM and PM to create various musically interesting sonic effects.

In practice there is one important difference between AM and FM. AM can always be done after the output of the oscillator with a controllable amplifier module, there need not be a specific input or function on the oscillator to be able to use AM. The controllable amplifier however must be able to accept bipolar input signals. A standard VCA module has a bipolar input for the audio signal, but the input for the amplitude level is unipolar. Whenever the control signal on this input becomes a negative value the VCA simply shuts off. However, when a module named a ringmodulator is present on the system, this module can be put to good use instead of a VCA, as this ringmodulator is technically a bipolar controllable amplifier that can handle both positive and negative signals on both its inputs. When a lot of ringmodulator modules plus additional mixer modules are available, Chebyshev functions can be patched to do the timbral shaping, using one single sinewave as the initial waveform.

When using the FM technique for waveshaping purposes a special FM input on the oscillator is needed. This FM input must be able to control the frequency in a linear fashion, the standard pitch input with its exponential V/Oct control curve is less useful, as it will quickly detune the pitch.

Timbre and acoustic instruments

The difference in timbre between acoustic instruments depends on a lot of factors, for instance the dimensions and materials of the instrument body and whether it uses strings, skins, reeds, etc. to be excited. Even ambient temperature, air pressure and dampness of the air can have an influence on the timbre. Additionally, variations in playing style can create different timbres from the same instrument. And as there are so many different types of acoustic instruments, it is hard to generalize on how their timbres are created. The resonant body can be a fife, like with a flute, where it is air that resonates within its cavity. It can also be a wooden resonant box or metal can that can resonate along with strings or skins. It can be a sound board that resonates or a sound board mounted in a resonance box. Some instruments have more than one resonance box, like some ethnic string instruments. Some resonance boxes are real boxes, like a guitar, or they may be pipes that are mounted close to the part of the instrument that functions as the exciter, like with a vibraphone. So, resonators can take on many forms and be made of different materials, but the generalized purpose of the resonator is to sustain the sound and give the sound its main timbral character. In practise, most of the sound which is actually heard from an acoustic instrument is radiated from the resonant body. To get into resonance the resonant body needs to be excited by some sort of excitation function. This can be the plucking, bowing or hammering of a string, the beating on a skin or a strip of metal or wood, a reed, the air pressure of a flow of air, etc.

As an example let’s have a look at a plucked string instrument like the guitar again, it has a resonant body plus one or more strings mounted in a way that the strings can swing free, while one side of the strings rest on a bridge. The bridge is the path through which the kinetic energy in the swing of the string can be transferred to the resonant body. The kinetic energy will start to travel through the resonant body in the form of waves, which get reflected at the sides of the resonance box. Depending on the form and dimensions of the resonance box the waves and their reflections will form interfering wave patterns with knots at certain locations on the surface of the box. These knots add to the formation of formants, which are small frequency bands at fixed positions in the sound spectrum where frequencies get strongly emphasised. Imagine that the kinetic energy, which flows from the string, gets moulded into a typical timbre with strong resonances at certain fixed frequencies. When the frequency bands where these resonances occur are narrow and have a strong resonance, they will add more to the pronounced character of the timbre of the instrument.

Musically important formants are found in the frequency range that lies roughly between 500 Hz and 3500 Hz. E.g. human speech is based on how three to five strong formants shift from place to place in this range over short amounts of time. The formants that are present in the sound will melt together into one timbre and the relation between these formants is named the formant structure. In other words the formant structure is the total of the formants present in the sound and how the formants relate to each other. The individual formants can hardly be heard, as the human mind uses the total formant structure to recognize sounds. The basic technique used in sound design is to create sounds with expressively controllable formant structures. When using a synthesizer, very expressive and characteristic timbres can be created by causing strong and dynamically moving formants in the 500 Hz to 3500 Hz range.

Instruments like the grand piano have a sound board which is mounted in a resonance box. The kinetic energy first travels from the strings to the sound board and then from the sound board to the resonance box. Strings, board and box together form the mechanics which are responsible for the final basic timbre. The heavy sound board and thick and tight strings of the grand piano can store a lot of energy. This is one of the main reasons why the grand piano can play relatively loud compared to other instruments. E.g. plucked and bowed instruments like the guitar and the violin sound less loud, as their resonance box is made of relatively light and flexible material. In the case of a flute the fife itself is the resonator, and the prime resonance frequency of the fife will define the pitch of the sound. There needs to be a constant flow of air into the fife to sustain the vibration at the resonance frequency. When the air pressure increases by overblowing the flute there will be more turbulence in the air flow and this can create resonances at higher harmonic frequencies.

To summarize, almost every acoustic instrument or sounding object can be assumed to be a resonant body that is excited in some way, the excitation causing the resonant body to vibrate and resonate on the body's natural resonance frequencies. The resonance frequencies together form a formant structure that is mainly responsible for the final timbre. Energy is fed into the resonant body, which transforms the energy into a timbre with a specific formant structure. Most of the transformed energy leaks away into the air while the rest is transformed into warmth. This assures that the sound of an acoustic instrument or object will always die out when the excitation function stops and no more energy is fed into the resonant body. The shape of the resonating part of the instrument will add significantly to the final timbre of the instrument, a reason why acoustic instruments have their particular appearance.

Playing the timbre

As synthesizers are in practice often used to emulate existing instruments, recognition is the keyword when trying to emulate such a sound. The sound doesn't have to sound exactly like its real world counterpart, as long as people recognize it as sounding like that instrument. The trick is to make the mind of the listener associate the synthesized sound with the sound of the real world instrument. When the sound has the right sort of timbre and it is also played in the playing style of the real instrument the association is quickly made. As said earlier, playing style is very important here, and playing style can include playing the timbre. An example is how a trumpet player can drastically modulate the timbre by muting the trumpet with a beaker. Using a certain playing style can apply for totally new synthesized sounds as well. When a sound is created which is not modelled after an existing real world sound it often pays to experiment with different playing styles, until a style is found that seems to suit the sound best.

Changing formants can be very important in expressively playing the timbre, a well known example is the effect of the Wah pedal as used by electric guitar players. The wah effect is created by introducing a strong formant in the timbre, which is swept through the audio spectrum by a foot pedal. The popularity of the Wah pedal amongst guitar players has to do with the fact that with only a single controller, the foot pedal, the timbre of the sound can be expressively shaped. The guitar player can still do everything to pitch and amplitude with his hands, but now he has his foot as an extra way to express himself through tonal shaping of the timbre. For controlling a keyboard synthesizer two hands, and optionally feet, can be used. On the first monophonic synthesizers from the seventies the melodies could be played with the right hand, leaving the left hand to expressively play the timbre. One or two modulation controllers mounted to the left of the keyboard could be used to either bend the pitch, add some vibrato or sweep the timbre. When the modulation controller is a modulation wheel, it can control one single parameter in a sound. Another popular controller from the seventies is the joystick or X-Y controller, which allows for two parameters to be played by one hand. E.g. by letting the joystick sweep two independent formants or resonance peaks, expressive talkative timbre modulations can be played. Another possibility of the joystick is to crossfade between a maximum of four distinct formant structures.

Playing the timbre with polyphonic synthesizers is a bit more difficult, as on such an instrument the melodies are generally played by both hands. When the keys on the polyphonic keyboard are velocity sensitive, the velocity value can be used to control the timbre. However, the velocity value is sampled when the key is hit and keeps constant for the duration of the note. For this reason some of the better polyphonic synthesizers are fitted with an aftertouch sensitive keyboard. After a key is hit the timbre can be modulated by pressing harder on the pressed keys. Aftertouch can replace the modulation wheel effect, but it needs a lot of practising to learn to play it well. Some polyphonic synthesizers are equipped with a connection for a breath controller. This is a little tube that can be worn on the head like a headset, with the end of the tube right before the mouth. By blowing into the tube the air pressure is converted into a control signal that can be used to play the amplitude and/or timbre of the sound. And almost all polyphonic synthesizers are equipped with a connection for at least one foot pedal. Still, modern synthesis techniques allow for an enormous degree of controllability and the traditional human interfaces like the above mentioned controllers are not up to unleash the true sonic potential of the present day modular synthesizers. There have been many experiments with new controllers, like gloves with bend sensors, distance detectors like Theremin antenna's or infrared light distance sensors and all other available types of sensors. But no matter how well the sensors and interfaces work, they all require to learn a new playing style to play the sensors in a musical way.

The basic architectur of a modern synthesizer can be subdivided in three parts, the human interface to play the instrument, the sound engine that houses all the modules and does all the synthesis work, and some intelligence in between that connects the two parts in a sensible way. The intelligence part is housed in the microprocessor that has been present in polyphonic synthesizers since the end of the seventies. Many times this is the same processor that also processes MIDI information received form another instrument or play controller. Over the years these processors have become very powerful, today it is really like there is a small computer present. One of the newer functions that makes use of this extra power is the possibility to use a single physical controller to control several control signals or values at the same time in an intelligent way. This allows for modulation of the timbre over a range from very subtle to very complex. This technique is named morphing. In essence morphing does a crossfade between a number of knob settings to a new set of knob settings, the knobs that participate in the crossfade are named a morphing group. Morphing allows one hand to simply and intuitively play very expressive timbral modulations.

Analysis of timbres

Harmonic spectrum

The timbre of a single pitched sound with a static amplitude and a static timbre can be analysed into a harmonic spectrum plot. Such a plot reveals graphically all the partials present in a single pitched sound, and it is a useful means to analyse or define a static waveform from an oscillator sound source. The maths used in the analysis actually assumes the data to be a single cycle of a waveform to produce meaningful results.

In the nineteenth century it was discovered that all sounds are in fact the addition of a number of sine waves at different frequency and amplitude values. When the sound has a single pitch these sine waves will have a simple harmonic relationship to each other.

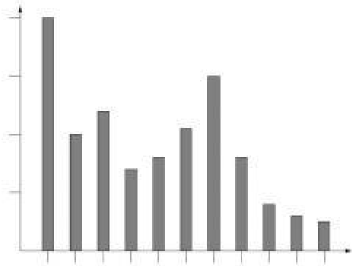


Figure 5 – Example of a harmonic spectrum plot

The sinewave with the same frequency as the perceived pitch of the sound is named the fundamental or first harmonic. All other sine waves present in the waveform have a frequency value that is an exact multiple of the frequency of the fundamental, the second harmonic will be two times higher in frequency, the third harmonic three times, etc. The group of all possible harmonics with their individual amplitudes is named the harmonic series. A harmonic will always have a harmonic relationship with the fundamental, but there might be components in the sound that do not have this harmonic relation. Then the name partial is used, as a partial does not necessarily need to have a harmonic relation, like the harmonics do.

In appearance a harmonic spectrum is a plot that on the horizontal axis shows the numbers for the harmonics. There is a vertical bar at each harmonic number position, which shows the amplitude on the vertical axis scale for the corresponding harmonic. The horizontal axis has a linear subdivision in whole numbers from the number one for the fundamental to a theoretically infinite number. The frequency of the nth harmonic in the plot has a frequency ratio of n:1 to the fundamental frequency. In practise it suffices to plot only the first fifty to hundred harmonics, as higher harmonics might very well be above the highest frequency of the human hearing range. The amplitude values of the vertical bars are in general percentages, not absolute values. The harmonic with the strongest amplitude is normalized to 100% and the amplitude values for all other harmonics are scaled to percentages between 0% and 100%. The relation or ratio between the amplitudes of the harmonics defines the timbre of the sound. The plot shows no absolute frequency values for the harmonics, but to get absolute values the frequency of every harmonic can be easily calculated by multiplying its number by the actual frequency of the fundamental. The amplitudes are calculated by first defining an absolute amplitude value for 100% and then calculating the amplitude values for each harmonic by scaling them to their respective percentages.

In the simplified exciter/ resonator model that was used earlier to describe the mechanics of acoustic instruments, the harmonic spectrum can be used to define the spectrum of a continuous excitation function. However, the harmonic spectrum is always a snapshot at a certain moment in time. In the real world the harmonic spectrum of an excitation function will vary over time, depending much on playing style and modulations applied by the musician. E.g. when the harmonic spectrum of a reed is analysed, it will show that it changes by the air pressure that is exercised and by the position and pressure of the lips on the reed. Morphing between two or three harmonic spectra allows for a more expressively playable excitation function. By using e.g. a breath controller assigned to a morph group it is possible to morph between two spectra, while an X-Y controller can morph between up to four spectra.

In scientific research papers harmonic spectra are generally plotted a bit different, as they might express not only sine but also cosine components. With such plots additional phase relations between harmonics can be analysed. But the why goes beyond the practical purpose of this book.

Sound spectrum

Next to a harmonic spectrum a spectrum plot of the total human hearing range can be drawn. Such a graph is named a sound spectrum plot. The horizontal axis will show absolute frequency values. As there is an exponential relation between the octaves and frequencies the horizontal axis has in general a logarithmic scale. When the harmonics of a sound of a certain fundamental frequency are plotted as bars the plot will become denser to the right.

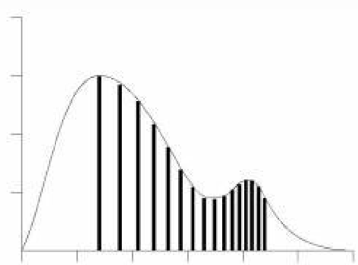


Figure 6 – Sound spectrum showing a harmonic series

The sound spectrum can also show partials that do not have a harmonic relation, show chords or show the sound spectrum of a very complex sound. There will be a bar for every sinewave component that is present in the sound. It is difficult to exactly read values of bars in such a plot, and in general it is not meant to be exact, but instead to give an impression of the overall sound spectrum. By connecting the tops of the bars a curve can be drawn that estimates the current sound spectrum. Such a curve is named the spectral envelope. The spectral envelope is in general used to get an idea of the sonic power that is present in a certain frequency band of interest.

Formant spectrum

The harmonic spectra for notes with different pitches can differ significantly on an acoustic instrument. By analysing the harmonic spectra of all notes and plotting them in a sound spectrum, a plot is generated that on the horizontal axis reveals the places where resonances or formants occur. Such a plot can reveal the formant structure of an instrument and can be very helpful in designing a sound that closely resembles the instrument. Such a plot is named a formant spectrum and is plotted as a spectral envelope on a logarithmically scaled horizontal axis. In appearance it looks just like a sound spectrum plot, but it has no bars, only the spectral envelope. The difference is subtle, a sound spectrum plot shows an analysis of an existing sound, while a formant spectrum plot shows which formant areas are needed to create a sound that is not yet in existence. A formant spectrum plot is an important piece of information for a sound designer.

In the exciter/resonator model the formant spectrum plot can describe the effect that the resonant body has on the sound signal that comes from the excitation function. It shows the frequency ranges which are boosted and ranges which are attenuated. There might be small strong peaks, indicating a very strong resonance, and small dips or notches where a frequency is strongly attenuated.

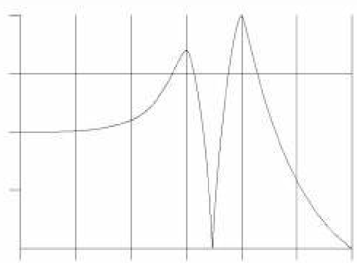


Figure 7 – Formant spectrum with two formants and a notch

The reflections of the waves that travel through a resonant body will cross waves that travel in other directions, causing an interference patterns similar to the interference patterns when some stones are thrown in a small pond. Sometimes a wave of a certain frequency will be cancelled completely by its own reflections and at that frequency there will be a notch in the formant spectrum. But another frequency might be amplified by its own reflections and this will show as a resonance peak or formant in the plot.

A formant spectrum is relatively static, but slight variations might occur depending on how strongly the resonant body is excited. Formants will hardly shift place but some might broaden or become more emphasised. Being able to morph between somewhat more complex formant spectra is an interesting option in sound synthesis, but in practice this needs special complex filters that are hardly found on synthesizers. Instead, on analog synthesizers level dependent distortions based on non linear characteristics of certain electronic components, aptly named distortion, are commonly used to emphasize sonic differences between soft and loud notes. When tweaked subtly, this technique can in practise work out very well. Digital techniques offer the possibility to use mathematical functions or lookup tables to describe level dependent operations that mimic the effects that can happen when the resonator gets excited by different levels of energy.

When the effect of a filter is described the same sort of plot can be drawn. Although in research papers you might find a different way to accurately describe the effect of filters, named the impulse response. The impulse response is the signal that will be on the output of the filter shortly after the filter input has received a single pulse of infinite short duration and with an infinite amount of energy. In practice a very short spiky pulse is used, with the maximum signal level the device can handle. When the signal on the output is sampled and analysed in a plot it should then reveal the formant spectrum of the filter. A similar method can be used to analyse the reverberant characteristics of a space like a concert hall, which in a way is an enormous resonant cavity. To produce the impulse a hydrogen implosion is used. A little bit of hydrogen gas is led by a small tube into some soapy water, forming a little bubble of hydrogen gas at the surface of the soapy water. The hydrogen is ignited by pushing a burning matchstick into the bubble, causing the bubble to implode. Such an implosion creates an almost ideal pulse. The sound wave of the pulse reflects against the walls and all the reflected waves form interference patterns in the space, colouring the sound of the reverberation of the pulse. This describes nicely what the impulse response actually is, in this case the literal reverberation of the space right after the hydrogen implosion. The analysis of the recorded impulse response can be used to program an huge electronic multi-tapped delay line, that will then give a very close simulation of the reverberation effect of the analysed space.

When a formant spectrum plot is specifically used to describe the effect that an electronic device like a filter or distortion function, a resonance box or a reverberating space has on a sound, then scientists name the plot the spectral transfer function of the effect. This is the graph that shows how the sound spectrum is changed by the effect. This transfer function is all important as it describes exactly what will happen to any frequency component in the original signal or sound. When working with synthesizers musicians use names of several typical transfer functions almost unconsciously. Like when they insert a lowpass filter or a highpass filter in a signal chain the lowpass or highpass refers to the type of transfer function of the filter.

Devices like microphones and loudspeaker boxes also have a transfer function. For these devices two transfer functions can be plotted, one that reveals how frequencies are affected and another that shows the phase shift or time delay for each frequency. These phase shifts or time delays are caused by the reflections of sound waves within the loudspeaker cabinet and the placement of the loudspeakers that have to reproduce the different frequency bands. A set of loudspeaker boxes that have a flat frequency response, but a wildly varying phase response, might faithfully reproduce a single monophonic sound, but will probably totally mess up the original stereo field for a stereophonic sound. So, note that a loudspeaker box in itself is also a resonant box and can significantly influence the colour and the spatial character of the reproduced sound. Ideally, both the transfer function plots for microphones and loudspeakers should show a flat horizontal line, which would mean a perfect device. But in practice microphones and loudspeaker boxes are far from perfect, meaning that coloration of the sound is inherent. That doesn't need to be a problem, as this coloration might very well be a wanted feature. Just think of an electric guitar amplifier and accompanying loudspeaker cabinet. In this case the cabinet actually takes over the function of the absent resonance box on the electric guitar. A strong coloration of the sound by the cabinet is very important here. For doing different kinds of sound recordings, a typical music recording studio will have several types and brands of microphones available. A microphone used to record vocals will most probably never be used to record a drumkit, unless maybe a special effect in the recording is wanted. The art of recording is very much about picking a microphone that gives the right sort of coloration for the timbre, and at the sound level produced by what needs to be recorded. Of course plots of transfer functions are really of little use here, a good set of ears and a lot of experience is much more helpful. As in the end the only rule is that it has to sound right.

Sonogram

To analyse how a timbre develops over time requires to go another step further with the plot. An example of sound with a very complex and dynamic timbral development is human speech. The human vocal tract is actually a very complex filter where several formants are created in different places of the vocal tract. Additionally the vocal tract can modulate some of these formants to create effects like e.g. growling sounds. Each individual's vocal tract has slightly different dimensions and several muscles are involved to shape the vocal tract. All these muscles can have their own individual tremors, causing their own different modulation effects. There is an unlimited amount of subtle sonic effects possible, giving each individual his or her individual voice. When thinking about this, it is pretty miraculous that humans can instantly recognize the voices of an enormous amount of individuals. The reason for a musician to use a modular synthesizer is many times to create his or her own individual sound, a sound that clearly stands out against the sounds used by other people. Such a sound needs character, and then it is good to realize that a good example of sounds that definitely have character are vocal sounds. So, when there is some basic understanding of the mechanism of vocal sounds, it is probably easier to create individual sounds with a definite personal character. Regrettably, human sound is a very complex matter, up to this day synthesized human speech still does not sound very natural, though recent technologies do come very close. The main clue to create individual synthesized sounds is to realize how formants play an important role in vocal sound. Human speech researchers divide human speech into phonemes, the short sounds that from the characters of speech. A phoneme has definite timbral development which cannot be analysed with a single formant spectrum plot. A formant spectrum of a phoneme can have up to maybe twenty five formant peaks or notches which are continuously altered, shifted and modulated while text is spoken. Additionally it might be voiced or unvoiced, meaning that there is either a definite pitch or more a noisy character without a detectable pitch. To be able to plot such sounds the sound is split into very short parts and for each part an analysis is made. These analyses are then plotted glues to each other in a special way, each individual analysis is plotted in a straight vertical line where the vertical position is the frequency axis. When a certain frequency component is present it is plotted by a grey dot, the dot becoming darker when the amplitude is stronger. The vertical lines are put next to each other to result in an image showing grey wavy patterns. The image is named a sonogram and reveals how the formant areas in a sound develop in time. The sonogram must be interpreted from left to right. Here are two examples of sonograms.

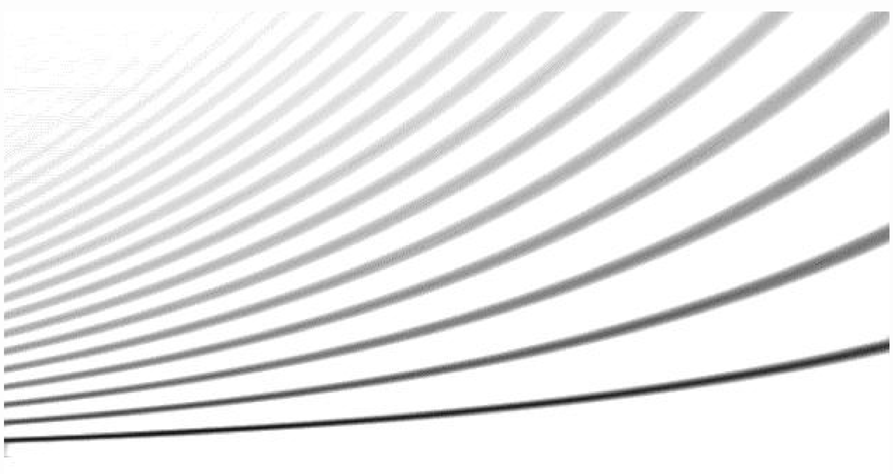


Figure 8 – Sonogram of an upward sweeping saw tooth waveform

The sonogram in illustration Figure 8 shows the analysis of a saw tooth wave sound that is swept up in pitch. Each grey line shows a harmonic, the lowest line being the fundamental. It is not difficult to imagine what happens in this sound.

Figure 9 - Sonogram of the Dutch word 'jasses'

The sonogram in illustration Figure 9 is the analysis of a Dutch word ‘jasses’, as spoken with much expression by the late Dutch poet Johnny van Doorn. The word expresses a strong feeling of disgust, like when one expects to drink a good wine but it has turned into vinegar. The initial unvoiced ‘j’ is shown in the lower left corner and very quickly morphs into the ‘àh’ when the two distinct dark lines start. Then it reveals that the ‘àh’ shifts up in pitch, while the more pronounced formants in the ‘àh’ appear, and then the pitch shifts down again. The ‘àh’ then morphs into the ‘sh’ that is clearly shown by the irregular grey stripes at the top half in the middle of the sonogram. The ‘uh’ clearly stands by itself and is shown by the four groups of stripes that together look like a distinct column. Note that a single stripe is actually a single harmonic, so the four groups of stripes actually indicate four formants. After the ‘uh’ comes another unpitched ‘s’, which has a less overall amplitude as the ‘sh’ in the middle of the word.

It is clear that when the actual sound recording and the sonogram are used together, it is not difficult to figure out what the sonogram shows. But a sonogram without the sound it utterly useless, without the explanation of this particular sonogram nobody would have been able to guess the actual word or sound it represents.

Summary

The graphs mentioned in this chapter are commonly used in sound synthesis. The harmonic spectrum is used to describe waveforms. The formant spectrum or spectral transfer function plot is used to describe filter characteristics. The sonogram is hardly ever used in sound synthesis and is for most people just a picture that looks interesting but without much meaning. These plots are generated by means of what is known as a Fourier analysis. The maths behind this analysis is pretty complex and you won't find it in this book. Instead a hands-on approach towards creating certain sonic effects will be used in the rest of this book.

Patchsheets and schematics

Making patchsheets

On analog modular synthesizers, which use cables to interconnect the available modules in the system, the cabling of a previous patch gets lost when a new patch is made. To be able to remake a patch later it is important to make a schematic drawing showing the cabling and the knob positions. Such a drawing is named a patchsheet. It is very important to make patchsheets on paper when the system has no provisions to store and recall patches by using some sort of patch memory.

Block schematics

It appears like digital systems with editor programs have made the use of patchsheets redundant. Still, it is a good custom to use paper to draw block schematics representing the structure of modular patches, as this creates a platform independent way to communicate about patches. In a block schematic each module or function is represented as a symbol. The symbols for modules and functions are interconnected with arrows, where the direction of an arrow shows the direction of the signal flow. In essence a block schematic represents a model. A model is a design which schematically shows all the aspects that are of importance in the design. Note that a model specifically refers to a 'design', and not to a brand or type of device. Manuals for analog prepatched synthesizers would always show the block schematic for the structure of modules and signal flows in the synth. By having a look at the block schematic it becomes immediately clear how many sound sources are present, how and where a modulation signal can be routed, etc. On a modular synthesizer each patch can be a different block schematic, which shows the power and flexibility of a modular system over a prepatched sytem. In a block schematic certain knob positions can be drawn or highlighted if these knobs have a special meaning in the patch, e.g. the knobs that are used to tweak the sound while playing.

On a digital synthesizer or a softsynth there is no apparent need to make oldfashioned patchsheets. Some softsynths use block schematics as a graphic interface to build patches, others use graphic representations of modules. With such systems block schematics on paper become even more important, as in a drawing on paper it is possible to group several modules that perform a certain function into a personalized symbol, a symbol that one can make up oneself. Symbols for sensors and mechanical input devices or even other synthesizer devices can be incorporated, when the block schematic is part of a score or a project description. So, no matter how elaborate the graphical interface for e.g. a softsynth might be, a block schematic on paper can always show more aspects of the project itself, aspects that happen outside the softsynth. There are many people that draw block schematics on paper to help bringing the stream of inspiration going or as a point of concentration while some deep thinking about a project is going on. Studying old block schematics can suddenly give new insights on the ideas expressed by the old schematics and inspire new projects. So, for several reasons it is a good idea to get familiar with block schematics.

Reading block schematics is quite easy, every block or symbol represents a physical module or function and following the arrows it quickly becomes clear what happens to the signals flowing between the modules and functions. In the western world it is customary to show the direction of a signal flow from left to right and/or top to bottom. This means that physical controllers or input devices, like keyboards, sensors, microphones, etc., are to be found at the top or on the left side. Sound sources are also often at this side, as sound sources deliver the source audio material to be processed later on in the patch. At the right side or bottom side one would typically find symbols for power amplifiers, loudspeakers or tracks on recording devices. Traditionally it was common to draw audio signal flows from left to right and draw modulation signal generators at the bottom with their signals flowing up into the symbols for modules that get modulated by the modulation signals. Today, there is a tendency to let signals flow downwards through symbols like a downwards pointing triangle for a mixer, the lower half of a circle for a sound source or signal generator, a full circle for a multiplication or controllable gain element, and annotations to the left side of a symbol to show details like the graph of a transfer curve or spectrum, etc.

There are no standardized rules how a block schematic or a symbol for a specific module should look like. Basically a block schematic and its symbols should simply be selfexplanatory. Still, there are some defacto standards on how e.g. a computer algorithm can be represented in scientific research papers or patent descriptions. But these defacto standards only standardize basic mathematical functions and do not include symbols for e.g. a distance detection sensor used to control the pitch of a sound source. Such a symbol can be devised by oneself. The amount and detail of the information in a block schematic depends fully on its purpose, e.g. if it is just a sketch for an idea or part of a score to be used by others.

Figure 1 shows an example on how symbols in a block schematic for a modular synthesizer patch could look like.

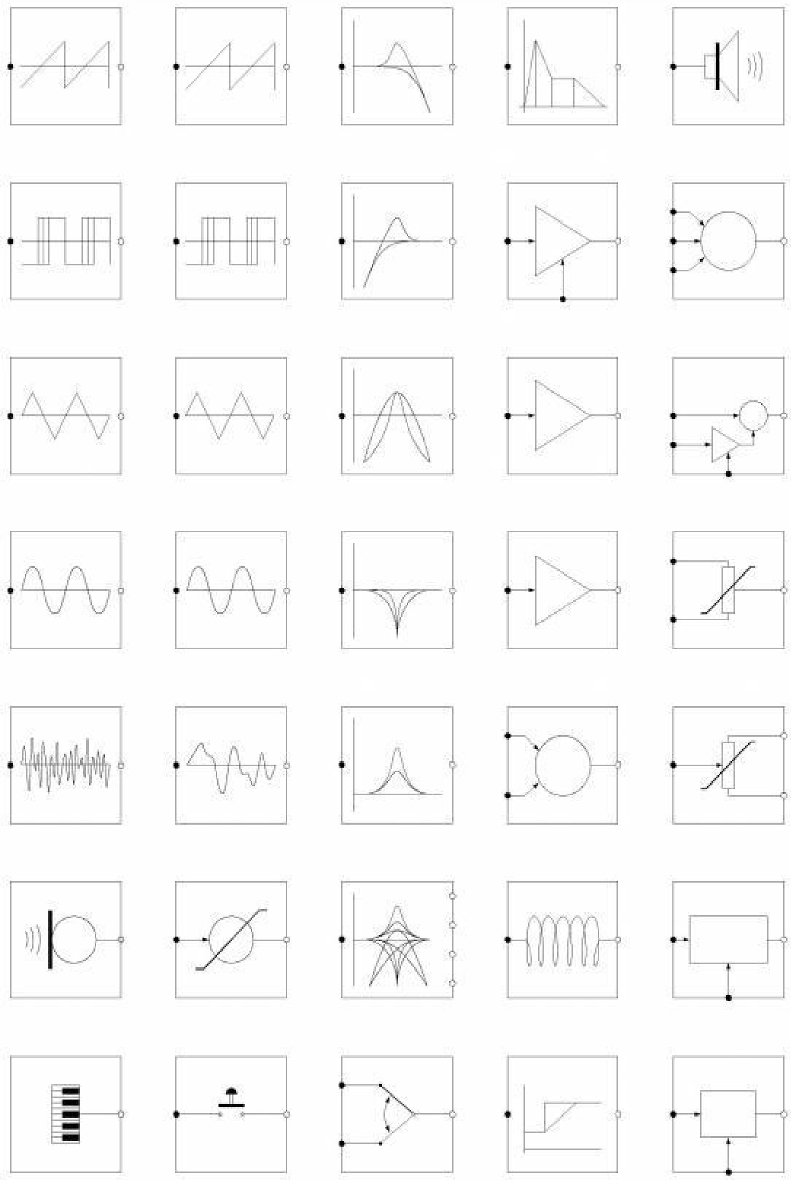


Figure 1 – Examples of symbols for common synthesizer modules

Introduction to the G2 system

Hardware and software

In the rest of this book the Clavia G2 system will be used to conduct experiments. The G2 system is a fully fledged modular synthesizer system based on fast DSP hardware. Clavia has released free demo software that emulates this G2 system in software. The demo software is less powerful as the DSP hardware, but is still powerful enough to conduct the experiments described later in this book. The good thing about the demo software is that there are hardly limitations in synthesis functionality or sound quality. Instead the limitations are in polyphony; as the demo software is basically monophonic while the hardware system is both polyphonic and four part multitimbral. The demo software is the ideal tool for learning and can be used very well in a teaching or workshop environment. There are versions for Apple Macintosh and Windows PC platforms. But although the demo software is somewhat limited in power, it still requires a fast personal computer with a 2 to 3 GHz CPU.

The latest demo software can be downloaded for free from the Clavia website at www.clavia.se. The full G2 manual can be downloaded as a .pdf file. You should always refer to the G2 manual for G2 specific subjects, as they go beyond the scope of this book. In the rest of this book it will be assumed that you have familiarized yourself with both the G2 demo software and the G2 manual.

The next few chapters will familiarize you with some of the general principles used in the G2 system that are not explained in detail in the G2 manual. These principles can in many instances be mapped on other systems as well. So, if you are using another system you will find most principles back on your system, although they might in cases be named slightly different.

Signal types

The G2 system

The G2 system is a true modular synthesizer, meaning that there are a number of different modules, each having their own function in a sound. There is a limit to the number of modules that can be used in a sound, each module eats away a little bit of the computational resources of the DSP chips, and when all resources are in use the limit is reached. Some modules eat away more than others, so it depends a bit on the sort of patch how much modules can be used. Still, if the G2 were to be compared to an analog modular synth a G2 patch would be the equivalent of a couple of square meters of analog modules. Compare each patch to be equal or even bigger that one of the real big systems that you may have seen on pictures from the sixties and seventies. And there is a system like that in each of the four slots. The modules in the G2 are inserted in a patch by means of the editor program. Next, cables have to be drawn between inputs and outputs of the modules and this is done in the editor program as well. And although doing this is dead easy, you of course have to know what the sensible connections are. And there are many, one could possibly write several books on this subject alone.

The G2 demo software uses only one slot that is in essence one monophonic voice plus one monophonic effects section.

Audio and control signal types

Before starting to make your own sounds on the G2 it is important to take a look at the signals that can flow from the outputs of one module into the inputs of another module. The signal outputs of modules are easily recognized, as they always have a square form. In contrast, all inputs have a round form. Trying to connect the output of a module to another output is simply not accepted by the program, which means that it is not possible to make 'dangerous connections' or short circuits between module outputs that could do damage to these outputs. This is very convenient, as anything that the editor program will allow you to do is completely safe. Note that on traditional analog modular synthesizers from the past it was very well possible to connect two outputs to each other and create a connection that could blow the analog circuitry.

A quick summary of the connection possibilities that the editor will allow or refuse:

- An output can be connected to one or more inputs

- An input can be connected to only one output and optionally share that output with other inputs

- Two outputs can never be connected to each other

- Several inputs can be connected to each other, but they must be connected to one single output to actually receive a signal

Meaning of cable colouring scheme

When several inputs are connected but there is no connection to an output somewhere, the cable colours will be light grey, meaning there is no signal running through these cables. These light grey cables can always be connected to an output later, it is not necessary to remove these light grey cables. But there is a convenient 'Delete Unused Cables' function, which will clean up the patch from any optional light grey cables present in the patch. Note that when a cable is of a red, blue, yellow, orange, green and purple colour there must be always some sort of a signal running through these cables.

When a module is placed in the patch its inputs and outputs have a certain default colour: red, blue or yellow. These colours indicate the quality of the signal and not really whether it is an audio or a control signal. It is up to you to decide if a signal really is audio or is controlling another module. When the signal is listened to it becomes audio by definition, and if it is not listened to but modulating something else, then again by definition the signal becomes a control signal. No matter if the quality of the signal would be high or low.

It is best to think of red and orange signals as hifi quality and blue and yellow signals as lofi quality. Lofi signals need less handling by the DSP chips, their generation and processing eats less computing power. In contrast, all red hifi signals are computed with very advanced techniques to offer the highest sound quality possible with digital techniques. E.g. the audio waveform oscillators with their red outputs use delicate antialiasing calculations to make the waveforms sound as smooth and analog as possible. But when an envelope control signal is used to control the amplitude of an oscillator signal, the control signal itself does not really need this antialiasing; it can be safely used with the lofi blue signal quality without any audible lofi effect in the resulting enveloped red signal sounds.

Actual sample rates in the G2 system

The signal quality depends on the sample rate of the signal. On the G2 the internal sample rate of a signal can be either 96kHz for red and orange signals or 24kHz for blue and yellow signals. Make note that green and purple coloured cables inherit the quality of the original cable, the green and purple colours are only graphic make-up applied by you and have no specific meaning.

Red and blue signals are virtual continuous or analog signals, like those used for audio waveforms and for smoothly gliding control signals. The yellow signal has only two states and its main use is to notify musical events, like the gate signals from the keyboard. The yellow signal is in fact much like a binary signal, knowing only two values that may be interpreted as on or off, 0 or 1, false or true, etc. Modules that have both yellow inputs and yellow outputs can sometimes have their inputs and outputs changed into an orange colour. This can happen when a red signal is connected to a yellow input. When this happens the samplerate of the module is changed from 24kHz to 96kHz, enabling some logic operations to be done at the fastest possible rate within the G2. Still, these orange signals will again only have the two on and off states, though now they can be used to operate upon audio signals and retain the audio sample rate of 96 kHz. Not all yellow inputs will have this behaviour, e.g. the envelope modules will not change to orange, but e.g. the sequencer modules and modules from the logic tab will. This enables some modules with yellow inputs to produce audio signals, e.g. the sequencer modules can be run at full audio speed to be used to create audio signals themselves.

Red 96kHz signals are the primary choice for audio signals. Oscillator modules will always have a red output to insure the highest audio quality. Blue signals are commonly used for control signals like envelopes and low frequency modulation signals, as these signal types do not need to be updated at very high speeds. In general it is said that it suffices to update an envelope signal a couple of hundred times a second. And on many other systems, like some softsynths, it is not uncommon to update envelopes and other control signals at a rate which is one hundredth of the audio samplerate, so when the sample rate is 48kHz the control signals would be updated 480 times a second. On the G2 however, the blue signals are always at a quarter of the red samplerate of 96kHz, so at 24kHz. The advantage is that this 24kHz lies just above the hearing range. On the G2 any zippery noises caused by the control signal being updated are now just above the hearing range and cannot be heard, though it might start your dog barking. The G2 has audibly a clear advantage over many other softsynth products here.

Still, a blue signal can be used as an audio signal, e.g. by using the output of a low frequency oscillator, which would normally be used as a control signal, and set the pitch to an audio frequency. This signal will have a slightly 'lofi' sound character when compared to the red signal from an audio oscillator module, but this lofi effect can of course be a wanted feature in your sound. It is totally up to you if you want to use the blue signals to carry your audio.

Most modules that process a signal or sound, like mixers, have a blue input by default. When the blue input is connected to a red output signal from another module the blue input turns into red and also the blue output of that module turns into red, if it wasn't already. This is a very convenient feature, as it optimises the DSP power used by the patch. The optimisation process for the patch, also named recompiling, necessarily has to briefly silence the G2 when a module changes from a blue to a red colour. During this moment all the DSP programming code is reshuffled to optimise the resources the DSP uses. This takes only a very short while, almost unnoticeable, but all modules will fall back to their initial states, meaning that e.g. a low frequency oscillator waveform is reset to its initial start-up value and sequencers restart at their first step. This silencing is in all practicality unavoidable on a system like the G2, the fact that adding a module or reconnecting a cable changes the 'architecture' of the synth model in the patch must mean that something must happen to cause that. While this happens the system simply does not know how to calculate audio as the code to calculate is momentarily out of order. This causes the brief silence, until the internal reshuffling is done and the system continues to do its musical work for you. This silencing happens when a patch is loaded in a slot, when the polyphony of a slot is changed or when in the editor program a new module is placed or a cable is connected to an input of a module.

Signal levels

Signal levels in the G2 system

Something that must be understood is how the levels of the signals relate to musical properties. In fact this is probably the only real difficult subject when working witha system like the G2. When this issue is well understood all other subjects suddenly become more clear and the G2 can be patched in a more intuitive way. It is important to get a feel for signals, e.g. how deep and how fast a certain modulation signal will modulate another module, e.g. will a vibrato sweep just be very shallow or will it sweep the sound wildly all over the place. This feel will come quite fast, just as the effect is so very audible. But it might still take some weeks or months before this feel becomes a second nature. The time this takes depends a lot on how much time you can or want to spend in experimenting with the G2.

Of course there is system to the signal levels. In fact, much effort was put into making the signal levels and their musical relation as balanced as possible. To explain this system some technical talk is regrettably unavoidable. However, the technical issues involved are not much and they apply to other digital systems as well. In the professional audio world these issues are considered the basic technical understandings one must have to be able to work professionally with digital equipment. So, if you're not a pro yet, hang on and struggle with great courage through the next few paragraphs. And if you are a pro you are kindly invited to refresh your knowledge a bit.

Signal levels

In a traditional analog modular system voltages and currents are used for every signal. But in the G2, as it is a digital system, there are of course no true voltages and currents that go through the virtual cables that are drawn on the computer screen. What actually runs through these virtual cables are digital signals represented by streams of digital numbers. There are two things that define the quality of such digital signals, the amount of digital numbers per second that is fed through the system and the precision of each of these numbers. As mentioned in the previous chapter there are two rates to feed numbers through the system, 24000 numbers a second and 96000 numbers a second. The precision of the numbers is expressed in bits and the numbers used for all signals in the G2 are in fact high-resolution 24bit numbers. To give an idea on how the quality of 24 bits turns out to be in practice, the signal-to-noise ratio is often used as the signal-to-noise ratio can be easily paired with the number of bits in a digital number. Every extra bit in a binary number represents an increase of 6 dB in the average signal-to-noise ratio of the digital system. It might look like the signal-to-noise ratio is a strange way to say something about the quality of a digital signal, but it is not. The idea is that a digital signal is always an approximation of an analog signal. Any deviation from the original analog signal will be perceived as noise. This noise doesn't sound like the soft noise from analog equipment, but it rather sounds like 'lofi' digital noise. The higher the precision of the digital signal, the closer it will approximate the analog signal, and there will be less 'left over' noise. E.g., an eight bit number has an 8 times 6dB is 48 dB signal to noise ratio, a sixteen bit number 16 times 6dB is 96 dB and a 24 bit number a 24 times 6dB is 144 dB of signal to noise ratio. 144 dB is well below the noise floor of the human ear, the sound of the heartbeat and the rushing of the blood through the veins are louder. So, 24 bits of precision is generally considered well enough for processing audio. Still, there are some angles to this 144dB as the 24 bits is what is totally available; it is in fact the whole dynamic range of the system. Meaning that when a signal would exceed this 24 bits the tops of the waveform of the signal are clipped off, as there is simply nothing beyond this 24 bits dynamic range. The important thing to understand about digital signals is that the bit depth is also the absolute boundary beyond which nothing else exists! It is not like with an analog tape that can be softly driven into saturation. This goes for every piece of digital equipment. This principle is even more important when making digital recordings, as when the audio signal has been recorded too loud and there is clipping in the recording, the clipping is final and basically a part of the signal is lost forever. There is no way to later construct what it was that has been clipped away, other than by pure guessing what it might have been. This means that with any piece of digital equipment the internal signal levels never use the full 24 bit resolution, as some headroom is needed to reduce the chances of clipping. In fact the total mix of all signals, waveforms, voices or tracks has to fit within the 24 bits dynamic range. So, the signals are 'embedded' in 24 bit numbers, but maybe only 22 of the 24 bits might actually be used. Which would give a headroom of two times the remaining bits times 6dB is 12 dB of headroom, while having a signal to noise ratio of 22 times 6dB is 132dB in the waveform or recorded track. In a digital synthesizer there must be a balance between the number of bits used for the actual recordings or generated waveforms and the available headroom for mixing these recorded tracks or waveforms later on. Take note that all headroom issues that apply to recording and mix tracks on a digital recorder apply equally to mixing audio signals within a digital system like the G2. In the G2 the waveforms are calculated with a headroom of 12dB, meaning that there is 22 bits of precision in each single oscillator waveform.

The G2 numbering system

To make working with the signals easier a special numbering system has been implemented on the G2, dividing the total dynamic range of 24 bits into units. In the editor screen and on the G2 panel the values are not represented in bits but in convenient units that actually have a musical meaning. Some Nord Modular users fondly named these units ClaviaUnits. Remember that on a traditional analog synthesizer there was the 1V/Oct norm for defining pitches. On the G2 there is a similar norm, defining that an increase of one single unit stands for an increase in pitch of a half note. So, the G2 has a '12 units/Oct' normalization. A single 'unit' is the exact equivalent of 1/12 Volt on a purely analog synth, but more importantly a single unit now also represents a key on the keyboard. What goes through a cable in the Editor program are streams of numbers with values expressed in 'units', just like voltages with a certain Volt value go through the patchcables on analog modulars.

Waveform levels

The waveform signal that leaves the output of an oscillator or LFO module swings between +64 and –64 units. This means that this signal can directly sweep another oscillator 64 half notes up and 64 half notes down, so a pitch sweep of almost eleven octaves! This sweep will not be stepped like in an arpeggio, but instead be a continuous smooth sweep. Between +64 and 64 there are 129 unit divisions (64 plus 64 plus one step for a zero value), but the units are in fact fractional numbers with a decimal point. Actually there are another 32768 subdivisions between two consecutive unit values. Meaning that a half note step is subdivided into 32678 additional sub steps. In practice the internal frequency resolution of the G2 is 0.0057 Hz, which is about 4000 intermediate steps between two half notes at the middle of the keyboard. Which for all practical purposes is pretty accurate and will make all pitch glides sound as smooth as they should.

To summarize, one unit represents a half note pitch step. The output signals from oscillators sweep over 128 half note steps between +64 units and –64 units, which can produce a sweep of almost eleven octaves. The units are always fractional numbers that can have something before and something after the decimal point, enabling very smooth and zipper free glides.

Manipulating signal levels

Now it gets a bit more obscure, as what happens when a control signal changes the signal level or amplitude of an oscillator waveform signal, e.g. when the oscillator signal is processed in an envelope module or by an attenuation knob. First, make note that envelope signals swing between 0 and +64 units. When an envelope generator is in rest, the control signal output on the module produces a value of zero. This is a very convenient value as when multiplying whatever value the oscillator signal happens to have with this zero value, the result will always be zero, as zero times anything is always zero. So, this zero value is able to effectively shut of the sound. When receiving a gate pulse from the keyboard the control output value of the envelope module will rise at the attack value speed until it reaches a maximum value of +64. Then it drops slowly back to zero again. So, the peak value of the envelope signal is +64. When this +64 is multiplied by the waveform's positive peak value of +64 the result is +4096 and when multiplied by the negative peak value of 64 the result is –4096. However, these values are way beyond the headroom, as the clipping level of the whole system actually lies at +256 and –256 units. So, when a straight arithmetic multiplication would be used to envelope the oscillator signal with an envelope value, most of the audio would be rocketed away into the nevernever lands that lie beyond the limits of the dynamic range of the system, resulting in very severe clipping. To solve this issue scaling is used in all operations that can dynamically alter the level of a signal. It is obvious that when the audio signal swings between +64 and –64 and the envelope control signal is at its peak value of +64 the audio signal should be passed with unity gain similar to the 0dB mark on a mixing desk channel fader. Unity gain means that the level at the input is exactly equal to the level at the output. Now, note that multiplication of a number by 1 leaves the number unaltered. So the +64 units peak value of the envelope control signal should behave like it is an arithmetic number with the value 1. To create that situation the +64 units signal is scaled down to an actual value of 1 before the envelope is applied. This downscaling happens automatically within a module whenever a level is being changed. There is no need to worry about this downscaling, it is simply hidden within the system and always works how it should. The only rule to remember is that 64 units always translate to unity gain or 0dB. Similarly –64 units translate to the inverse of unity gain, it negates the signal turning it into an inverted or phase reversed copy of the original signal.

The G2 levels and units standard

Now the rules regarding levels and units can be summarized more completely:

When units relate to frequency, a one-unit step equals a half note shift in pitch and each unit also stands for a key on the keyboard.

When units relate to amplitude, all signals generated by oscillators swing between +64 and 64 units. A value of +64 units causes any gain controller to have unity gain and the value of 0 units will shut the gain controller completely of. A value of +64 units emulates the arithmetic number one. And a value of +32 units arithmetically equals a half or –6dB.

When amplitude relates to frequency, or when the output of any oscillator is used on an unattenuated pitch input of another oscillator or filter, the full oscillator signal will cause a pitch sweep of almost eleven octaves on the other module. An envelope control signal, which goes between 0 and +64 units, will cause a sweep of over five octaves. On filter modules the modulation input with the attenuation knob is twice as sensitive and an envelope signal between 0 and +64 units can cause a filtersweep of almost eleven octaves if the attenuation knob is fully opened.

Remember that internally in the G2 all values expressed in units are in fact fractional values and have a fine subdivision after a ‘decimal point’, which always allows for smooth and zipper free glides and sweeps of both pitch and amplitude.

When for some reason a patch does not seem to follow these rules, there must be a module that is inadvertently messing things up. The bad news is that this module must have been placed by you (or someone else) for some reason, so the trick is to find this module and see if another setting of knobs might solve things

Attenuation of signals

But what do the attenuation knobs do to a signal? Basically, when an attenuation knob is closed it will just shut off the input and when it is fully open it will pass the input signal with unity gain. When a knob is slowly opened the scale or attenuation curve of the knob can be linear or exponential. If the knob behaves in a linear fashion the shown knob value is in fact a percentage, if the knob is behaving exponentially the shown knob value can be either just a number between 0 and 100 or a value in dB. This number between 0 and 100 for the exponential scale does not have any particular meaning. On some modules it is possible to set the knob to an exponential, linear or a dB scale. If you want the exponential scale to have a meaning then you must set the knob to a dB scale. The exponential and the dB scale have exactly the same feel, in fact they are the same scale but are only shown with a different unit descriptor. But the linear scale will give a very different feel. Only by turning the knobs and listening to their effect can you develop a feel on how the scales behave musically.

In the editor program the little yellow value popup that appear when the mouse pointer is held over a knob show two values. The top number is the value on the scale and the bottom number is the MIDI value of the knob. A MIDI value can have 128 possible values between 0 and up and including 127. This makes it a bit cumbersome to display values that have a particular musical meaning as all scales must have 128 positions to be compatible with MIDI. It is an inheritance of how things were when MIDI was invented. So, a scale of 100% must be subdivided in 128 steps, making each step equal to a fractional number instead of a whole number, which would probably make life easier for many of us. But to remain compatible with MIDI and all your other MIDI-equipped musical instruments and computer programs this dividing of scales in 128 steps can not be avoided.

Summary

The G2 set of rules about signals and values is in practice a nicely balanced system. In general it works out so well that there is hardly a situation where unwanted clipping occurs or the signal inadvertently seems to drop to a much lower level. When clipping or a drop in signal level occurs it has always to do with something in the patch. An example is when more than four oscillators are mixed together, as this mix might occasionally exceed the headroom. In such cases the mixed signals need to be attenuated to a level that is roughly the same as the level of a single oscillator before being processed further. This is just common sense and doing so will quickly become a second nature. There are many modules that have a small attenuation control where the signal can be attenuated by –6dB, -12dB and at some places also by –18dB or be boosted by an extra +6dB. These attenuations are applied to the input signals before being mixed, so the internal mixing process will not cause internal clipping in the module. Polyphony might also push the total mix of voices over the headroom limits, as pressing eight keys at the same time is like mixing eight oscillator signals. So, when a patch with lots of voices of polyphony is used it in general needs to be attenuated somewhat. The exact amount can be easily tested by what is named the full hand test, when pressing a lot of keys at once by putting your whole hand on the keyboard there just should be no clipping. The best place to set the right amount of attenuation for a polyphonic patch is where the voice signals enter the FX area, the area in a patch where effects like reverb and echo delays are commonly placed. The FX input module has a dB attenuation control setting and this control setting should be set to a value so that the patch passes the full hand test. The output module in the FX section can in general be boosted, e.g. if the input module is attenuated by –6dB you can try to boost the output module by +6dB or even +12dB. Just try different settings until the average volume of the patch is loud enough but still no clipping occurs when six to eight keys are played with maximum force.

Signal routing

Switches

A powerful feature of a modular synthesizer is that the signal flow through a set of modules can be rerouted by using switches as an alternative for repatching patchcords or rearranging a pin-matrix. A switch can be a module by itself and be patched between other modules to create alternative routings controlled by the switch module. Clever use of switches avoids having to repatch the patch cables that connect module inputs and outputs. A choice of different selections can be made by using rotary switches. There are two types of rotary switches; one type with multiple inputs and a single output, and the other type with a single input and multiple outputs. Rotary switches can also have ‘multiple decks’, meaning that two or more similar switches are mechanically connected to allow e.g. the switching of stereo or multi-channel signals.

Some analog modules use rotary switches for selecting module options. An example is a waveform switch on an oscillator module. Often oscillators provide several different waveforms, and either several mixing knobs or instead a single waveform rotary switch could have been implemented by the synthesizer designer to route one of the oscillator waveform signals to the module output. In this example the rotary switch is a cheaper alternative to using several mixing knobs. Rotary switches built into a module can also be used to do things like switching the pitch range of an oscillator up or down by one or more octaves.

Matrices

A very special type of device is the matrix switchboard or pin-matrix. This is basically a two-dimensional ‘multiple input / multiple output’ switchboard where any input can be connected to any output by either a toggle switch or a pin that must be plugged into the matrix. On an analog modular synthesizer equipped with a matrix all output to input connections can be made by using pins instead of patch cables. Matrices give a clear overview of the signal routing, it is much easier to see which output is connected to which input by just looking at the pins, instead of having to look at the noodle of cables hanging out of the front of a modular system using patchcords. In the old analog days pin matrixes used to be quite expensive and also prone to crosstalk, so they were not commonly used. On a digital system matrices can be easily programmed in code or prepatched by combining a bunch of switch and mixer modules. In the experience of the author matrix synthesizers are definitely the best balance between ease of use and flexibility.

Sources and destinations

When a prepatched synth uses rotary switches to route modulation signals, there are two systems that can be used, a system of destinations or a system of sources. The difference between the two systems is that in the destinations system a single source can be routed to one of several destinations, and that in the sources system each destination can select one of the available modulation sources. In the destinations system the rotary switch is positioned at the modulation generator module, the module would have a switch that would say; "Where do you want the modulation signal to go to". With the sources system the rotary switch is at the module to be modulated, the module would have a rotary switch saying; "Where do you want the modulation signal to come from". The system with the destinations is the cheapest to implement, but it limits a modulation source to be used for only one single possible destination. So, when a low frequency oscillator is used to add a bit of vibrato to the oscillator it can't be used anymore to also sweep the filter, as the signal can go to only one destination. With the sources system both the oscillator and the filter can select the same low frequency oscillator as a modulation source. So, the advantage of the sources system over the destinations system is that several modules can share the same modulation source, which the destinations system does not allow. When a modular system has separate switching modules available, these will most commonly be used in a sources system, using a multiple input to one output switch to select a source for modulation or to add an effect to.

On the G2 there is the choice to use a sources or a destinations system. Instead of switches, mixer knobs with mute buttons can be used to quickly turn a modulation on or off. It is also possible to patch the equivalent of a matrix pin system, such as was used on the vintage EMS VCS3 synthesizer. By using mixer knobs with mute buttons to build a matrix the ultimate in flexibility in signal routing is achieved. A destinations system is the simplest system and cheapest to implement, but also the most limited. A matrix pin system is the most flexible, as in theory literally anything can be modulated by any available modulation source with this matrix pin system. It is also the most expensive to implement. The sources system does well in many cases and is often the best balance between flexibility and the use of computational resources. In practice many patches can use a mix of the sources system and some of the 'add along' mixer chains that will be explained later.

The G2 offers an abundance of different types of switches that together provide for a lot of possibilities. There are switches that have multiple inputs and a single output and switches that have one single input and multiple outputs. These two types are commonly named selectors and distributors. Both types are available as manual switches, where nameable pushbuttons select and display the source or the destination in the frontpanel displays. But there are also controllable switches that can be set into any position by a control signal. These controllable switches have no pushbuttons, but instead have a control input that defines the current position on the switch. This means that it is the level of the control signal on the control input that defines which source or destination will be connected to the output or input of the switch. Controllable switches are commonly named multiplexers or demultiplexers and the G2 has at present five of these modules.

Chaining switches into 'multiple-deck' switches

The nice thing about the manual switches is that all have a control output that produces a level signal with a value that denotes the position of the switch. When this control output is routed to the control input of a multiplexer it will make the multiplexer act as a slave switch, conveniently following the setting of the manual switch. The slave switch will now act as a second 'deck' of a mechanical multi-deck rotary switch. This allows for making stereo signal switches or complex multi-channel switches. The control outputs of the switches increase in steps of four units. The manual eight input switch will produce values of 0, 4, 8, 12, 16, 20, 24 and 28 units on its control output for the eight positions it can be in. The multiplexer modules will use these values to switch to another position. E.g. when the signal on the control input of an eight channel multiplexer is below 4 units it will be in the first position, when the control signal is 4 units or up to but not including 8 units, it will be in the second position. Exactly at 8 units it will switch to the third position, until it receives a value of 12 units, etc. At 28 units and above, the switch will rest in the eighth position. So, the control input does not need to receive an exact number, but it uses numbers within well-defined ranges.

Exceptions to the rule

The G2 'crossfading eight channel multiplexer module' differs from the other controllable switch modules, as it uses steps of eight units instead of four units in its control range to select the mnext switch position. The reason is that the signal of a modulation signal generator module, which must be set to a 'unipolar' (=positive values only) signal output range, can be used to easily step through the whole 'crossfading' range of this module. Note that the first four multiplexer modules on the G2 are primarily meant to be used as slaves for the manual switches, but this special 'crossfading multiplexer' is specifically designed to be controlled directly from a smoothly varying modulation source signal. The eight input cross fading multiplexer can be used for a variety of special effects. Imagine that the eight outputs of an eight-tap echo delay unit are connected to the eight inputs of the multiplexer and the delay receives audio from a beat box connected to an audio input of the G2. When the delay time follows the beat of the beat box the delay will hold e.g. the last half bar or full bar of the beat box pattern. Connecting a triangle low frequency oscillator signal will dynamically switch from tap to tap. As the output signal of each tap has a different time delay the output of the multiplexer will be a signal where the audio contents of the tap delay will be warbled in time. When instead of a low frequency oscillator a sequencer module is used, the contents of the delay line can be warbled in all sorts of wacky patterns. You can let your imagination run wild on how much musical fun this 'time warbling' can be. It is one of the possible techniques that make the G2 unique amongst other synthesizers.

Multiplexers and sequencers

The controllable eight input and eight output multiplexer switches are very related to sequencer modules. By controlling them with an upward sloping sawtooth modulation signal the switch positions are sequenced from left to right, just like on a sequencer. A down sloping sawtooth will select the positions in reversed order. By using a triangle waveform the positions are selected back and forth. And by using a sequencer to control the position the multiplexers can be stepped in any pattern. Other possibilities are to combine two modulation signals of different rate, so the interference pattern that results makes the multiplexers step in very complex patterns back and forth. Also an envelope signal can be used to step through the positions. It is even possible to add the output of the switches to the signal that controls the position of the switch, which creates the equivalent of a ‘cellular automaton’.

A/B compare switching to turn a module function on or off

The two input switch is often used in more complex FX patches to bypass a FX modules signal chain. To do so, the first input is connected to the input of the FX chain, while the second input of the switch is connected to the output of the FX chain. On the output of the switch module there is now either the clean, unprocessed signal that is present on the input of the FX chain, or the processed output signal of the FX chain. This also allows for a chain of mutable FX modules, where each module can be conveniently switched in and out of the FX chain by using this A/B switch on every single FX module in the chain. This can also be done with the bypass buttons on the FX modules themselves, but using a switch with a nameable button allows for a clearer interfacing with the G2 frontpanel displays. And by using a two input multiplexer module (the value switches), the bypassing can be controlled with e.g. a gate signal or a clock signal, so an effect can be rhythmically turned on and off.

The two input multiplexer is a lot like a crossfader module, though it can only toggle between the two inputs and cannot fade smoothly like the crossfader can. This means that a crossfader can also be used to bypass an effect. Unlike the multiplexer switch it can also smoothly fade from dry to wet. For many effects, like echo’s, this is a nice feature. In fact such a dry/wet control is already built into e.g. the reverb module. But the ‘bare bones’ G2 echo delay line modules have no dry/wet control, nor a bypass button, and here one can easily make a bypass or dry/wet control with either a two input switch or the crossfader.

Mixing

Importance of mixer modules

Many musicians believe that the oscillators and filters in a synthesizer are the most important for synthesizing sounds. But even more important for a musician is to be able to play these sounds expressively. Expression is created by adding modulations like note bends, volume and timbre accents, vibrato's, etc. With a modular system all sorts of modulations can be created; manual modulations, automatic modulations at a relatively slow speed to ceate a sense of development or superfast modulations at audio rates that will produce changes in the timbre. Increasing and decreasing the modulation amounts gives those expressive effects to the sounds that are so unique to the modular synthesizer. Mixing becomes an issue when sound sources and/or modulation sources need to be mixed in some way. And except for the most simplistic cases mixing will always be needed. Which makes mixing actually the single most important issue on a modular synthesizer. This might seem a bold statement, but keep in mind that mixers are the glue that binds everything together. Just keep in mind that mixers let you blend the effects of the oscillators, filters, distortions, echo's, reverbs, etc., into that one total sound you're after. Look at it this way; when preparing food blending is what makes the difference between a magnetron meal and a haute cuisine dinner by a five star chef-cook. The final dinner will depend on the blending skills of the chef, and not only on the ingredients. Now, when e.g. a rompler synthesizer relates to a magnetron, a modular system relates to haute cuisine where all has to be blended. Keep in mind that on a modular system it is you who has to be the chef. Like any chef, earning your stars happens through building up your experience over the years. Using a single filter in the straightforward way may not give the filter sound you are after. But using two or more filters plus some modulations, distortions and effects, and then delicately blending their balances with some proper mixing techniques, might give exactly what you need. So, mixing on a modular system is really more than just tying things together, it is also blending different sonic aspects into the final sound. During the blending of sounds and modulations certain psychoacoustic issues are very important as well, as finally it is all about how the sound will be perceived by the ear and in the mind of the listener.

Different types of mixing

There are different approaches to mixing signals and at least two techniques are used with modular synthesizers. The first is the common type of mixing that is done on a mixing desk in a recording studio or during a live performance. In essence every single mixing channel is individually set to a certain range between silence and a level that is named the 0dB point. This point is nicely marked on the faders of a mixing desk. Audibly, the reference is 'the mix'; each fader is used to set the presence in the mix of the instrument or track that lies under the fader. With this type of mixing the absolute volume level of each instrument is set individually, just until it has the right presence in the mix and the total output level does not exceed the headroom of the recording device used. For this type of absolute mixing faders with an exponential response curve work best. These faders can be easily recognized by their dB scale printed next to the fader knob.

Relative mixing

On a modular synthesizer absolute mixing is also present, e.g. when setting the presence levels of various drum and percussion sounds in a percussion patch. But in between modules there is also another type of mixing with the distinct purpose to set a certain ratio between two or more signals. An example is when two different waveforms are mixed; resulting in a single new type of waveform that might have some desired new properties. Maybe these properties are only present when the ratio in amplitude is exactly 2 to 1. To set this type of ratio the amplitude relation between the two signals must be set to these exact values. This type of mixing is named relative mixing and is very common on synthesizers. Another common example of relative mixing is the dry/wet setting on an EFX box. For relative mixing the faders and knobs with a linear curve are the most useful, as they offer a more balanced range over exponential knobs to set exact ratio's. E.g. when a ratio of two to three is needed, the first linear knob can simply be set to twothirdth's open and the other linear knob to fully open. This will give the two to three ratio. But when exponential faders would have to be used it is in fact quite difficult to find the right setting for this twothirdth's on the dB scale. As mentioned before, many mixer modules on the G2 have a button that can change the mixer knob curves instantly from exponential to linear behaviour and vice versa. The rule of thumb is that in between modules, while synthesizing the basic sound, linear curves often work best. While at the end of the patch or a 'signal chain', where the audio comes out and the final volume is set, exponential mixing works best. The exception to the rule is when only very small amounts of modulation need to be added, in this cases it is exponential knobs that actually work best, as they offer the finest resolution at the low end of the knob. E.g. when a little bit of an LFO signal needs to give just a little bit of vibrato on an oscillator the exponential knob is a necessity, as the pitch sweep of the vibrato is very small compared to the whole pitch input range of almost eleven octaves. The question to ask oneself is: 'do I set the absolute presence of a single sound in the final mix', or 'do I set the exact relative mix between two or more intimately related things'. One way to solve this question up front is to ask if the mixing could basically also be done with a crossfader plus maybe some additional scaling after the crossfader. If so, then there is definitely a clear case of relative mixing, as a crossfader in fact sets the ratio between two signals. In practice the whole issue is easily solved, as with absolute mixing the exponential curve has a better feel and with relative mixing the linear curve feels better. On many mixer modules the type of knob curve, linear or exponential, can be quickly set with a button. Simply trying out these curves reveals pretty quickly which setting has the best feel, and so which curve needs to be chosen for the knob.

With relative mixing it is often the case that it is not two signals that simply need to be added, but that in fact one signal needs to be subtracted from the other signal. This is many times the case with control signals, but it also happens with audio signals. It might make a big difference in sound when a signal is added in antiphase to another signal. In this case subtracting instead of adding can do this. To provide for this possibility some mixer modules have an invert button next to their inputs. This button inverts the signal by changing it into an antiphase signal, before it is added to the output signal of the mixer. And adding an antiphase signal is equivalent to subtracting that signal from the other signal.

Add along mixing

The G2 mixers have the unique feature that they can be chained. The main property of the chain input is that a signal that comes in on this input falls through the module unaltered to the output. Meaning that this input has always unity gain for the chain input signal. This is not only out of convenience to easily add en extra mixing channel when needed, there is in fact a very powerful mixing technique based on chain inputs, named add along mixing. In a modular synthesizer it is many times the case that there is some reference value defining some musical aspect and then one or more signals are added along to this reference value to create expressive modulations. This happens a lot when mixing control signals. Keep in mind that a control signal is always related to some musical property. An example is when the pitch of an oscillator is controlled. The reference value would probably be the keyboard note value. Added along to this keyboard note value can be a note transposition value from a sequencer. Added along to the transposed note value can be an envelope value that temporarily bends the transposed note on the attack of a key press or a new sequencer note. And added along to the transposed and bent notes there might be a little bit of a vibrato control signal. So, to the reference value that originates from the keyboard first a transposition signal and then two more modulation signals are added along to the reference value before the result is finally fed into the oscillator pitch input. A big advantage of this type of mixing is that it doesn’t matter at which point a modulation signal is added along in the chain, it will never influence the level of modulation of the other modulation signals.

Figure 1 shows how this example can be patched with a couple of one-channel mixer modules. Each one-channel mixer chain input is connected to the output of the previous mixer. At the beginning of the chain is the reference value, in this case the note value from the keyboard. Keep in mind that this reference value falls unaltered through the whole chain. In the first one channel mixer the transpose signal from the sequencer is added along to the note value, then the note bend envelope signal is added along and finally the LFO vibrato signal.

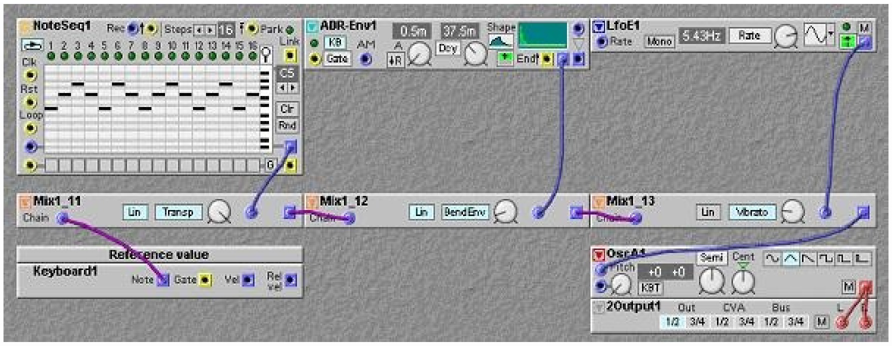


Figure 1 - Example of add along mixing

The purple cables show how the original reference value falls through the chain without the possibility of being inadvertently attenuated by a mixer knob. The three modulation control signals are added along the purple cabling. The importance of this type of chaining on the G2 is that the buttons on the one-channel mixer modules can be named with a proper name, clearly referencing the musical feature it is related to. Pressing or depressing this button switches the feature on or off. If switched to on, the mixer knob sets the amount of modulation and if switched to off the modulation signal is instantly decoupled from the chain, stopping the modulation immediately. When the mixer knob is assigned to a G2 front panel knob the button is automatically assigned to the pushbutton under the front panel knob. The button caption text is shown in the display, together with the positional value of the mixer knob. The panel pushbutton can be conveniently used to toggle the musical feature on or off, with the pushbutton light indicating the on/off state. On the G2 this is the preferred way to interface the musical feature with the panel controls. The mixers that are specifically designed to be interfaced to the G2 panel knobs are the one channel mixer, the one-stereochannel mixer, the two-channel mixer, the four-channel mixer and the four-stereochannel mixer. When an odd number of channels is needed the mixers are simply chained, e.g. for five channels chaining a four channel and a one channel mixer will do the job.

Dynamic reference for add along mixing

There are cases when increasing the amount of some modulation signal, which gets added along in the chain, should also change the reference value. A very common example is when a filter is modulated by an envelope sweep. When the envelope signal is applied directly, the sweep is referenced against the cutoff setting of the filter. When the modulation amount for the sweep is increased, so the sweep gets deeper, it is musically preferable to automatically lower the cutoff frequency a bit. This has the effect that the sweep doesn't appear to be on top of the cutoff frequency, but rather be more symmetrical around the cutoff frequency. To do this an extra layer of modulation is needed. Which means that the envelope modulation signal is itself also modulated before being added along in the chain of modulation mixers. The signal that modulates the envelope amplitude is a variable value that will be derived directly from a knob. In effect this knob will be used to set the depth of the envelope sweep. Additionally this variable value will also be used to lower the cutoff frequency. For this value to be able to lower the cutoff frequency when its value is increased, the variable value needs to be inverted before it is added to the chain. As the envelope signal itself will be modulated, it doesn't need a mixer knob anymore. It is the amount of modulation signal that will control how much the envelope sweep is present in the final modulation signal for the filter.

Figure 2 shows an example of how this type of expressive envelope modulation can be patched. The module named Cutoff provides a value that is varied by the knob on the module. It is set to unipolar mode, meaning that turning the knob fully left will produce a value of zero units, while when turned fully right it produces a value of +64 units. The purple cables show that this signal falls through the chain unaltered and so will directly set the cutoff frequency for the filter. The maximum value of 64 units will shift the filter cutoff frequency by 64 half notes, so slightly over 5 octaves.

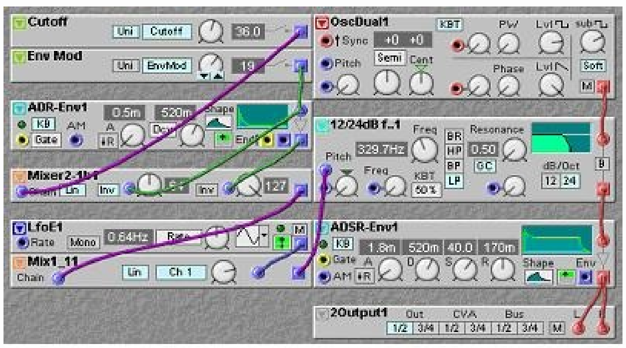


Figure 2 - Envelope sweep modulation example

The module named Env Mod also produces a value that depends on its knob setting. But this module is set to bipolar mode, meaning that it has a range of –64 to +64 units, while at the centre position of the knob the value will be zero. This value is fed into the input of an envelope generator module, which means that on the output of the envelope module there will be an enveloped control signal with it's peak value exactly equal to the value set on the variable value module. So, turning the Env Mod knob will modulate the envelope manually between a negative envelope with a peak value of –64 units, though no envelope signal at all in the centre position, to a fully positive envelope signal with a peak value of +64 units at the right extreme of the knob. Then, a two input chainable mixer is used to combine the cutoff value on the chain input with the modulated envelope signal plus subtract a bit of the Env Mod value to lower the cutoff when the envelope amplitude is increased through the EnvMod knob. Musically the amount of envelope sweep can now steplessly be tweaked with one single knob between an upward sweep and a downward sweep. The downward sweep does not shut the filter completely off, as the sweep gets automatically centred around the cutoff setting.

The mixing chain can also be drawn in a schematic. Drawing schematics is never a bad idea, it gives insight from a different angle and allows to see the structure better than in the actual patch screen. There are no specific rules for the style of these schematics; one can use a personal style with personalized symbols. As long as the style is able to clarify matters any style will do.

Here are two possible schematic drawings of the two previous mixing examples. To simplify matters only the mixing chains are drawn.

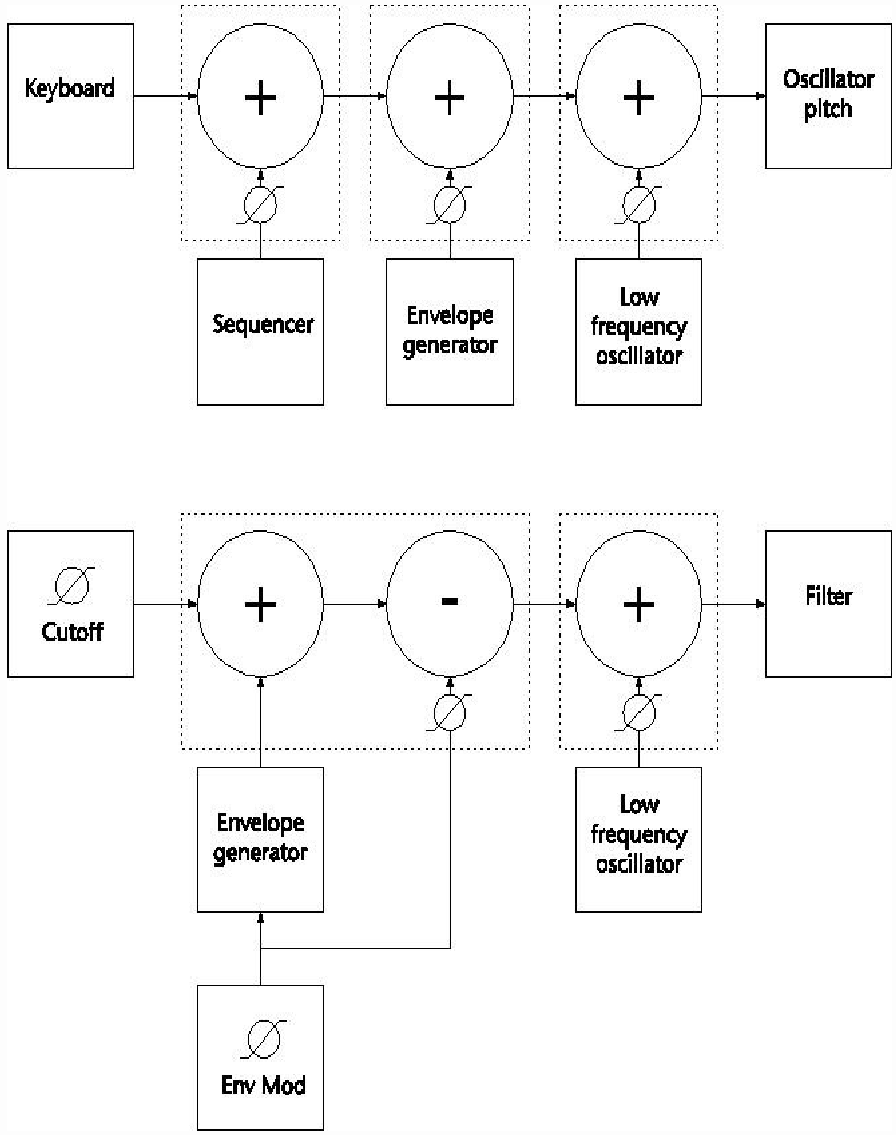


Figure 3 - Schematics of the mixing examples

The rectangles are symbols for a specific module and the circles are symbols for a specific operation, like add or subtracting. The dotted rectangles symbolize the actual mixer modules used in the G2 patch. The small circles with the diagonal lines symbolize panel knobs. The chain starts at the left module, which provides for the reference value, and proceeds from left to right along the fat horizontal arrows. The modules at the bottom generate control signals that flow upwards to be added along the mixer chain. There is now a clear control signal path for a specific parameter, the top example shows the path for the pitch parameter and the lower example shows the path for the timbre parameter.

In a synthesizer built according to the VCO->VCF->VCA model, like the traditional analog monosynths and polysynths, there can be three of these mixer chains for the musical parameters pitch, timbre and volume. Each chain would control either the VCO, the VCF or the VCA.

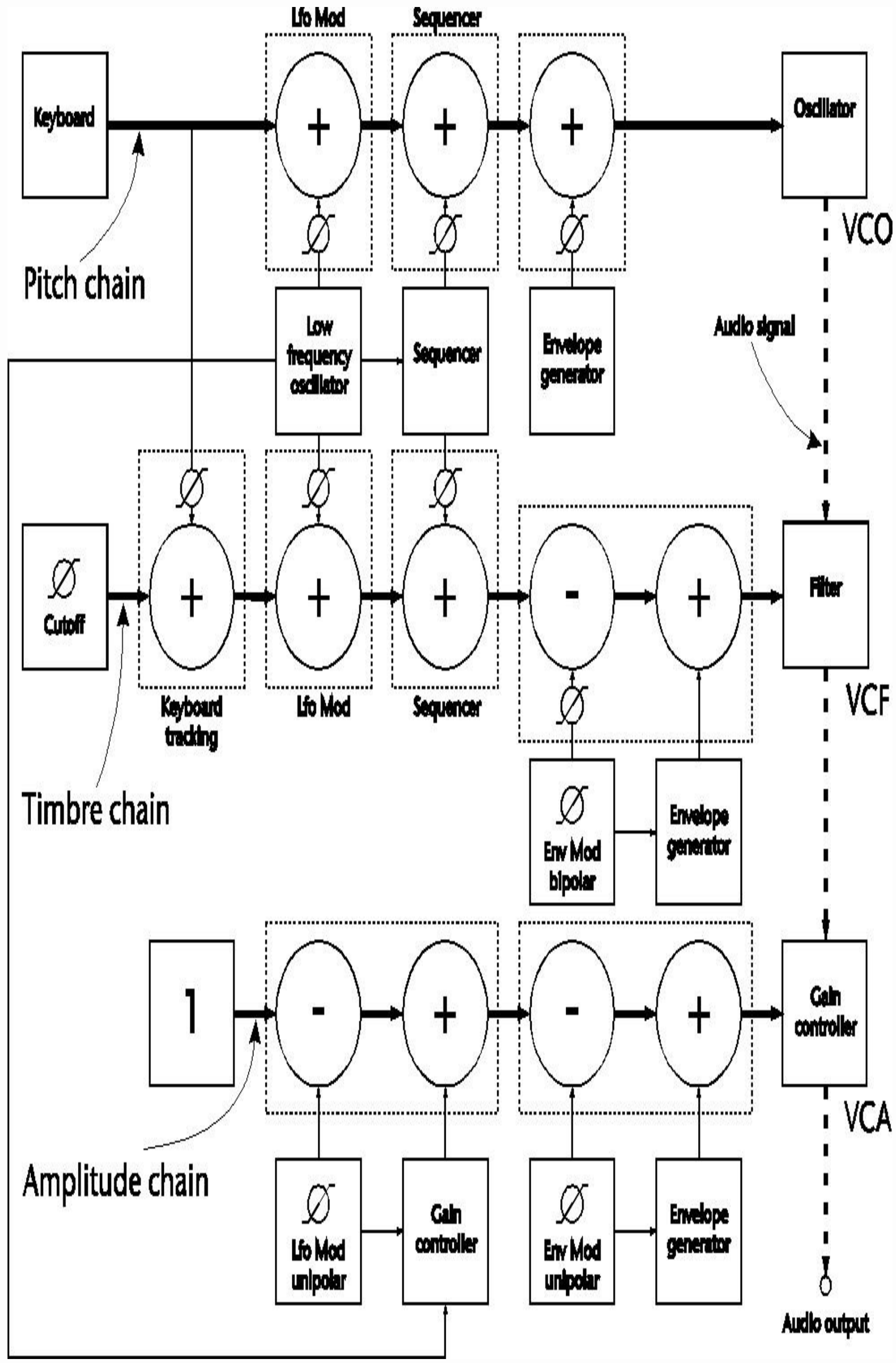


Figure 4 - Schematic of a VCO->VCF->VCA model

Figure 4 shows the schematic of how such a model could be patched. Note that the schematic does not say anything about oscillator waveforms or filter types, instead the sole purpose of this schematic is to show how expressive modulations can be applied to the musical parameters pitch, timbre and amplitude. The three fat horizontal arrows symbolize the three mixing chains. The pitch chain takes the keyboard note as its reference. Then, certain amounts of LFO modulation, a sequencer pattern and a frequency sweep envelope can be individually added along to the reference. The timbre chain takes the filter cutoff knob as its reference. When the cutoff reference value falls through the chain, certain amounts of keyboard tracking, LFO modulation, the sequencer pattern and a bipolar envelope sweep are added along.

Musically the model in Figure 4 is versatile and expressive. There is variable keyboard tracking for the filter. Note that, as the reference for the filter is the cutoff knob and not the keyboard note, causing the keyboard tracking to modulate around the cutoff frequency. The same goes for the sequencer signal, increasing the amount of sequencer modulation on the filter will also sweep around the cutoff. The type of mixing for the envelope modulation, which was explained earlier, will also cause the envelope sweep to appear around the cutoff frequency. In practice all modulations happen around a certain brightness level, which can be set with one dedicated knob. Changing a modulation amount will not drastically change the perceived overall brightness when playing the sound, as this is nicely compensated for with the mixing tricks. Which simply means that all modulations can be tweaked safely while playing.

The amplitude chain takes the value of the arithmetic number 1 as reference. This value equals +64 units and must be interpreted as unity gain. Meaning that when no modulations are applied in the amplitude chain the gain controller will have unity gain, very convenient as now the model will always give full volume when no modulation is applied. Then some LFO modulation can be applied to give a tremolo effect and next the envelope modulation can be applied. To give a proper envelope when playing polyphonic notes on the keyboard the modulation depth should here be set to 100%, of course. So, when no amplitude modulations are applied there is the full amplitude on the output. This allows for a drone beat that is modulated by the LFO and the sequencer pattern. When the envelope modulation amount is slowly opened the envelope will gradually increase in effect until the full envelope effect. The same happens to the LFO, meaning that in this model the amplitude modulation can be tweaked between a LFO tremolo and an envelope, taking a full amplitude as a reference and not silence. The sort of expressive effects that can be seamlessly tweaked with this sort of control signal mixing are commonly used in various old skool, drone and electro music styles. Figure 5 shows how the patch could look like in the G2 editor screen. Note that with the schematic at hand it is now much easier to follow the signal flow in the actual G2 patch.

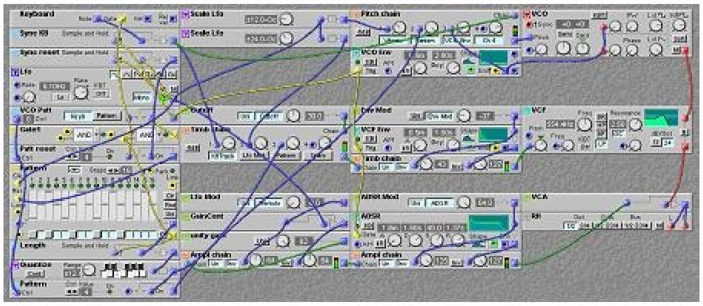


Figure 5 - The patch on the G2

Matrix mixing

Switch modules and mixer modules are closely related to each other. In fact, a multiple input mixer can be seen as a ‘superswitch’ where each input can not only be switched on and added to the output by fully opening its mixer knob, but the individual level can of course be set as well by tweaking the mixer knob. When a set of mixer modules are connected as an array they can together form a matrix with an attenuation knob and a mute button on each row/column matrix intersection point. This will create a ‘matrix mixer’, which is definitely the most flexible way to route signals between modules and set the levels of the signal paths. But the downside is that this will use a lot of computational resources. There can be redundancy in a matrix, which is basically a senseless connection. E.g. in many cases it is not very sensible to connect a module output to its own input, like when connecting the audio output of an envelope module to its own audio or trigger input. So, when patching a matrix mixer it is very important to analyse the matrix for possible reduntant crosspoint connections. he good thing is that these redundant crosspoints can be used for inserting extra signals in the matrix. This can save computational resources and reduce the size of the matrix. But it will also make the matrix slightly less easy to work with. Another approach is to use small submatrices that are connected to one small main matrix. In any case a lot of puzzling is almost unavoidable in setting up a matrix mixer. But after it is set up it will be a very fast and intuitive way of creating an enormous range of sounds, without having to repatch cables.

Summary

The examples in this chapter are to demonstrate that mixing is indeed a very important subject on a modular synthesizer, as it defines what sort of musical expression a patch allows. And while it is dead easy to connect the output of an oscillator to the input of a filter, some deep thinking might be required on how to mix several audio signals and modulation signals to provide for certain expressive musical effects. Experience is very important here, the more you experiment the easier it all gets. And luckily experimenting is a lot of fun. Just remember that the real art of sound design is in how things are mixed together and not really in what sort of a filter or oscillator is used. The secret of the experienced sound designer is that he blends the sonic aspects of the modules with the proper mixing techniques into that splendid sound, just like the five star chef does with the food. The good news for beginners is that corrections in a patch can always be made later; it cannot be permanently spoiled like a bad cook could spoil the food. Unless you forget to save and backup your patches, of course. Don't be worried when on a first try you can't get a sound exactly like you want it to sound, over time abilities will increase when experience grows and sounds will start sounding closer to what you have in mind.

Four different systems of mixing; absolute mixing, relative mixing, add along mixing and matrix mixing have been explained. The advice for now is to give these four types of mixing some very deep thought, until you feel you've got the hang of it.

Handling musical events in a logical way

Introduction to logic

There must be some logic in having logic modules in a modular synthesizer, and yes there is. Logic is quite similar to mixing, it is just that another type of signal is used and that the sort of methods used to combine signals in logic offers different sorts of manipulations as those found in mixing. The idea about logic is that there are signals that flag musical events. The most obvious event is the playing of a single musical note on the keyboard. The signal that flags to the patch that a key is pressed is named the keyboard gate signal. This signal is either on or off. In the chapter about signal types was described that the yellow and orange signals can have only two states, on or off. These on and off signals are the foundation of logic. So, logic is nothing else but the manipulating of signals that can be either on or off.

Keeping time

Next to the keyboard gate signal there is another commonly used type of logic signals named clock signals or timing signals. Clock signals allow for the timing of events, e.g. a sequencer has to advance steps in a certain rhythm and the clock signal will tell the sequencer when to advance a step. It is best to consider a clock signal as some sort of metronome signal. And like human musicians use a metronome to time their play, there are many modules that can listen to clock signals and synchronize their effect to the clock.

Signals that can be used as clock signals for other modules can come from several sources. Probably the most well known clock signal source these days is the MIDI masterclock. This is musically a very important signal, as it can synchronize several MIDI devices and computer programs to each other. In a MIDI set-up there can be only one device generating a MIDI masterclock signal, all other device must listen to this masterclock. In most situations a drum computer or a sequencer program running on a computer is used to generate the MIDI masterclock. When a MIDI clock signal is present and running on the MIDI input of the G2 there are several functions, like the arpeggiator, that immediately start to synchronize automatically to this MIDI clock. The tempo of the MIDI clock will show in the G2 Master Clock display in BPM. This display of BPM values of incoming MIDI clock signals on devices is not always shown absolutely accurate. It should be accurate on the device that is generating the masterclock, but for all the machines that slave to this clock the MIDI clock rate must first be measured before it can be displayed. When the masterclock is not completely stable, which can be due to heavy traffic over the MIDI cable, the measured BPM value in the display might flicker a bit. This is absolutely normal and the displayed value should only be seen as an approximation. It is only a quick indication of the tempo, and not the absolutely exact tempo value. To know the exact tempo value, always have a look at the display on the masterclock device. When there is no MIDI clock to slave to, the G2 has its own built-in masterclock generator that takes over automatically. In the patch a module can be placed that is named the Clock Generator module and this module has a setting that has two options, Master and Internal. When set to Master the clock output signals will slave to either the MIDI clock or the G2 masterclock, depending on whether a MIDI clock is sensed on the MIDI input. When the module is set to Internal the module itself generates a clock signal that is only available in the patch/slot itself and independent from the masterclock and clocks in other slots. To synchronize a patch to a MIDI masterclock from another device the Clock Generator module must be put in the patch and the module must be set to the Master setting.

Instead of using a dedicated clock generator signal a low frequency oscillator can also be used to provide modules with a clock signal. In fact even an audio oscillator can be used for a clock source to create hyper fast tempi effects.

A yellow or orange clock signal goes on and off and on and off, etc. In the on state the output signal is at a fixed level of +64 units and in the off state the level is fixed at 0 units. This means that the signal can be used directly to chop an audio signal when the audio signal is fed into one input of a gain controller module and the clock signal is fed into the other input of the gain controller. The zero unit level will shut off the audio while the +64 unit level will pass the audio at unity gain. This way the clock signal is used as an audio gate control signal. The signal is named gate signal as it can literally open or close a door or gate. And this door is either fully open or fully closed; there is no halfway open. A yellow input literally works this way, if the signal on a yellow input is zero units or negative it interprets the signal as a closed door. When the signal gets slightly positive, so the door opens just a little bit it interprets the door as being fully open. Just remember that with a yellow input there is no such thing as a half open door. This means that although a yellow output will produce a signal of either 0 or +64 units, a yellow input only looks if the signal is any positive value, which means door open, or if it is zero or a negative value, which means door closed. How a yellow input interprets its input value is very important to realize, as any signal type can be fed into this input. E.g., when a triangle LFO signal is connected to a yellow input, the input will think that the gate is on when the triangle is in its positive upper half of its waveform, while the yellow input thinks that the gate is off during the time the triangle is in its negative lower half of its waveform. So, any type of signal can by connected to a yellow input and will be interpreted by the module as a logic signal. In contrast it is the yellow outputs that can only produce the two on and off levels. This means that e.g. a LFO can rhythmically start an envelope module when the LFO signal is connected to the envelope module yellow input.

Combining logic signals

There are several situations imaginable where two yellow logic output signals need to be mixed in some way by combining them. An example is when a sequencer needs to be temporarily stopped for some beats and restarted later. In this case there is the clock signal that steps the sequencer and a second signal that defines when the sequencer runs or is stopped. This second signal is in essence a gate signal, as it can be used to open and close a gate where the clock signal has to pass through. It doesn’t really matter now where the gate signal comes from, as what is needed here is a module that acts as the door and can be opened and closed by the control signal. When taking a look at the logic modules tab in the editor program the leftmost module in the Logic tab is conveniently named a Gate module. This module is not a module that creates gate signals by itself, instead it is the ‘door’ module that can pass or block a logic signal depending on the on or off state of another logic signal. This module is the logic equivalent of a two input mixer, but acts specifically on logic signals. Like the two input mixer the two inputs are exchangeable and the module can do six combinations of operations on two logic signals. These operations should be looked at as various possible ways to mix logic signals. Mixing is not really the good word here; instead the word combining is better. The most useful combination is the AND function. This means that AND input one AND input two must have a logic on signal to produce a logic on signal on the output. This means that if one of the signals is off the other signal is effectively blocked in the module. So, this AND function is our door for the clock signal to be interrupted by the other signal, by feeding the clock signal to one input the signal on the other input defines whether the clock signal is passed on or is blocked. So, if the signal on the other input is the keyboard gate signal the clock will only be passed through the module when a key is pressed.

Working with logic signals needs some getting used to, initially it might be confusing, but in reality it is often very simple. Just remember that the signals on the input both have some musical meaning, like a clock signal stepping a sequencer or like a key press signal from the keyboard.

The six combinations are shown with little tables that exactly describe how the possible signals on the inputs will combine into a certain output signal. These tables are named truth tables, as 'they tell in logic truth' what will happen in the module. There are three basic functions, the AND the OR and the XOR. The other three are the same, only the output is additionally inverted after the function is applied on the two input signals. Study this table well and try to understand what happens with each function when on one input is a clock signal and on the other input is a keyboard gate signal.

Table 1: AND function

Table 2: NAND function

Table 3: OR function

Table 4: NOR function

Table 5: XOR function

Table 6: XNOR function

Triggers and triggering

Yellow inputs can react on the level of the input signal, but there is also the possibility that it only reacts on the moment the signal changes from off to on. This moment defines the exact instant when something starts to happen and there are modules that use only this moment to do something. A good example is the decay envelope module. This module will immediately start its envelope when the signal on its yellow input goes from off to on. But it doesn't do anything with the information of how long the gate signal is on. This behaviour is named edge triggering. Another way to say this is that the yellow decay envelope input is not gated but triggered. In essence there are two ways modules can use logic signals, as gates and as triggers. In practice it might be the same pulse signal, but when a module uses the starting edge or flank of the pulse it is named a trigger and when it uses the whole length of the pulse signal it is named a gate. So, it all depends on how the yellow input works whether to name the pulse on this input a trigger or a gate signal. When there is a little arrow drawn next to a yellow input the module will be triggered, or start its work on the moment the input signal goes to its on state and ignore the moment when the input signal goes off. The little arrow actually makes clear if the module is triggered. The ADSR envelope module is a clear example of a gated module, as on the ADSR module the envelope will remain in its sustain phase as long as the gate signal on the yellow input remains in its on state.

Summary

To summarize, logic is about musical events and about timing and clock signals. These events are in general represented by yellow signals. A yellow output signal can only have the two levels on and off represented by the values 0 and +64 units. But any type of signal can be used to connect to a yellow input and be interpreted as an on/off signal, where any positive value is seen as on, and zero plus any negative value is seen as off. Yellow gate inputs will react on the on level and how long the on level stays active, while yellow trigger inputs only react to the moment when the signal on this triggered input flips into an on state.

The manipulation functions of the modules in the Logic tab all relate to the processing and combining of timing signals, synchronization and signals that flag a musical event.

Conclusion

Sometimes two logic signals need to be synchronised, and example is when a key press must be delayed until a clock generator module flags a sixteenth note or the start of a new bar. The module that can help here is the Sample and Hold module. When the clock signal is connected to the trigger input of the S&H; module and the keyboard gate is connected to the value input of the S&H; the module will 'test' on every pulse of the clock signal if the keyboard gate is on or off. And if it is on it will pass it on to the output right at the triggering edge of the clock, so at the start of a note in the beat. By using a S&H; module this way the keyboard can be timed to the tempo clock and all notes pressed on the keyboard will be exactly in beat. This is especially handy when a sequence of notes programmed in a sequencer module needs to be transposed. By using a S&H; on both the keyboard note value and the keyboard gate signal and adding the sampled keyboard note value to the sequencer note value the transposition can be timed automatically to the beat. This method of using a S&H; module is an important technique to get things in sync, whenever there is a need to get things in sync and they don't do so automatically always remember to try to use a S&H; module to solve the timing issue. When the S&H; module samples a logic signal the effect is that the change of the logic signal is always delayed until the clock pulse on the trigger input of the S&H; arrives.

Sound sources

External sources

Two basic types of sound sources are available on a modular sound synthesizer; internal sources and external sources. An external sound source can be literally anything that produces sound, but for internal sound sources a specific reference is made to modules that are at the heart of a sound, in general the modules that are also responsible for the pitch of the sound. To be able to use external sources the synthesizer must have audio inputs. On an analog modular synthesizer the audio inputs of the modules expect signals that are much stronger than the line level signals that are produced by standard audio equipment like CD players. All signals that come from such equipment, and also signals generated by microphones, electric guitars, etc., need a preamplifier to be useful in an analog system. Many of the old analog modular systems offered a special external input module that would amplify the external signal up to a level where it could be used with other modules. On digital systems special inputs must be present that convert the audio signals into digital information, so the audio signals can be processed on the digital level.

Internal sources

In essence pitched musical sounds are a dynamically changing complex structure of repetitious waveforms with a certain pitch sensation, loudness contour and characteristic timbre. One single instance of a repetitious waveform is named a cycle. Many synthesis techniques simply try to produce and manipulate these waveform cycles. Mathematically every single waveform cycle in a short sound clip can be seen as the accumulation of a series of sine and cosine partials of certain amplitudes and by being able to handle these partials individually any conceivable sound can in theory be made. So, a single cycle of a waveform can be broken down into little parts, each part being a sinewave of a number of cycles that fits exactly into the 'space' of the waveform cycle of interest. It is a bit difficult to imagine how and why a waveform cycle should be broken down into these sinewave partials, the math to do this is pretty complex, but the reason is because the ear does really hear these partials. The hearing mechanism of the human ear translates the partial information in a sound into the sound sensation that the mind experiences, with all the sense of timbre, loudness, harmonicity and even the sense of recognition and meaning that sounds can have. By using the amplitude information about these partials the structure of the sound can be defined in the frequency domain, which is basically a description which partials will be present in a single cycle at a given moment in time. This means that every cycle can be descibed in a separate spectral plot. The whole sound clip can be described by creating a series of spectral plots, one for each consecutive waveform cycle in the clip. Next, all this data in the frequency domain can be converted into the time domain, which is the actual signal or recording. In essence the time domain describes the vibrations of the air pressure during the time the sound is actually heard. The method to handle all partials as single entities and control them in time is named additive synthesis. Regrettably this method needs so much data on all the possible partials for all the waveform cycles in a sound, plus that this data needs to be made available in a very fast rate, that in practice it is very hard to design an instrument using this additive synthesis method that allows to be played easily and expressively. Which means that in practice about all sound synthesis techniques used in musical instruments are about simplification, as the theoretically perfect additive synthesis technique is practically just too cumbersome to be implemented in an instrument. However, simplified models of additive synthesis, like in the drawbar organ, have become very popular. Still, the drawbar organ does not allow for precise imitations of acoustic instruments, as in general any simplification will imply a certain characteristic sound caused by the simplification. This will automatically put the instrument in a class of its own. The last statement is a very important notion, as many believe that sound synthesizers are designed to be used to imitate already existing traditional instruments. Which is a limited view, as the best one can ever get when imitating is a close approximation where the difference in sound between the real world instrument and the synthesized approximation adds a little characteristic of its own to the imitation. In general it is better to see electronic music instruments in a distinct class of their own, just as these instruments can be so much more that just imitators. In fact, when an electronic instrument can do imitations well, it can most certainly do proprietary stuff even better.

To overcome the need to handle the big bulks of data in additive synthesis the analog modular synthesizers from the sixties have used the subtractive synthesis method. The popular conception is that this method does not build brick by brick but tries to take the opposite approach by using a signal that contains at least all the partials needed and later simply remove what is not needed in the final sound. The modules that are used as the primary sound sources are named oscillators. Oscillator modules will provide the musician with a tuneable, single pitched raw sound with a static and in general very rich timbre that lends itself well for filtering. There are several types of oscillators, each optimized for certain fields of application. All oscillators have at least an input for a control signal that will define the pitch of the sound signal it will produce, plus at least one output where the sound signal can be taken from. Depending on the type of oscillator there can be one or more extra inputs for specific modulation purposes. Some oscillator types even need an audio signal from somewhere else before they can produce anything, an example is the type of oscillator which is commonly used in a technique named physical modelling or waveguide synthesis.

The advantage of using oscillators and filters is that they suit the earlier described exiter/resonator model very well, as in this model the oscillator will function as the exiter. A resonant filter that removes what is not needed and emphasizes what characterizes the sound will function as the resonator for the oscillator signal. When making a comparison to a violin the oscillator relates to the string and the bowing action, while the resonant filter relates to the wooden violing body acting as the resonance box.

Common oscillator waveforms

There are two commonly used waveforms which are very simple to generate and that have the very rich sound that is useful to be filtered later. These two waveforms are named the sawtooth waveform and the pulse waveform. When plotted graphically, the sawtooth waveform may rise up or slope down, but the human ear does not notice any difference if it slopes up or down. Still, it can sometimes make a difference if a sawtooth waveform slopes up or down when it is processed later. Sonically the sawtooth sounds very rich and bright. When two or three sawtooth oscillators are closely tuned to create an unison effect, and their mix is filtered with the right sort of filter, a very rich sound with a spacious, reverberant character is created. Exactly this sound was very easy to patch on the first modular system designed by Bob Moog halfway the sixties, and has become one of the hallmark sounds of the synthesizer. This type of sound is still very popular in dance music where it is the foundation of a thick unison sound named a hoover. This hoover sound is thickened even more by a chorusing unit and then played in a dramatic way with lot's of pitchbend at the start of the notes.

The sawtooth waveform contains all the possible harmonics of the pitch it is tuned to. Which makes the sawtooth an ideal waveform to be filtered, as in a sense the basic timbre of a sawtooth is neutral.

The pulse waveform is a signal that is basically only on or off, in this respect it is similar to a binary signal. There is a ratio between the time that the signal is on and the time it is off, this ratio is named the pulsewidth and can be expressed in a percentage. The pulsewidth has a pronounced effect on the basic sound of the pulse waveform, if the pulse is perfectly symmetric, meaning that the time it is on is exactly the same as the time it is off, the sound has a distinct hollow character, a bit similar to the hollow sound of a clarinet. Such a symmetric pulse waveform, where the pulsewidth is exactly 50%, is named a squarewave. The important property of this 50% pulse waveform is that it has only the odd harmonics of the basic pitch present. It is the absence of even harmonics that creates this typical hollow sound. The moment the pulsewidth is changed from 50% to a smaller pulse some of the even harmonics will return. There are pulsewidth settings where other harmonics disappear, e.g. when the pulsewidth is 33.3% the third harmonic will disappear but the second harmonic will be significantly present. On virtually all analog synthesizers the pulsewidth can be controlled dynamically, a feature named pulsewidth modulation. Every pulsewidth setting has a different harmonic spectrum, and a very lively effect is created when the pulsewidth is dynamically changed. This pulsewidth modulation effect sounds close to the unison effect of two closely tuned oscillators. Common methods to modulate the pulsewidth are by using a low frequency oscillator set to a triangle waveform pitched to around 1 Hz. Another common trick is to use an AD envelope with a fast attack time and a decay time between 300 msec and 1 second to smoothly glide the pulsewidth from e.g. 20% to 50%. When this same AD envelope is also used to sweep a lowpass filter which filters the oscillator signal, the typical snappy sound is produced that was often used in the sequenced or arpeggiated synthlines in the electropop genre of the eighties.

Filtering

After generating a basic signal by one or more oscillators, one or more filters can do the removal of all unwanted partials. The quality and controlling possibilities of the filters define how accurate the method will be in practice. But to be theoretically perfect the filter would have to be so complex and need so much dynamic control data that probably the same amount of data would be needed as when using additive synthesis. Again simplifications are made. In fact subtractive synthesis as used in 'analog' synthesizers and their 'virtual analog' digital equivalents can better be seen as a form of formant synthesis where resonant filters are used for the purpose to create a strong but easily controllable formant at the resonant frequency of the filter. The reason why the sawtooth and pulse waveforms are used as the raw material to be filtered has much more to do with how these waveforms excite the resonant filters than with the spectral content of the waveforms. The sharp transients in the waveform, these are the flanks in the waveform plot where the level suddenly changes from one extreme to the other, are what 'fires' the resonance in a resonant filter. Transients contain an enormous amount of 'energy'. They have to, as when such a waveform directly drives a speaker, this is the moment when all the mass of the speaker has to be moved from one extreme to the opposite extreme. In the resonant filter this energy is transformed into a 'ripple' lagging the transient in the waveform, which creates a strong formant at the resonant frequency of the filter. Sweeping the resonant frequency of the filter creates a musically expressive sweeping formant with only a single parameter to be controlled. More expressive results can be obtained by sweeping two or more formants, at the cost of extra filters and controllers.

An important experiment to gain some more insight into this matter is to connect a sawtooth oscillator directly to the output of the synthesizer. Don't set the volume of the amplifier too loud to avoid damage to your speakers! Now set the pitch as low as possible, somewhere around 1 Hz is perfect for this experiment. If the oscillator cannot go this low a low frequency oscillator (LFO) with a sawtooth waveform or a pulse waveform can be used as well. When the sawtooth is at 1 Hz only clicks are heard, in fact one click per second. In between clicks there appears to be nothing. Now put a resonant filter in between the oscillator output and the synthesizer output and set the filter cutoff frequency setting to somewhere around 1 kHz and slowly open up the resonance knob. The click will gain more timbre and slowly transform into a percussive 'claves' sound. The short 1 kHz resonance ripple lagging the transient can easily be heard. This little experiment proves that it is indeed the transient which fires the resonance in the filter. Now slowly raise the pitch of the oscillator and you get an idea of what a sawtooth waveform does to a resonant filter and how both act together to create the final timbre.

The sawtooth waveform is extremely easy to generate by both analogue and digital circuitry. In an analog sawtooth oscillator the waveform is created by charging a capacitor, an electronic component that can be charged by a current, similar to the rechargeable batteries in modern mobile phones. In comparison to a battery a capacitor can store only very little charge, but a capacitor can be fully charged and discharged almost instantly. By gradually charging the capacitor at a controlled rate the voltage over the capacitor rises. When the voltage reaches a certain level a relay circuit like a switching transistor is used to instantly discharge the capacitor, after which it is slowly charged again, discharged, charged, etc. The gradual charging creates the rising slope of the sawtooth and the instantaneous discharging moment creates the flank in the sawtooth waveform. When the discharging is indeed instantaneous the pitch of the sawtooth will depend on the charge rate only. By controlling the charge rate by a knob or a control signal the pitch of the sawtooth wave can be precisely set. Discharging the capacitor will still take a little time on analog oscillators, an average of about 1 to 2 microseconds is not uncommon. As the discharge time is fixed it will make the frequency behaviour of the oscillator slightly non-linear, which can sometimes be corrected by a trimmer control named ‘high frequency tracking’.

The relationship between the charging current and the generated frequency of an analog sawtooth oscillator is linear, doubling the current will double the frequency. The ear however perceives frequency in an exponential way, it ‘hears in octaves’. This means that a frequency perceived by the ear as three octaves higher than another frequency, has an actual frequency that is eigth times higher when measured in Hertz. The calculation here is simple, raise the number 2 to the number of octaves of the pitch transposition and the result will be the amount that the actual frequency in Hz is raised to, in the previous example 2^3=8. The analog sawtooth oscillator needs a circuit to easily transform the equally tempered scale note data from a keyboard into the correct charging current for the capacitor. This device is named an exp/lin converter. The synthesizers built by Moog in the sixties used a 1Volt/Octave translation in the exp/lin converter to drive the oscillator and this has become the de facto standard for analog synthesizers. The circuitry that does this conversion can easily drift on changing temperatures and temperature compensation must be built in. The quality of analog oscillators depends largely on the temperature drift behaviour, the accuracy of the exp/lin converter and the presence of a proper high frequency tracking trimmer control. As these three factors must be implemented with top quality components, causing good quality analog oscillators to be costly.

The digital sawtooth oscillator algorithm is incredibly simple, in essence it is just a single addition instruction in the DSP chip. By repeatedly adding a certain fixed value to a register the value in the register will increase, just like the charge in the capacitor increases by the charging current. At a certain moment the register will overflow and an overflow condition will be set in the DSP. If simple integer arithmetic is used and the register is allowed to simply wrap around on overflow it is not even needed to ‘discharge’ the register as this is implied in the wrap around. If the DSP does not allow for wrap around the register can be ‘discharged’ by subtracting the maximum value the register can hold. This can in many cases be conveniently done by an AND instruction with an operand that has all bits set. If floating point arithmetic is used a modulus function can be used to ‘discharge’ the register. Or alternatively rounding the result in the register to the nearest integer, which in this case will be the number one, and subtracting the rounded result from the value in the register. In this particular case the value to be added must be a fractional value between zero and one. The preferred way to implement the digital sawtooth is by using 24 or 32 bit integer arithmetic, running at a sample rate of 96kHz or higher and allow for wrap around. It's the simplest, most efficient and fastest method. It also allows for a frequency parameter with a 'negative' value, which will produce the waveform in antiphase. The integer result of the addition can instantly be used to scan waveform tables and read and write index points in delay lines, but to be able to use the result as an audio waveform it must be bandwidth-limited and probably rescaled to get the best sound quality. Bandwidth limiting is necessary as the digital sawtooth is actually too perfect. Let's assume a sawtooth at a pitch of 100 Hz is generated at a sample rate of 96kHz. Not only the audible harmonics up to 20kHz will be present but a lot of harmonics above the hearing range will be present as well. Between 20kHz and 48kHz there are 280 harmonics present at 100 Hz intervals. These very high harmonics are up to no good, as they can intermodulate with the sample rate and cause an audible distortion named aliasing. Audibly the best sound is achieved when all possible harmonics above 20 kHz are not present at all and the harmonics between 5kHz and 20kHz gradually decrease in energy. The best thing is if the harmonics above 20 kHz are not generated at all by the algorithm used in the sawtooth oscillator. This will make the algorithm for a good audio quality sawtooth oscillator much more complex than the just described accumulation method.

With the sawtooth signal a lot of things can be done, in fact most synthesis methods use a sawtooth signal at their heart to drive their synthesis engine. On both the traditional and on the virtual analog synthesizers it can drive a resonant filter very well. But the waveform can also be manipulated in a more 'constructing' way to obtain different waveforms with specific desirable properties. E.g. the pulse waveform is constructed from the sawtooth waveform. The way to do this is by comparing the level of the sawtooth waveform to a fixed or slowly varying control signal and providing an output signal that is either on or off, depending on whether the pulsewidth control signal or the sawtooth signal has the highest momentary value. The circuit that can do this type of comparison function is named a comparator and the output of the comparator circuit is the pulse waveform. This comparator circuit is commonly built into the oscillator and provides an extra output with the pulse signal. A triangle waveform is also constructed from a sawtooth waveform by folding down the upper half of the sawtooth waveform. From this triangle a sine wave can be constructed by passing the triangle through a device with the right non-linear function, in cheap synthesizers two diodes or more properly a more expensive circuit using a balanced modulator. In a digital oscillator these pulse, triangle and sine waveforms can be derived from a sawtooth in a similar way. There are basically two methods. The first method uses the sawtooth signal to scan a wavetable, a small part of memory where a 'graphic' representation of the waveform is stored. The second method uses functions to construct the other waveforms, e.g. a simple compare instruction in the DSP can create the pulse waveform from the sawtooth. But a good quality digital pulse waveform will need bandwidth limiting just like the sawtooth. The triangle wave can be constructed with some more instructions and from the triangle waveform a sine waveform can be constructed by using suitable mathematical functions, some of which can be executed quite efficiently. There are other ways to generate these waveforms directly in a digital system, but going into these details is beyond the scope of this book.

There can be a little difference in sound between similar waveforms on an anlog system and a digital system. Analog systems are said to have a warmer sound and digital systems are said to sound more brilliant in the very high. These are in many cases quite subjective differences, it all depends a lot on the quality of the analog oscillators, the bandwidth limiting of the digital oscillators and the quality and bandwidth of the DA converters used in a digital system. The main issue here is the area between 10kHz and 20kHz, analog oscillators tend to have a little less energy in this very high part of the sound spectrum. Most analog circuitry is bandlimited to about 5khz to 10khz to fight analog noise. Filtering away this area on a digital system can make it sound warmer and additionally less conflicting with e.g. cymbal sounds or the 'air' in vocals.

Waveshaping

Roughly there are three types of manipulations possible on a waveform on the oscillator level. First, the oscillator output can be modulated in amplitude by passing the output signal through a controllable amplifier or multiplier.

The second manipulation is to modulate the waveform in time by smoothly shifting the waveform forwards and backwards in time. This will compress and expand the waveform in a rhythmic manner and when done at audio rate it creates new partials in the sound. The third possibility is to 'make a jump in time' by prematurely restarting the cycle of the waveform. These three techniques are respectively called amplitude modulation or AM, frequency modulation or FM and oscillator synchronisation or sync. If the aim of these techniques is to create a new waveform from an existing one it is common to talk about waveshaping. The purpose of waveshaping is to change the sonic properties of the waveform into other sonic properties that are special to the new waveform. Waveshaping can change the sound of a certain waveform dramatically, which means that it is musically a very interesting technique. In subtractive synthesis it is equally important as filtering, simply because shaping the waveform into a new waveform can remove certain aspects of the sound that are hard to remove with filters. Additionally, when a waveform can be shaped into another one in a smooth transition over time special musical effects can be created. Waveshaping can be present on many levels in a sound synthesizer. It can be used in an oscillator to create a new set of static waveforms from one reference waveform. But when the waveshaping is dynamic, it can be used to interactively and expressively play the timbre of the sound. Either under manual control or under control of a control signal from a modulation source, like a low frequency oscillator, an envelope generator, all sorts of sensors that can produce a useable control signal or a changing midi control signal received from e.g. a sequencer program running on a computer.

Amplitude modulation or AM

To control the amplitude of an oscillator an extra modulatable gain control module is needed after the output of the oscillator. On analog synthesizers the multiplication circuit is either a VCA or a ringmodulator, digitally it is a single signed or unsigned multiply instruction. VCA stands for Voltage Controlled Amplifier. The module has an audio input and a control input plus an output. Often there is a knob that sets the initial gain of the module. A ringmodulator has two identical audio inputs and one output. The main difference is that a VCA can modulate the audio signal by a positive control signal only, whenever the control signal is zero or becomes negative the VCA suppresses the output fully. The ringmodulator is a true signed multiplier, meaning that just like in an arithmetical multiplication it can accept both positive and negative values on its inputs and on the output is the arithmetical product of the input signals. In theory the inputs are identical and it doesn't matter which of the signals is fed into which of the inputs. But in practice there might be small differences, depending on the quality of the circuit. Ringmodulators that will accept both audio signals and fixed or slowly varying control signals are only found on the most expensive analog modular synthesizers. On the simpler systems a ringmodulator will in general only accept audio rate signals and block control signals on both its inputs. One of the big advantages of digital modular synthesizers over analog modular synthesizers is that the analog modulars invariably have a very limited amount of VCAs and ringmodulators. Analog modules are probably not very accurate, due to component tolerances that might be up to 10%. They also most likely exhibit leakage of controlling and modulating signals on the output. In contrast, the digital multiply instruction is at least accurate within the bit depth of the system and does not exhibit leakage. And as it is only a single DSP instruction many multiply operations can easily be done, although some scaling of the inputs and output might be necessary, this depending on the actual system.

For now let's assume that the multiplier is capable of handling both positive and negative values on both of its inputs. The multiplier can be controlled by a fixed value, which will change the volume level. A fixed control signal of negative polarity will bring the signal in antiphase. When a wildly varying control signal is used, like an audio signal, several sonically interesting things happen, as this can create new partials that are not yet present in either of the two input signals. The new partials can be harmonics of the original waveform, but can also be enharmonic partials. The multiplier can also be controlled by a signal that is derived from the input signal. This last case means that a transfer function is applied to the oscillator waveform. An example of a transfer function is when distortion is applied. E.g. in the case of a saturation distortion the input signal itself will 'control' the transfer function, the higher the momentary signal level the more saturation will be applied, caused by compressing the signal at audio rate. Note that there is a difference between the global sound level or volume and the momentary signal level, which is the momentary value of the signal at the extremely short moment named the now. Many complex mathematical functions can be made by using multipliers, mixers to do additions and subtractions and using constant values. In general these constant values are in a scale between arithmetically minus one and plus one. The transfer function is implemented as either a piece of programming code on a DSP system or on an analog system the patching of several ringmodulators and mixers.

As an example for a digital system, the function to generate a triangle waveform from a sawtooth waveform at amplitude 1 is to take the absolute value of the sawtooth times two minus one or ABS(Saw)*2-1. Using a Taylor-series function this triangle can be transformed into a sinewave. Chebyshev polynomials are well known functions based on taking sums of quadratures of sinewaves and rescaling to keep amplitude 1 results. They can be used to generate the harmonic partials from an amplitude 1 sine wave. If coded efficiently these functions can in many instances be faster than interpolated table lookup methods.

Basically any non-linear function can be used this way to amplitude modulate any audio signal, results may range from a great sound to totally havoc, but there are no rules, anything is allowed as in the end its all a matter of taste. In some cases there might be only one input and an output, meaning that the effect will fully depend on the signal level of the input signal. In other cases there might be knobs for controllable parameters that allow for dynamic timbre control. When the waveshaping technique is fully mastered an enormous range of basic sounds becomes available, many of them allowing for intuitive and expressive play.

Frequency modulation or FM

Frequency modulation is based on shifting the waveform smoothly backwards and forwards in time at audio rates. To do this, another waveform is required to control this dynamic shift. The waveform to be modulated is named the carrier wave generated by the carrier oscillator. The waveform that is used to modulate the carrier oscillator is named the modulator. The modulation process can be applied at several points in the carrier oscillator, both the exponential frequency value and the linear frequency value can be modulated. The change in frequency will in effect cause the timeshift of the waveform. On a digital system there is also the possibility to modulate the phase of the waveform in time. Applying the modulation to the exponential frequency input can quickly create enharmonic results, so using this input is of less practical value. Within an analog oscillator linear modulation can be implemented by adding the modulating signal as a current to the current that is charging the timing capacitor. Regrettably it is difficult to do this in a a accurate and stable way, so only few of the more expensive quality analog oscillators offer an input for linear frequency modulation that also works properly. Digitally it is no problem at all, the momentary value of the modulating waveform is simply added to the linear frequency value on the output of the exp/lin converter. To avoid enharmonic results the ratio between the frequency of the modulating wave and the frequency of the carrier wave should be kept constant in simple ratios like 2:1, 3:2, 5:2, etc.

FM modulation index

The amount of modulation applied is denoted by the modulation index. The value of the modulation index is the frequency deviation of the carrier divided by the frequency of the modulating waveform. If this ratio is constant, meaning that the modulation index is constant, a waveform is produced that has the same harmonic spectrum for all musical notes. This harmonic spectrum depends also on the phase relationship of the carrier and modulator, so preferably these should be locked in phase to get a stable waveform with a stable harmonic spectrum. Phase locking must be used to get predictable results. On a digital system phase locking between oscillators is much easier to implement than with two analog oscillators, which is one of the reasons why in practice all synthesizers that use any form of FM to generate their sounds are digital systems. In most FM synthesizers the system itself takes care of phase locking between oscillators, so it is of no concern to the musician to delve deeper into this matter.

To get a better understanding of what the modulation index really is, imagine a sinewave with a pitch of 500 Hz, which is modulated by a square wave at a very low frequency of 1 Hz. This will result in two steady tones alternating at a rate of two tones a second. The two tones are pitched around 500Hz, one tone is higher when the modulating square is at high level and the other tone is lower, due to the modulating square being at a low and negative value. The frequency shift of the two tones compared to the 500Hz is equal, but one of the tones has a negative frequency shift giving it a lower pitch. Let’s assume that the two tones have a 100 Hz shift. This results in one tone at a lower pitch at 500 - 100 = 400 Hz, and the other tone to have a higher pitch at 500 + 100 = 600 Hz. This 100 Hz shift is named the frequency deviation, it tells by how many Hz the new pitches deviate from the original pitch. As the linear frequency parameter is modulated, this shift of 100 Hz depends on the signal level of the modulating waveform. When the signal level of the squarewave is increased the two pitches will deviate further away from the 500 hz. But if this signal level remains the same all the time, the frequency shift for a 1000 Hz sinewave will also be 100 Hz and this would result in two tones of 900 Hz and 1100 Hz. And here is the catch, the shift at 500 Hz pitch is 20% as 100 Hz is 20% of 500 Hz. But the shift at 1000 Hz is only 10%, which you can imagine is up to no good. In fact the basic trick in FM is to create a constant percentage of shift and use this as the reference to manipulate the modulation depth. By remembering that the percentage of shift should by default be constant the FM technique can be better understood. And this percentage of shift is in fact directly related to the modulation index. It happens that the amount of frequency shift can be expressed as the frequency deviation of the carrier divided by the frequency of the modulating waveform, which results in nice and easy to work with numbers. In the case of the low frequency squarewave it was easy to imagine how it works, as only two distinct pitches are produced. When instead of the squarewave a sinewave is used the frequency glides smoothly between two extremes instead of jumping from one pitch to the other. In this case it is the maximun frequency shift caused by the sinewave that is used in the formula. So, if the resulting pitch glides between 400 Hz and 600Hz the deviation is 100 Hz up and down compared to the 500 Hz pitch when no modulation is applied.

It should be clear that to keep the modulation index constant the amplitude of the modulating waveform should be corrected for each pitch on the musical scale. Luckily the relation between the overall amplitude of the modulating waveform and the pitch of the carrier is very simple, it suffices to multiply the modulating waveform amplitude by the original linear frequency parameter on the carrier oscillator before it is added to the internal carrier frequency parameter. If this condition is met, increasing the amplitude of the modulating waveform will simply brighten the timbre to a richer sound and create a similar type of timbre control as sweeping the resonance frequency of a resonant filter. Which effectively creates a single expressive parameter that can be easily played by a controller like a knob or a modulation wheel. And the timbral effect tracks the keyboard in the same way as a filter can track the keyboard. On analog oscillators there are actually two points in the exp/lin converter circuitry where linear frequency modulation can be applied, and one of them has the inherent property to keep the modulation index constant over the pitch range. The other point keeps the deviation constant, which results in a strong formant that stays fixed to a certain frequency area. This can give a nasal effect to the sound when it is played over several octaves. By applying a little of the modulating waveform to both these modulatable points the keyboard tracking effect can be steplessly set between no tracking and 100% tracking. On an oscillator in a digital system there might be a button to choose between tracking and no tracking. Additionally FM synths have a feature that is named keyboard scaling or level scaling which can be used to control the keyboard tracking of the timbral effect of the FM modulation.

The modulating waveform can be basically any waveform, but for the carrier oscillator it is best to use a waveform without any strong transients, as these transients can get shifted in and out of the resulting waveform. Which might in cases sound quite harsh. An exception is when a square wave is used as both the modulator and the carrier and a deep modulation is used, this will have the effect of a deep and bright pulsewidth modulation effect. The sine wave and triangle wave seem to always perform very well as carrier wave, but the sawtooth waveform is definitely tricky for a carrier.

Phase modulation

On a digital sawtooth oscillator not only the exponential and linear frequency parameters can be modulated, but instead the actual output can be phase modulated by adding the modulating signal directly to the output signal and applying a 'wrap around' or modulo function on the result to make the resulting waveform fold back to the minus one to plus one signal level range. This sounds equal to when a frequency parameter is modulated. From the phase modulated sawtooth waveform other waveforms can be derived by the proper waveshaping functions or the table lookup method mentioned before. When the phase is modulated and the modulation index must by default remain constant, it is again possible to multiply the modulating waveform with the internal linear frequency parameter from the oscillator, before the actual modulation is applied.

Modulation at audio rates of the phase of sinewave oscillators was explored in the sixties by Chowning. Later the japanese synth manufacturer Yamaha would use Chowning's work to build their hugely successful DX7 FM synthesizer and the whole range of FM synths that followed. The advantage of phase modulation on a sinewave over modulation of the frequency parameter is that if selfmodulation is applied, meaning that the carrier wave is routed back to its own modulating input, there will be no unwanted pitch shift if the modulation amount is increased and the oscillator remains neatly tuned. Increasing the depth of the selfmodulation will gradually change the sinewave into a sawtoothlike waveform and with even deeper modulation will force the oscillator into a chaotic range that sounds like white noise. When instead of the selfmodulation of the phase, selfmodulation of the frequency parameters is used the basic pitch will drift away. This drifting away of the basic pitch is due to an inherent increase of a DC component in the modulated output signal, which will bring the oscillator badly out of tune. A workaround to this drifting away is to use a high pass filter on the modulation input. But even a simple 6 dB high pass filter tends to oscillate at a very high frequency if it is fed back, even through the carrier oscillator, and this will make the carrier oscillator unstable at higher modulation levels and not produce the proper chaotic behaviour. The rules are that when the pitch of a FM modulated oscillator should remain the same and selfmodulation is applied, only phase modulation should be used. But to create chaotic and noise sounds it is sometimes better to selfmodulate the linear frequency modulation input.

This chaotic range is quite interesting to explore, to get much better results in this range a lowpass filter can be inserted in the feedback patch, the steeper the filter the more interesting the chaotic waveforms that result. Other filter types like a variable width bandpass filter can give very good results as well. The variable width bandpass filter works very well because the highpass part will prevent a pitch drift and the lowpass part will give more control over the brightness of the chaotic range and prevents the highpass part to oscillate at a very high frequency.

Zero Hertz FM

An interesting case of FM is when the carrier oscillator frequency is set to zero Hertz by using a value of zero for the linear frequency parameter. This will in fact stop the oscillator. This technique can only be done on digital oscillators that can also be set to a negative frequency by negating the frequency parameter which should bring the oscillator output waveform into antiphase. Applying a modulation signal to a linear frequency parameter which tracks the keyboard will rhythmically start, stop, start in antiphase and again stop the oscillator. The musical importance of this frequency modulation of a zero frequency carrier oscillator is an audio signal that will always inherit its pitch from the modulating oscillator and has a strong formant area in its formant spectrum which location depends directly on the modulation index. Rule of thumb is again that the sound brightens if the modulation depth is increased. Frequency modulation of an oscillator at a 0 Hz pitch can never produce enharmonic results if the modulating signal isn't already enharmonic. Using a square wave as modulating waveform will produce the timbral result of an analog technique named softsync.

This is an interesting example of frequency modulation, as although the frequency parameter is modulated the pitch will always be the pitch of the modulation waveform. In this respect the technique behaves much more like waveshaping done with amplitude modulation. Later there will be practical examples of amplitude modulation techniques where a steady detuning effect is created byt the waveform remains the same. In fact amplitude modulation and frequency modulation are intimately related, and both can shape a waveform at a fixed frequency and additionally amplitude modulation can change a frequency without changing the waveform.

Oscillator synchronisation

Oscillator synchronisation lets an oscillator restart its waveform in synchronization with another waveform. Analog oscillators that are capable of synchronizing commonly use the flank or transient from another waveform waveform to synchronize to. In such an oscillator a circuit named a transient detector generates a very small pulse that is used to prematurely discharge the capacitor. This implies that on an analog sawtooth oscillator the synchronized sawtooth restarts with the maximum negative value from where it ramps up. On digital oscillators it is common to restart the waveform at the upward zero crossing point. It is also common to let the oscillator synchronize on an upward zero crossing point in the synchronizing waveform. To detect this zero crossing point the current sample is compared with the previous one and if the current one has a positive value and the previous one a negative value the zero crossing point is detected. At this moment the register that holds the current sawtooth waveform value is filled with a certain value instead of doing the addition.

Oscillator synchronization introduces a new flank in the synchronized waveform at the moment it is synchronized. This makes the current level change to a certain fixed level of either zero or the maximum negative extreme value. This ‘sync’ transient is very audible and the characteristic timbre effect of a sync sweep is caused partly by the changing magnitude of this transient. On waveforms like the sawtooth this magnitude changes gradually and doesn’t contrast too much with the timbre of the original wave. But with sine and triangle waves the contrast is greater and doesn’t always sound very well. In many cases the sound can be improved dramatically by suppressing this transient by applying an envelope over the waveform. This envelope is called the mask and the technique is named masked sync. The mask must be synchronized to the synchronizing waveform, so it is obvious to construct the mask from the synchronizing waveform. It is preferrable that if the mask is applied the gradient at the start of the next cycle is equal to the gradient of the previous cycle, but this depends a bit on the waveform to be synchronized. If this waveform is a sinewave it is best to use the first half of a bell-shaped curved mask, if it is a sawtooth, a square or a triangle wave a simple downward slope can be used for the mask as well. This downward slope can easily be derived from a rising sawtooth by applying the function $x' = -0.5 * x + 1$. In other words by inverting the sawtooth, halving the amplitude and adding just enough fixed voltage to make the result a positive only signal. On an analog synth the VCA can probably be modulated by an amount control that fades between full modulation and full signal, so in many case it suffices to invert the sawtooth, feed it to the VCA modulation input and set the amount control half open and tune the sound by ear. To get the half bell-shaped mask the sawtooth can be soft clipped by maybe a log-type function before it is converted into the mask.

Combinations of AM, FM and oscillator sync

The three basic techniques can be combined in all possible ways to create even more waveshaping possibilities. As an example an expressive waveshaping oscillator can be patched by using two synchronized sinewaves and multiplying them before a half bell-shaped mask is applied to their multiplied result. The gradient of a sin^2 wave is zero degrees at the start of its cycle, and multiplying the two sinewaves also gives a gradient of zero degrees as the startpoint of the cycles are synchronized to the synchronizing sawtooth oscillator. In this case it is the synchronized waves that produce the zero degree gradient at the start of the cycle and the mask that causes the zero degree gradient at the end of the cycle. Setting the two sinewaves to different frequencies above the frequency of the synchronizing sawtooth oscillator that supplies the mask will create a timbre with an expressive character with two distinctly audible sweepable formants. The sinewaves can be manipulated before or after they are multiplied together, but before the mask is applied. Transfer functions like a sin^3 or a sin*abs(sin) function perform very well to make the timbre even more talkative. Applying some heavy saturation distortion can add a lot of beef to the resulting sounds as well. Using a joystick or any other X-Y controller to offset the frequencies of the synchronized sine waves allows for very expressive timbre control. Envelopes and LFOs can be used equally well to create slowly evolving timbre changes. Applying FM on the two sine waves can also give expressive results. This waveshaping oscillator can easily be patched on both analog and digital modular synthesizers. On an analog modular one sawtooth and two syncable sinewave oscillators are need plus a single ringmodulator and a VCA with a level control an d level modulation. On a digital modular these modules will be plenty available and even more complex variations with more sinewave oscillators can be patched in various configurations.

Another interesting example is when syncing a pulsewave to a sawtooth wave and again using the sawtooth wave as a mask over the pulsewave output. By applying pulsewidth modulation and routing the sawtooth wave also to a linear FM input on the pulse oscillator several interesting timbres with smooth changes in brightness can be made.

Feedforward and feedback

Feedforward

Connecting two cables at an output of a module will create two separate signal paths coming from this output. This effectively splits a signal into two parallel signals, that at this point are exact copies of each other. The reason one might want to do so is to give one or both signal paths a different sonic treatment. After these sonic treatments the two manipulated signals can be mixed together again. The final result will be an effect that is a combination of the two effects applied to the signal after it was split up into the two copies. This technique is named feedforward and is an important technique to create subtle effects. Using feedforward techniques is often the way to get more control over effects that are applied to already recorded sound tracks or sound samples and loops. An exampe is to first feed the copies through bandfilters set to two or more different bands and then giving different effects to these different frequency bands in the audio signal.

Spitting a signal into two signals, manipulating the two copies and then mixing them together does not necessarily have to create a simple ‘addition’ of the two applied effects. It is better to try to imagine how the two manupulated signals will later interfere with each other. E.g., it is possible to ‘subtract’ one effect from another effect by inverting one of the signals before they are mixed together. This creates an interference that is defined by the difference between the two applied effects. A good example is when one of the signal paths is filtered with a simple, non-resonant lowpass filter and inverted in phase, while the other signal path is left unaltered. Because what the filter is passing on is subtracted from the original signal, the output of the mixer will be a signal that contains ‘everything what the lowpass filter threw away’ from the original signal. Which are the high frequency partials in the original signal. So, this technique effectively creates an extra highpass filtering output in addition to the lowpass output of the filter. The result of this feedforward operation is that the single output lowpass filter is basically changed into a two output crossover filter effect that splits the audio spectrum into two bands. However, the highpass slope will not be of the same steepness as the lowpass slope. In practice this is not much of a problem, as the human mind will not perceive the steepness of a highpass slope as pronounced as it perceives the steepness of a lowpass slope. More important is that when the lowpass signal and the highpass signal are added together again, there will be an exact copy of the original signal. This last step might appear senseless, but when another effect like e.g. a distortion is applied to either the lowpass signal or the highpass signal before they are mixed together again, this distortion will only work on the chosen part of the spectrum and not on the other part. There are many sound manipulations that work best when they are only applied to one part of the sound spectrum. A good example is a chorus effect, which is best applied on the mid part of the sound spectrum, as chorus on low bass notes will easily sound muddy and in the very high parts of the spectrum the chorus might kill what is named ‘air’. Too much chorus in these parts of the audio spectrum will make the sound loose its definition. Another example is when odd harmonic distortion is applied to a signal. It is often best to limit this distortion to the band below 2.5kHz, in which case the distortion will seem to enhance the presence of the sound in a mix. It will also keep the high part of the sound clean of ‘intermodulation’, an effect where the high frequencies seem to be amplitude modulated by the lowest frequencies in an unnatural sounding way. In these crossover filter examples it is important that the passband of the lowpass filter passes the signal at unity gain and the mixer inputs have exactly the same sensitivity. This could pose a problem on an analog system, but with the precision of a properly designed digital system this technique works quite well and is very effective and of great practical use.

Another case of feedforward is when one or both of the signal paths are given a well-defined time delay or a phase shift. A phase shift is a frequency dependent time delay with different delay times for different partials. When the signals are mixed together later, the interference can create a dramatic change in the timbre of the processed sound. This is caused by the fact that some partials will be delayed in a way that they will become in phase with the same partials in the other path, and so boost these partials in the final mixing. Other partials might be delayed in a way that they will be in anti-phase with their counterparts in the other path, and so cancel each other out and simply disappear into thin air. The basic purpose of this type of feedforward can be understood as creating an interference effect on the different partials in the sound, with the aim to change the timbre. The interference can be made to be dynamic, meaning that the manipulation of one of the copies of the original signal is controlled by a continuously evolving modulation signal. This will make the resulting interference pattern change in a lively way. Many popular effects like chorusing, comb filtering, flanging and phasing are based on this principle. This technique is also very interesting when two filter modules are placed in parallel. An example is when.two lowpass filters are patched in parallel and their outputs are subtracted in equal parts from each other, which will create a bandpass filter. The advantage of using two parallel lowpass filters over using a lowpass and a highpass filter in series is that it doesn't matter which filter is tuned higher, there will always be a passband between the two set cutoff frequencies. Additionally it is possible to morph from a lowpass response to a bandpass response by slowly fading in the output of one of the filters on its final mixer input. With a lowpass/highpass combination in series it works out a bit different, as when the highpass filter is accidentally tuned higher as the lowpass filter most of the sound will disappear, while with two parallel lowpass filters the sound will not disappear but simply reverse in phase. Also, morphing from lowpass to bandpass is less straightforward with a lowpass and highpass in series, which would involve a crossfader over the highpass.

It is very important that there are no unwanted inherent delays in the two signal paths caused by the order in which modules are calculated by the digital system. On many modular softsynth software packages that run on computers this might pose problems, due to the fact that these packages tend to have modules process on whole blocks of samples before output is passed on to other modules. If so, feedforward might not be sample accurate anymore and the module's input to output propagation delay will cause unpredictable results. However, modules on DSP-based systems like the G2 are calculated one sample at a time, making feedforward sample accurate and quite easy to work with. Additionally, the G2 system employs a very sophisticated algorithm to ensure that the calculation order of the modules is automatically set in the proper order. This algorithm ensures that the output value of an earlier module in a signal path is already calculated and available for the following modules in the same path. Such an algorithm is not simple, as signals can branch into several directions and all cabling connections need to be analysed to reorder the calculation order when a new module is inserted into a chain of modules or new cable connections are made. This means that insertion of a module or reconnection of a cable briefly silences the patch when a new order of calculation is necessary and the modules will be reordered. This only happens at patchtime, when a patch is set up, or on the moment when a patch is reloaded from patch memory.

Convolution

Convolution is an advanced application of feedforward. Basically a signal is split into a multitude of feedforward paths that each have a different delay. These delays range from one sample, two samples, three samples, and on and on, until an n-th sample. Then, each delayed path is fed into a mixer with n inputs channels. In essence convolution uses a block of consequetive memory locations as a multi-tap delay line, and each memory location in the delay line is connected to one input of a multi-input mixer. By setting all the mixer faders to specific individual settings, all sorts of advanced effects can be created. Examples are filtering by an arbitrary filter curve, realistic reverberation by the superimposition of the reflection characteristics of a certain room on a sound, the superimposition of the timbre of a certain sound over another sound, pitch correction, etc. Regrettably one needs a really huge amount of mixing channels, and the fader setting of each individual channel is very critical to get a certain final result. Note that when a convolution is made of the last second of sound all samples from this last second will be used, meaning that convolution easily takes up much of the computational resources. E.g., when the sample rate is 96kHz and a mono reverberation of only half a second is the aim, 48000 feedforward taps on a delayline to 48000 mixer inputs plus their specific fader settings are necessary. The only way to work practically with this technique is to have a computer calculate the fader settings and do all the feedforward and mixing in software. Note that at a sample rate of 96kHz 48000 calculations have to be made for only one single sample for a 0.5 second reverb effect and a total of 96000 times 48000 calculations would be necessary. This means that really powerful computers are needed to do such operations in real time. But if such a powerful system is available the effect will be very convincing. In practice these sort of effects are mostly rendered off-line on an audio file or an already recorded track.

When using convolution for filtering the situation is less severe, as only the amount of samples that just fit in one waveform cycle of the lowest frequency to be filtered are needed. If the lowest frequency to be affected by the filtering is 100Hz only 96kHz/100Hz = 960 calculations are needed. Within the bandwidth between this 100Hz and the upper frequency limit of the ear, any filtering curve can be made with such a filter. Still, there are 960 fader setting parameters to make, to define the filtering curve. Again a computer program will be needed to calculate these settings. Convolution is a very powerful technique, but as it needs an enormous amount of control data it is hard to use in realtime with dynamic control. It goes way beyond the practical purpose of this book to go deep into all the possible convolution techniques and why convolution works at all. Still, convolution is mentioned to show how important and powerful feedforward as a technique actually can be.

Convolution on control signals

The amount of control data and calculations involved in convolution depends on the time duration of the convolution plus the sample rate. For audio it can be quite a large number. Still, it is always interesting to see if there are processes that are run at a much slower rate. Sequencing notes is a good example of a process that runs at quite a low rate. Convolution can be conveniently used to modify the static pattern of an analog style step sequencer without destroying the programmed pattern. Only a few delayed steps are necessary to be able to slowly variate the pattern in a controlled manner with four to five knobs. In most cases a step sequencer will have one single step output for the current step, and an extra multiple tap delayline in the form of a shift register is necessary to give access to the other steps in the proper time-delayed order. When the convolved pattern must play notes in a certain scale an extra note quantizer module must be used to force the convolved pattern output back on the wanted note or chord scale. On an analog modular synthesizer the shift register would have to be patched from a couple of sample and hold modules put in series and clocked by the same clock as the step sequencer. The G2 system has an eight output shift register present as a dedicated module, which can be clocked by the step sequencer clock signal.

There are two ways to go about using the shift register to convolve a pattern from the step sequencer, an asymmetrical or a symmetrical convolution. The asymmetrical convolution simply has a multi-input mixer connected to all the taps of the delay line. When only one mixer channel is fully opened the pattern will come out unchanged, but with a delay of some steps that depends on which tap the mixer channel is connected to. When more mixer channels are opened the pattern will be filtered, resulting in smaller steps between the adjacent notes in the original pattern. To still be able to have relatively large steps the mixers channels should be able to also invert their input values. To do so it is often more handy to use bipolar gain controllers before each channel input and control these gain controllers with a bipolar control value. The pattern is varied interactively by playing with the knobs that set the bipolar control values.

The symmetrical convolution uses an odd number of taps. The taps are located symmetrically around a center tap. Sets of two taps are first mixed together before they go to the final mixer. When seven taps are used it is tap number 1 plus 7, 2 plus 6 and 3 plus 5 that are combined, while the center tap number 4 gets its own mixer channel. The middle tap is connected to the first input of the final mixer, tap 3-5 to the second, tap 2-6 to the third and finally tap1-7 to the fourth mixer input. The way to get the original pattern is by fully opening the first mixer channel that is connected to the first input of the mixer. This does however delay the sequence by four steps. Symmetrical convolution creates different sorts of pattern variations compared to asymmetrical convolution, asymmetrical convolution appears more like a canon or echo effect.

Instead of using a step sequencer to produce the pattern to convolve a low frequency generator can be used. By just playing with the rate control different patterns can be produced, the convolution will transform the low frequency waveform into the pattern. Even a squarewave will produce stepped patterns, as the pattern is basically created by a weighted sum of delayed squarewaves with different levels.

When two mixers are connected to the shift register outputs it is possible to create two simultanious but differing patterns. These patterns can e.g. interactively be brought closer to each other or made to differ more. Experimenting is of course necessary to find out how this technique can be used best in a certain musical context.

The convolving filter from the previous example can be used on audio signals, but its effect will be limited to the very high frequency range. Luckily there is a practical application as a part of an equalizer. Applying subtle EQ in the very high with the purpose to shape the sound of cymbals, hihats and the sybilants in voices is difficult, if at all possible, with standard EQ's. Mostly there is only a high shelving EQ control available, but this only set level and does not do much tonal shaping. To use a convolving filter to shape the high only a short delay with a few taps is needed. The convolving filter that was use to change sequencer patterns can be run at 48kHz by supplying it with a clock signal at that rate. On the G2 it is easy to patch such a clock, only one logic inverter module is needed. By connecting the output to the input of an inverter module it will change in a pulse generator that runs at exactly half the sample rate. Standard sample rate for this module is 24kHz, but by connecting the input of the other inverter on the module to any red output the module will change to orange and run at a sample rate of 96kHz. The pulse clock will now increase to half the 96kHz sample rate and so produce a clock at 48kHz. This is perfect to drive the shift register module at audio rate. Sonically the best results are produced when the filter is used in symmetric mode and so an odd number of taps are needed. One extra tap can be created by adding another sample and hold module in front of the shift register and clocking it with the same 48kHz signal. There are now nine taps that will form a filter with five parameters to set, which is an easy number of knobs to handle interactively. The bandwidth is 48kHz divided by the number of taps, so the filtering action will mainly be between 5kHz and 20kHz. The effect on this band wll be quite dramatic and the tonal shape of e.g. cymbal and snare sounds can be precisely controlled. The knob on the middle tap will pass the clean high signal, of only this knob is fully opened there is no filtering action. By slowly tweaking the other knobs the very high region of the sound can now be EQ'd. By using a crossover filter set to 5kHz before the convolving filter and feeding the low band clean to the output the convolving filter will only work where it should. It is important to give the low band a slight delay equal to the delay of the middle tap of the convolving filter. This can be done with the clocked delay modules, that should be clocked with 48kHz as well and be set to 5 taps. The audio will now pass at a 48kHz sample rate, but as there are no dynamic modulations involved there will be no apparent difference in sound to a 96kHz sample rate, and note that 48kHz is still the professional DAT sample rate.

Feedback

Feedback is when a signal is split into two paths and one path is fed back and mixed with the original signal on an insertion point before the split is made. This insertion point is made by inserting a two input mixer at the point in the signal chain where the feedback will have to be applied.

Integration and lowpass filtering

The most simple example of feedback is when the output of a mixer module is fed back to one of its inputs. On an analog system this would probably immediately cause a race state and quickly clamp the mixer output at the positive or negative power supply voltage. But on a digital system something else and actually very useful happens, as in a digital system there will be a delay of at least one sample in a feedback loop. The explanation of the effect caused by this very short time delay is a little technical, so prepare yourself for the next paragraph. First thing to note is that when modules in a digital system are calculated, it is common to store the module output values in memory locations named output registers. Other modules can read these memory locations later and use the found values as input values. Second thing to note is that digital mixer modules make use of only one single DSP command for each mixer input, which will do a multiplication of the channel input value with the mixer input attenuation value set by the mixer knob for that input, and automatically add the result to a temporary output accumulation register. The effect is that the addition of one scaled mixer input value to the final output value is done by only one combined multiply/accumulate instruction and e.g. a three input mixer will just execute three of these instructions in a row to produce the final output value in the temporary accumulation register. Only after these three instructions are executed is the final output value stored from the temporary accumulation register to the mixer module output memory location where it can be used by other modules. This means that when there is a feedback of the output to one of the inputs of a mixer module it will always use the mixer module output value of the previous sample, simply as the intermediate output value is still in the temporary register and not yet stored in the final output register. This will cause the one sample delay in the feedback loop. The sample that is currently calculated is commonly named the Z sample, and the previous sample is named the Z-1 sample. So, the feedback on the mixer input will use the Z-1 sample. The effect of this very small time delay is that the feedback will now cause an effect named integration. In essence integration is an averaging effect, as the Z sample will contain an average of a whole series of previous samples. Lets take as an example a feedback of 50%. This will cause the Z-1 sample to have an effect of 50%, the Z-2 sample an effect of 50% of 50%, the Z-3 sample an effect of 50% of 50% of 50%, etc. Theoretically all previous samples up to the sample that is an infinite time ago would have some effect, the effect decreasing proportional to the age of the sample. In practice the mathematical precision or resolution of the digital system will put some limit to this time. The sonic effect of averaging caused by integration is a shallow lowpass filtering. The reason why this results in lowpass filtering is because the cycle of a high frequency is much shorter as the cycle of a very low frequency. And as the effect of the averaging is much stronger on recent samples as on samples that passed a long time ago, the effect on high frequency cycles is simply greater as on low frequency cycles. So, not only does the actual values get averaged out, frequency partials also get averaged out, and high frequency partials much more as low frequency partials. Which is exactly what a lowpass filter does. The more the feedback approaches unity gain, the lower the cutoff frequency of the filtering action will be as the averaging is active over a much longer time. This all simply means that feedback of a mixer output to one of its inputs will cause a high frequency damping in the mixer, and this can be put to good use. But as the feedback will also create a build up of energy in the feedback loop, the other signal input must be attenuated. This attenuation is necessary as the feedback loop increases the overall gain for all the channels on the mixer. And too much gain will cause the output signal to hit the headroom of the digital system, resulting in clipping. Luckily it is quite easy to figure out what the attenuation on the other inputs must be, as the relation is linear. E.g. when the feedback is 50% on a two input mixer, the other input should be attenuated by 50% ,and for 75% feedback the other input should be attenuated to 25%. So, the amount of feedback plus the attenuation on the other input should add up to 100% to pass the other input with unity gain to the output. In practice the gain for the other input will in fact never be fully unity gain, as the mixer now also act as a lowpass filter. This causes the gain for each partial that is present on the other input to differ from the gain for other partials, the higher the frequency of the partial the lower the gain. Very low frequency partials will however be passed with almost unity gain.

When a two input mixer is set up with a feedback loop as described, it turns into what is named an integrator. To work properly an integrator must work with that Z-1 sample, if the feedback delay is more than that one sample it doesn’t work reliably anymore.

If an integrator is placed within another feedback loop it will cause a very useful high frequency damping in this other loop and can additionally prevent the feedback loop to exceed unity gain. This will make another feedback loop much more stable and reliable. The high frequency damping will additionally allow for a more natural sound, as in a way it mimics the high frequency damping when a soundwave travels through the air. Such a damping is caused by atmospheric circumstances, like e.g. the moist in the air, which forms a resistance that is greater for higher frequency partials in the soundwave. Additionally, the human mind focuses more easily on high frequency partials, and some high frequency damping will shift the attention to the frequencies in the midrange, which can help in creating a more balanced mix where the mind's attention is guided to where the melody or the articulation is happening.

The actual amount of high frequency damping caused by the integrating mixer is depending on the sample rate, when the sample rate is 96kHz like on the G2, a pleasant amount of damping is created with a feedback of around 75% and an attenuation of the other input of about 25%. These values can be a good starting point to find the balance that works best in a certain application. As a rule of thumb a feedback loop by default needs damping. And consider the few cases where damping can be omitted only as exceptions to this rule.

Stability issues

Before looking at more applications for feedback it is important to note that there are two unwanted effects that can appear when applying feedback. The first is a possible severe overload and/or clipping, the second is a possible high frequency oscillation. If one is aware of these possible effects and takes proper precautions it is easier to patch stable feedback systems that work just like expected.

Preventing overload in a feedback loop

Feedback is much more critical than feedforward, as when the feedback signal is not properly attenuated it might build up to the headroom of the system, and finally try to exceed the headroom limits. If a small DC component is present in the input signal to a feedback loop the DC component will build up in the loop and shift the output signal substantially towards one of the headroom limits. This can cause clipping to occur much earlier than expected. The way to solve this problem is to insert a highpass filter set to a very low frequency into the feedback loop. The highpass filter will block any possible DC component as a DC component has a frequency of 0 Hz and a highpass filter will always infinitely attenuate a frequency of 0 Hz. In addition to the DC blocking action the highpass filter can also act as a low frequency cut to prevent a muddy sound in the bass octaves. The cuoff frequency can be set between 40 Hz to 120 Hz, depending on the amount of low cut that is wanted.

On an analog system and when there is no time delay in a feedback path, the energy in the loop will build up immediately towards the headroom limits of the system if feedback is over unity gain. Such a superfast build up is named a race state and can occur quickly on analog systems. Here, a feedback gain that is just slightly above unity gain can almost instantly cause the energy that goes around in the feedback loop to explode. Many times an explosion of energy in an analog feedback loop will simply clamp the output signal permanently to either the negative or positive supply voltage of the electronics and keep the output there, resulting in actual silence. Feedback on an analog system is much more tricky as on a digital system, as it is quite difficult to set a high feedback level without accidentally cause the feedback to exceed unity gain. Some analog components exhibit a saturation effect, which in practice means that they act as a signal limiter to keep the feedback level in check. Examples are the cheaper VCA circuits, vacuum tubes or e.g. magnetic recording tape. These components can prevent clamping in the feedback loop, but will generate a lot of harmonic distortion. In some styles of electronic music however this type of distortion is highly valued for its grungy character. Digital systems lack an inherent saturation effect and is acually tricky to implement, one of the reasons why digital systems can react different to analog systems when using patches that employ feedback.

In general, to prevent overload and/or clipping it is important to keep the feedback gain just below unity gain. But even then clipping can occur, in which case the input signal must be attenuated to a much lower level, sometimes some 10 dB to 20 dB below a normal input level.

When there is a relatively long time delay in the feedback loop, like with a long echo delay, there might be enough time to attenuate the feedback signal manually before the loop explodes and overloads. It is also possible to use a compressor or AGC (automatic gain controller) circuit in an echo feedback loop. Compressors are not ideal as they can easily introduce a pumping effect in an echo feedback. An important thing to keep in mind is that there is always a slight time delay before the compression takes effect. This is true for both analog and digital systems. The effect of the time delay is that fast attacks in e.g. percussive sounds are hardly affected by the compression. Varying the signal level that is fed into the compressor will change the sound and snappiness of the attacks. This is a great sound manipulation to tweak the percussive timbres, but it is in general unwanted on a compressor that is applied in an echo feedback loop. An AGC circuit is in general much slower as a compressor, it might take a few seconds to settle on the wanted signal level. In practice this makes AGC circuits a bit more useful in echo feedback loops compared to compressors. Another method to limit the gain in the feedback loop is to use a tape saturation emulation circuit. Such a circuit works almost immediately, but will also cause a lot of odd harmonic distortion. This type of distortion is typical for magnetic tape, and so this method is very useful to recreate the effect of a vintage tape echo device. Another possibility is to use an analog VCA circuit in an external echo feedback loop of a digital delay unit and tweak the VCA in a way that there is just the right balance between the limiting effect and the odd harmonic distortion. For this application it can be expected that cheap quality VCA modules work out better as expensive ones.

Preventing high frequency oscillation in a feedback loop

If in a digital system the feedback signal is over unity gain or is treated by a non-linear function, the feedback can cause severe oscillation at half the sample rate. It is the short time delay of at least one sample between input and output of the circuit that can cause this unwanted high frequency oscillation. Note that the feedback signal at the input must always use that previous Z-1 output sample, and so a one sample delay is inherent. Because the feedback loop in a digital system is inherently also a lowpass filtering function because of this Z-1 delay in the feedback loop, the inherent lowpass cutoff frequency is increased when the amount of feedback is increased. And at high feedback levels the lowpass filtering also becomes more resonant. So, if the feedback level approaches unity gain the resonance is very high at a resonant frequency at about half the sample rate. So, even if the feedback is below unity gain the loop can exhibit a ringing effect when resonance becomes high, but will not yet go into self-oscillation. The obvious way to solve possible ringing or self-oscillation at half the sample rate is by using damping. There are two ways to implement damping. The first is to simply attenuate the feedback signal to well below unity gain, but this is probably not what is wanted. The second way is to create a frequency dependent damping, notably high frequency damping or bandwidth limiting. It is a good idea to look at a feedback loop as a model of a self-controlling system that is in a balanced state. Input will try to get the system out of balance by causing it to ring or go into self-oscillation on a natural resonance frequency inherently present in the feedback system. The purpose of damping the feedback loop is to suppress the natural resonance frequency of the system, making it easier to maintain the balance in the model. This means that the gain of the feedback loop must be made frequency dependent, but without causing extra phase shift that could create a new resonant frequency. As a rule of thumb a 6dB lowpass filter in the feedback loop, set to 5% of the sample rate or the natural bandwidth of an active analog component like an operational amplifier, will make the feedback loop stable enough for most purposes. On a digital system with a 96kHz sample rate the lowpass filter can be set to 5kHz, which also helps in giving the feedback signal a nice, warm sound. But the main purpose is to make the loop stable, it is just a nice bonus if it sounds warm.

Anyway, when applying feedback the main concern is to have absolute control over the amount of feedback. If this control is properly taken care of, which in practice means that for any frequency partial in the feedback signal the loop gain will never exceed unity gain and feedback decreases slightly for higher frequencies, feedback will help to create a number of musicallly very useful effects, like the resonance in filters, deep phasing effects, tube-like harmonic distortion, naturally decaying echoes, early reflections in room simulations, reverberation, etc.

An important thing to keep in mind is that there is always a slight time delay before the feedback takes effect. This is true for both analog and digital systems. It is this short time delay that can cause unwanted high frequency ringing or oscillation even if the feedback is below unity gain, basically because any feedback loop will always have a natural resonance.

Selfmodulation

Selfmodulation is when the output of a module is connected to a modulation input on the same module. This is in essence also a feedback loop. But the effect of this feedback does not have to be linear, like it is with the integrating mixer. Depending on the type of module, this feedback can in fact be highly nonlinear. An example is when the output of a sinewave oscillator is fed back to its linear frequency or phase modulation input. Such a feedback connection is potentially chaotic, if the feedback level exceeds a certain level the output waveform will change into a chaotic signal. Such a signal is very close to noise, but it produces a very complex waveform that actually repeats. Basically this complex waveform denotes a state of balance. If the feedback amount is slightly changed the waveform will apparently change at random for a while until it gets stuck into another state of balance, where it will produce another complex but repeating waveform at a different frequency. In addition to the purely chaotic behaviour there is an additional tendency to resonate at half the sample rate if there is no damping applied in the feedback loop. Both effects actually influence each other and this can cause the resulting noisy signal to sound quite harsh and unpleasant. What happens in this example is that when a feedback loop is applied on this phase-modulation oscillator, no matter how much signal is fed back, the oscillator will always output a signal that is never higher in amplitude as the original sinewave it generates without modulation. So, the gain in the loop can be well over unity gain, as the nonlinear sine function in the oscillator always 'folds' the modulation signal back into a normal output level range, no matter the depth of the selfmodulation. Meaning that the output signal level can never explode. Instead of exploding it will start to create chaotic behaviour. This chaotic behaviour is deterministic, it is not purely random. It follows all the rules of what is known as the Chaos Theory, that tries to describe chaotic behaviour in natural processes. Chaotic behaviour normally develops through what is named waveform period doublings or bifurcations. These bifurcations can create musically quite useful subharmonics. But the tendency to resonate at half the sample rate will destroy the predictability of the occurance of these period doublings. Inserting an integrating mixer in the feedback path will dramatically increase the stability and predictability of the development of the chaotic process, as the integrating mixer will suppress the tendency to resonate at half the sample rate. The high frequency damping effect of the integrating mixer will in most cases be much less significant to the final result compared to the tendency to resonate. How to create sounds with subharmonics based on bifurcations will be described somewhere else in this book.

Summary

Feedforward and feedback are very important techniques that are easy to patch yourself on a modular system. They allow you to build all sorts of the more advanced sound processing techniques and can give much more control over the final sound. Feedforward can be used to apply an effect to only a part of the sound spectrum by first creating a crossover filtering effect. Another use is to create lively timbral effects caused by interference between the two or more parallel signal paths. Convolution is the most advanced type of feedforward, but on realtime systems it is at present limited to just a few simple applications. Feedback is used on a multitude of techniques, ranging from creating very soft or strongly resonant filters to echo delays and selfmodulation on oscillators. Feedback is also used in physical modelling where it is used to let energy recirculate in waveguides, which are short audio delay memories with the length of exactly one cycle of a waveform at the played pitch. These physical modelling techniques will be explained in a later chapter.

Filters

Introduction

Filters can selectively remove or emphasise certain areas in the frequency spectrum. Areas which are removed are named stopbands and areas which are passed are aptly named passbands. If a single frequency is strongly emphasized this is named a resonance, a filter that allows for this resonance is named a resonating filter. On the frontpanel of some synthesizers resonance is also referred to as emphasis or Q. Resonance can be wanted feature, especially when a timbre needs to be dramatically shaped. In other applications it can be an unwanted feature, as an example a strong resonance would not be accepted if it would occur in the bass and treble controls on a hifi amplifier. It is part of the design of a filter if the filter is allowed to resonate on a certain frequency. Some filters can resonate up to the point where they start to oscillate and other filters have no resonance at all. By dynamically changing the resonant frequency extra expressive dynamics in the timbre can be created. Many different types of filters have been developed over the years and all filters that work in the audio range are good candidates to spice up your music.

When a raw oscillator waveform is filtered in a filter module, the filtering will add an extra dimension to the basic timbre of the raw waveform. A common use is to create subtle and natural sounding decays by softly sweeping a filter at low resonance on a static waveform or a sample based sound, using an envelope control signal to control the sweep. Other common uses are to create formant areas in a sound or damp the higher notes a little to make the sound appear more natural or ‘acoustic’. Filters also allow for increased dynamic and expressive playing styles by dynamically tweaking a strongly resonant filter during play. In practice filters are equally important as oscillators in sculpting the timbre of a synthesized sound. Filters are not only used to dramatically alter the timbre of oscillators, but they can be used equally well on virtually any external sound source, like a microphone, a recorded track or looping sample, a drumcomputer, another synth, etc.

In the most traditional approach of subtractive synthesis the oscillator is responsible for the pitch by supplying a pitched waveform which contains all possible harmonics, like the sawtooth waveform. Then, a filter is used to create the formants which are important to create the timbral character of the sound. But if a more complex modulated waveform is fed to the filter the filtering can also be used to emphasize characteristics already present in the waveform. In this case the sound source and the filtering are equally important in shaping the timbre, the oscillator and filter work as a unity to get the desired sonic results. With this approach the range of timbres becomes much greater than with the traditional approach. Remember that what a filter does depends a lot on what is fed into the filter.

Filter classes

There are two common approaches in classifying filters. The first approach is the discrimination between static and dynamic filters. In a static filter the frequency bands are fixed and only the amplitude of each band can be controlled. A good example is the graphic equalizer. This filter basically splits the audio range into a number of bands and for each band the amplitude can be set by a dedicated knob or slider. Another example of a static filter is the bass and treble controls on an amplifier. Dynamic filters additionally offer the possibility to control the frequency range of the band, allowing it to become wider or narrower by knobs and control signals or a digital controlling parameter, e.g. a control signal received through MIDI from a sequencer program.

The second approach is from a totally different angle and much more technical. This approach discriminates between Finite Impulse Response or FIR filters and Infinite Impulse Response or IIR filters. FIR filters offer the possibility of any arbitrary filtering function and are in their practical use quite similar to e.g. graphic equalizers. FIR filters offer much more resolution in defining the filtering curve when compared to IIR filters. The disadvantage of FIR filters is that it is very hard to control them dynamically, a computing device is needed to process a vast amount of controlling parameters into a control array, which can easily hold several thousand filter parameters. This makes FIR filters impractical to dynamically and expressively control a timbre while playing live. In contrast IIR filters are much easier and intuitive to control as they have only very few controlling parameters. So it is this type of filter that is commonly found on sound synthesizers. Resonating filters, where both the resonant frequency and the amount of resonance can be controlled, are in general of the IIR type. The difference in the technical implementation of these two types is that FIR filters are based on a feedforward technique named convolution and IIR filters are based on a feedback technique named recursion. In the chapter on feedforward and feedback you will find more information on these techniques. All references to filters in the rest of this chapter will be to IIR filters.

Passband characteristics

A filter can at the same time suppress certain frequencies, leave other frequencies basically unaltered and optionally emphasize certain frequencies. This behaviour can be drawn in a graph of the audio spectrum where the curved line indicates the amplitude responses for each possible frequency in the audio range. Such a graph is called the transfer function of the filter and defines the passband characteristics of the filter. There are some typical basic transfer functions for simple filters. If the transfer function reveals that all lower frequencies up to a certain frequency are transferred with virtually unaltered amplitude, but the frequencies above this certain frequency are passed with much lower amplitude, the filter is named a lowpass filter. In practice it will simply pass on all lower frequencies and suppress the higher frequencies. If in contrast the lower frequencies are suppressed and the higher frequencies are passed with unaltered amplitude the filter is named a highpass filter. So, the highpass filter can be seen as the inverse of the lowpass filter. If only a frequency band somewhere in the middle of the audio range is passed unaltered, but both lower and higher frequencies are suppressed, the filter is named a bandpass or bandfilter. A filter can also suppress a band somewhere in the middle of the audio range, making the filter a bandreject or notch filter. A notch is the point in the audio spectrum where a frequency is totally suppressed. The opposite of a notch filter is when all frequencies are passed through, but only one small frequency band is strongly emphasized. Such a filter is named a peak filter or sometimes a resonator as it introduces a strong resonance effect on a certain frequency while the rest of the sound is left unaltered. A similar fiter exists that will pass all frequencies but creates a series of peaks at harmonic intervals, such a filter is named a combfilter. There is also a filter type that does not change the amplitude of any frequency at all but in fact gives each frequency a different phase shift. Such a filter is named an allpass filter.

So, in essence there are seven basic transfer functions, lowpass or LP, highpass or HP, bandpass or BP, bandreject or notch or BR, peak or PK, comb, and allpass or AP. By using several filters with different transfer functions in series and/or in parallel, a combined transfer function can be made that can be much more complex. Such a complex filter can be put to good use to simulate the very complex and expressive timbral changes like those found in spoken words.

Playing the filter

Many times the purpose of using filters is to get expressive control over the timbre with the least amount of controls to play with. If the intent is to play live there is probably only a single play controller available like the modwheel, keyboard aftertouch or a foot pedal to control the timbre. Using such a controller puts a limit to the possible complexity of the filter, as only one parameter of the filter can be controlled. An X-Y controller like a joystick allows for some more complexity, as it can control two parameters in a single movement. With a joystick it is for instance possible to control the resonance frequency with on one axis and the resonance amount on the other axis. Or each axis can be used to control a resonance frequency on a complex filter that can resonate on two frequencies. When the synthesizer is controlled from a MIDI sequencer or computer program there is hardly any limit to the complexity of the filtering function, as all parameters may be sent by the sequencer over MIDI, and changing these parameters over time can on most sequencers be edited graphically before the sequence is played.

Learning to play the timbre with the use of filters takes time, just like learning to play any instrument. There is no magical formula that will make a filter instantly sound good. Instead one needs to develop a feel for tweaking filters, to master both the dramatic and the very subtle changes in timbre that filters can produce.

Filter parameters

Cutoff

Most prepatched analog synthesizers are equipped with a single lowpass filter per voice. Such a filter will have a control knob for the frequency from where the suppressing of the high frequencies will start. This point is named the cutoff frequency. A lowpass filter on a modular system will also have one or more inputs to enter a control voltage or a control value to initially set or modulate the cutoff frequency. This input can be controlled by e.g. the output of an envelope generator module. This envelope will define how the timbre changes while the note sounds, creating a faster or slower swell and a faster or slower decay. The cutoff frequency control is also a very good candidate to be played by the modwheel or by the keyboard velocity value. Adding just a bit of keyboard velocity control signal to the cutoff will make the sound slightly brighter when hitting the key harder, which can give a more natural effect to the amplitude dynamics of the sound. Another common application is to add some signal from a very low frequency triangle waveform oscillator to the cutoff control signal. This makes the timbre slowly evolve, commonly used to give a 'sense of going somewhere' to a repetitive sequence of notes.

The cutoff frequency can also be controlled by a signal in the audio range, a technique that can create exiting new timbres. An example is to do this modulation with a waveform at half the pitch of the played note. This will create a subtle 'subharmonic effect' to the sound, giving the timbre a bit more 'beef'. It is common that this 'suboctave' modulating waveform has a tight relation with the oscillator signal that is filtered. However, by having a slightly detuned relation interesting beatings in the sound might occur that can work out very well on long droning sounds. In fact there are many more modulation possibilities for the cutoff frequency and it is great fun to think of new ones, to try them out and see how it can spice up your music.

Keyboard tracking

For many keyboard sounds the cutoff frequency of a lowpass filter needs to be raised if higher notes are played on the keyboard. By default filters are patched in a way that will make this happen by automatically adding some of the keyboard voltage value to the filter cutoff parameter. This is named filter keyboard tracking. It is quite important to have some control on this keyboard tracking. If the tracking is off, meaning that the filter cutoff is set to a fixed frequency regardless of the played note, the highest notes will be heavily suppressed by the filter and can hardly be heard anymore. But if the cutoff frequency tracks the keyboard fully the timbre will in many cases become unnaturally bright for the highest notes. In practice the keyboard tracking needs to be adjusted somewhere between no tracking and full tracking to get the most natural brightness changes when playing notes all over the keyboard. Adjustment is best done by ear to get the most natural feel. Tracking is mostly expressed in a percentage where 0% means no tracking , 100% full tracking and 200% means strongly exaggerated overtracking. A good way to set the tracking is to set an initial tracking percentage of about 30% to 50%, then play a note in the middle of the keyboard and tweak the cutoff frequency for the right sounding timbre for that note. Then play the highest and lowest notes from the melody to be played and adjust the tracking and possibly the cutoff frequency on that middle note again to get the filter to sound just right over the whole keyboard range.

Resonance, Q and emphasis

The type of lowpass filter that is commonly found in sound synthesizers offers the possibility to strongly emphasize the frequencies on and just around the cutoff frequency. The knob to control the strength of this effect is commonly referred to as resonance, Q, quality or emphasis, depending on the brand of synthesizer. Opening this knob causes the effect of a very strong resonance at the cutoff frequency and in practice this creates a strong 'whistling' formant in the sound. Earlier it was decribed that formants are narrow frequency bands in natural sounds where the frequencies have a much higher amplitude than in other parts of the audio range. Particularly the formants in the range between 500 Hz and 4kHz will strongly characterize a sound. In the real world formants are created by complex resonances in an acoustic musical instrument or in the vocal tract in human speech. The pure sawtooth wave, square wave, triangle wave and sine wave have no formants as in the harmonic spectrum plot of these waveforms all harmonics become gradually weaker when their harmonic number becomes higher. All waveshaping manipulations on these standard waveforms will alter their harmonic content and thus create a weaker or stronger structure of formants, depending on the type of waveshaping used. This formant stucture can be further enhanced by the filter resonance, meaning that here the resonance control on the filter is mainly used to increase the character of the timbre. In general only a moderate amount of resonance will do that job. Of course character is very subjective and a matter of personal taste, so adjustments are best done by ear. As the resonance frequency is the same as the cutoff frequency it is again very important to tweak the keyboard tracking to a percentage that best suits the sound.

How filter resonance comes about

The resonance needs to be activated by an impulse in the waveform, it needs to be excited. The abbreviation IIR actually refers to how the filter responds to an impulse, 'infinite' in IIR means that the resonance of an IIR filter can be set to a point where it responds infinitely to an impulse. This point is when the resonance knob is about fully open and the filter starts to oscillate and produce a sinewave all by itself. This is named filter selfoscillation. So, by gradually opening the resonance knob the filter starts to respond to impulses by producing increasing amounts of resonance at the cutoff frequency, until the filter starts to oscillate. Before selfoscillation happens the impulses that cause the resonance need to be present in the input waveform. If this waveform has steep edges, like the sawtooth wave and the square and pulse waves, the edges will act as the impulse and a strong resonance effect will result. But waveforms that are very smooth or do not have steep edges, like the sine and triangle waveforms, do not cause much resonance effect at all, even if the resonance knob is set to almost oscillation. Only if there is a strong partial present in the sound at exactly the resonating frequency the resonance will be heard. If the resonant frequency is set to the fundamental frequency of a sound the fundamental will be strongly boosted by the resonance. This is put to good use in bass sounds to create superboom basses. The idea is here to use a highpass filter and set the cutoff frequency on exactly the fundamental frequency. The highpass filter will now pass the signal unaltered, as all that is in the sound to be filtered is in the passband of the highpass filter. When the resonance control is opened the resonance on the fundamental will start to strongly boost the fundamental. The interesting notion about this example is that a highpass filter is used to give more beef to the low end of a sound. So, a highpass filter is not only capable of making a sound thinner, but also to make it more 'phat'. In the chapter on sound sources it was explained that the flanks in waveforms are named transients and are in fact points in time where the oscillator releases a huge amount of energy. Remember the little experiment descibed at page xx, where a low frequency sawtooth wave was fed directly into a speaker. Let's do this experiment again, though now by connecting a squarewave oscillator directly to the amplifier. When the frequency of the oscillator is drastically lowered to around two and a half Hz one can clearly hear the rhythmic clicks when the cone is moving in and moving out, and notice the silence between the clicks. Now insert a bandpass filter set to a fairly high resonance setting and a cutoff frequency around 2000 Hz between the squarewave oscillator and the amplifier. Instead of each click a very short burst of a 2000 Hz sinewave can be heard, sounding a bit like claves. Tweaking the resonance knob will make the burst shorter or longer in time. Apply a little bit of the low frequency squarewave to the cutoff modulation input to create an alternating higher and lower pitched clave sound. If you are lucky enough that the low frequency squarewave oscillator has a pulsewidth control you can also change the pulsewidth of the square just a little, to give some swing to the two blips. Lowering the cutoff frequency knob to about 70 Hz will change the sound into what starts to sound like a bassdrum. This is an example of how a simple swinging beat can be patched with only a low frequency oscillator and a resonant filter. But the important thing this experiment reveals is that it is really the flanks in the waveform that excite the filter, and that the resonance is in fact a small burst of a sinewave starting at and caused by the transient. If you now try a triangle waveform instead of the square waveform you will hear how important the flanks really are, as the triangle wave will appear to do nothing at all to the filter. The amount of energy present in the transient directly influences the amplitude of the sine burst. The amount of energy in a transient is determined by the vertical height of the edge in combination with the angle of the transient, the higher and steeper the angle, the more energy it can release in the filter. This explains why the filter resonance does not respond well to a triangle wave, but very well on sawtooth, square and pulse waves. The slope of the triangle wave is simply not steep enough to excite the resonance to make it significant. On the Minimoog synthesizer there is a waveform present that strongly resembles a triangle wave but at the upper and lower peaks of the wave there is a small transient present. Such a waveform can be created by adding about a fifth of a sawtooth wave or square wave to a triangle wave with the aid of a mixer module. The purpose of this waveform is to be able to have sounds based on the typical mellow sound of the triangle wave but that still allow for just a bit of resonance effect to add a bit more resonance character in the filtered wave.

Steepness of filters

Steepness is the amount of frequency rolloff in the suppressed frequency band. A lowpass filter with its resonance knob at its lowest value will hardly suppress frequencies around and below the cutoff frequency. When the pitch of the input signal is gradually raised above the cutoff frequency, the input signal gets suppressed more and more. If the amplitude decreases very quickly when the pitch is raised by an octave interval higher as the pitch, the filter is said to have a steeper cutoff slope. This can be tested very well with a sine wave. If the filter cutoff frequency is set to 500 Hz the sinewave will be passed without significant drop in amplitude below 500 Hz. But from 500 Hz up it will gradually decrease in amplitude. The steepness of a filter is expressed in dB per octave or simply dB. dB stands for deciBell and is a relative value. This means that first a reference must be defined and this reference is interpreted as 0 dB. 0 dB is referred to as unity gain when the amplitude of the imput signal is the same as the amplitude of the output signal. With a filter the output amplitude of the sinewave at the cutoff frequency is taken as the 0 dB reference value. Now raise the frequency of the sinewave by an octave. If the amplitude has dropped to half the amplitude at the cutoff point the suppression is said to be –6dB. If the amplitude has dropped to a quarter value the suppression is –12dB. And if it has dropped to an eighth the suppression is –24dB. The filters that would cause these attenuations would be said to have a steepness of respectively 6dB, 12dB or 24dB. Musically a steepness of more than 24 dB is hardly heard and so 24 dB for a lowpass filter is commonly considered as sufficiently steep for timbre shaping purposes. In the last examples a theoretically perfect filter is assumed, in practice the slope starts a little below the cutoff frquency and bends down smoothly before becoming an almost straight line downwards. But this is of little concern to a musician as in the end the sonic effect of a filter on the material that is fed into the filter must be judged by ear.

Filter building blocks and filter poles

Technically filters are made by cascading basic filter building blocks that have a fixed 6dB steepness. This is why the steepness of filters is commonly denoted by a multiple of 6dB. These building blocks are sometimes referred to as poles. The word pole is derived from the mathematics involved in designing filters and is not an entirely correct name for the building block itself. The word pole should only be interpreted as an indication on how many building blocks are used, so if a filter is referred to as a 4-pole filter, it indicates the use of four building blocks that can be combined in a number of ways. If combined as a lowpass filter it would indicate a 24 dB filter. But note that this 4-pole filter could also be a variable width bandpass filter with two 12 dB slopes, a bandfilter with a 6 dB highpass and a 18 dB lowpass slope or any other combination that can be made with four of the basic 6 dB highpass, 6 dB lowpass or allpass building blocks that are available to the synthesizer designer. This means that, on the most common synthesizer filters, the steepness is fixed by the architecture chosen by the designer of the filter. But when more filters are present there are a number of tricks to get some control over the steepness and the more interesting ones will be explained later.

Sometimes it is said that the steeper the filter the better the filter. This is not entirely true, it depends a lot on what is actually filtered. The steeper a lowpass filter the more the brightness gets suppressed. Which can be an important argument to use a less steep filter when filtering an audio track, a looping sample or the signal from a drum computer. Preferrably one would like to have several filter types available, all with different steepness and resonance characteristics. Here a modular synth has a distinct advantage over a prepatched synth. On a digital modular synthesizer there is the additional advantage that as many filters as fit in a DSP can be used to create a very complex filter with a very complex transfer function.

Combining filters into more complex filtering functions.

Filters can be cascaded in series and used in parallel to create more complex filtering functions. There are some simple cases, like cascading a lowpass filter and a highpass filter to create a variable width bandpass filter. When using filters in parallel and by mixing their outputs together with a mixer module, they start to interact in an interesting way. In a filter each frequency will be shifted in phase a little. The phaseshift for a certain frequency on the outputs of two parallel filters will be different if the filters are set to different cutoff frequencies. When the outputs of the two filters are mixed the phase shifts will cause an extra cancellation and emphasis of certain frequencies. There are filter types that specifically use this effect, these filters are commonly named elliptic or Cauer filters. These filter types can be easily created by combining two or more filters in parallel, feeding them the same input signal and adding or subtracting their outputs from each other in a mixer. If this is done with two resonant filters there is the additional bonus of two resonant peaks, effectively creating two possible formants in the filtered timbre. This can give the sounds a musically interesting ‘talkative’ character.

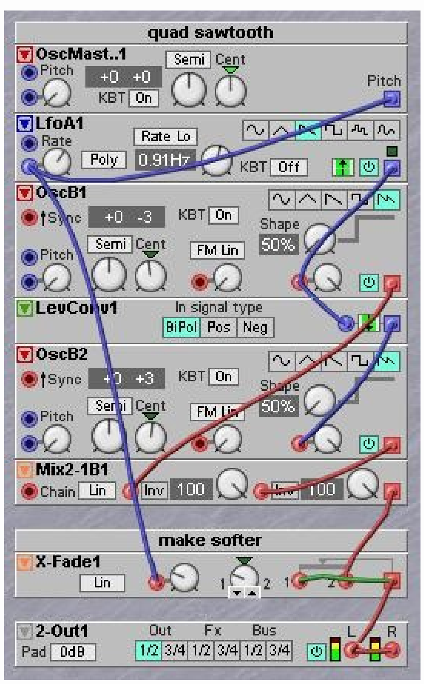
Common filter types

6 dB lowpass and highpass

These filters are more or less the basic building blocks all other filters are made with. The filtering effect is soft without significant coloration and these filters are very useful when only a shallow frequency rolloff is necessary. Resonance is not possible with these filters. An important characteristic of these filters is that their gain in the passband is slightly less than unity gain. This makes the 6dB lowpass filter ideal to be used in a feedback path in echo delay lines, selfmodulating FM oscallators and waveguide oscillators to create high frequency damping and to prevent feedback oscillation to occur. The phaseshift of these 6 dB filters varies for each partial in the input signal and is around 45 degrees at the cutoff frequency.

These types are not very useful in colouring the timbre of a sound, as their filtering action appears shallow and doesn't add any significant character of its own. Still, they are quite important as they can be used for all sorts of other useful purposes. The 6 dB highpass has the important property that it blocks DC components in audio signals, used this way it works the same as an AC-coupling capacitor in analog circuitry. This can be put to good use in digital systems, as there are synthesis methods where an inherent DC component is added to the generated waveform signal. If such a signal is applied in a feedback loop it can lead to a build up of the DC component, which will shift the audio signal towards the positive or negative clipping border and create unwanted clipping. Passing the audio signal through a 6 dB filter set to a low frequency around 40 Hz will remove any DC component and cure this unwanted clipping. There is one drawback and that is that a 6 dB HP filter does not allow for a feedback greater than one or unity gain, as this will cause a radio frequency oscillation on an analog filter or an oscillation at half the sample rate on a digital HP filter. If deep feedback is desirable, like in some types of digital FM synthesis, it is advisable to also enter a lowpass filter in the feedback loop set to a frequency of around 1/8th of the system sample rate to prevent unwanted oscillations of the feedback loop at half the sample rate. Another situation where a 6dB highpass filter can do wonders is when a kickdrum and a bassline conflict with each other and make the low in a mix sound muddy. By routing either the kick or the bass through a 6 dB highpass filter set to around 80 Hz to 150 Hz the mix can be improved. Judgement by ear is again important here. And when a microphone signal is fed directly into a synthesizer, e.g. to do vocoding or another type of speech mangling, routing the microphone signal through a 6 dB highpass set at around 150 Hz will fight rumbling sounds in the microphone signal and too much low end in a vocoded signal.

6 dB filters are not commonly found on commercial analog modular synthesizers. But they are commonly found on FX machines that have reconfigurable signal routing and on digital modular synthesizers. On DSP-based modular synthesizers a 6 dB lowpass filter can be easily make with a simple two input mixer, which was earlier explained in the chapter on feedforward and feedback at page xx. To summarize the principle; when the output of the mixer is routed back to one of its inputs there will always be a one sample delay as the feedback input will calculate with the value of the last time the output was calculated. Such a circuit is named an integrator. The one sample delay is sometimes referred to as a Z-1 function. This small delay is essential in building digital filters, it provides for a small time delay that is essential in the filtering function. In this example of a simple 'two input mixer integrator' the mix on the two inputs must be balanced, if the feedback is increased the input signal on the other input must be decreased in an equal amount, if feedback is 75% then the input signal must be attenuated to 25%. This means that a crossfade mixer in a linear mode will do this job perfectly and with only one knob. The actual cutoff can be calculated, but the formula is a bit complex. Instead it is better to tune this little 'do it yourself' filter by ear. In analog circuitry the same idea is used, but done by connecting a small capacitor over an inverting opamp. This simple soft filtering trick can be applied on the output of a sawtooth waveform oscillator to make the oscillator appear to have a warmer sound. Especially on unison sawtooth sounds this can increase the sense of depth in the sound. A nice side effect is that if a high resonance filter is used on this unison sound the resonance in the very high is dampened, which gives a much more analog sound and a more balanced filtersweep on a digital system.



6 dB allpass

The 6 dB allpass filter will pass all frequencies equally well. When an audio signal is fed into the allpass filter it appears like nothing happens. So, why on earth would one want to use an allpass filter when it doesn't filter away anything? The answer lies in the fact that the filter actually does something dramatic, but this can only be heard if the allpass filter is used in combination with other modules. What an allpass filter does is shift each partial in a sound in time. The timeshift is only very little and related to the wavelength of the waveform. In fact it is between virtually no time shift for the low pitched partials to almost half of the partials waveform period for high pitched partials.

The idea to make an allpass filter is simple. First a signal is filtered with a 6 dB lowpass filter. Then the lowpass filter output is subtracted from the lowpass filter input, which will regain what the lowpass filter threw away, effectively creating an extra highpass filter output. If these lowpass and highpass outputs would be added together again it would give an exact replica of the input signal. But before being added together the highpass output is reversed in phase.This means that low pitched partials remain virtually unchanged, but very high pitched partials will be almost reversed in phase. Still all partials will pass the filter. The 6 dB slope of the lowpass filter will take care that partials pitched in the mid range will have a phase shift somewhere in between, in fact a partial that is at exactly the cutoff frequency of the lowpass filter will have a phaseshift of exactly 90 degrees. When the phase shift is plotted in a spectral plot it will show a gradually smooth curve.

A common use for allpass filters is to put several in series, eight to twelve is common, and mix the output with the original input signal. As phase shifts accumulate in the chain of allpass filters there will be several frequencies where partials are in opposite phase and these will cancel out each other, while at some other frequencies partials will be in phase and come out at double the amplitude. This is the principle of a phaser module. By creating an extra feedback path from the output mix back to the beginning of the allpass filter chain the whole chain can be made resonant as well. By slowly sweeping the frequencies of the allpass filters a typical swooshy phaser sound is generated.

Another use for allpass filters is in the feedback path of echo delaylines. What the allpass filter does is mimic the behaviour of the coils in the recording and playback heads of a tape echo device. In general these heads were not of the most expensive type and together with the capacitive coupling with the rest of the electronic circuitry the heads would act like allpass filters. The effect is that when an echo repeats every repeat is slightly different which results in a much more natural timbral change in the decaying echoes as e.g. a straight digital delay that only uses a lowpass filter. So, when using an allpass filter in the echo repeat feedback path of a digital delay it will sound more natural. It is a bit like the echoing in a room where the walls are not exactly at ninety degree angles or that has reflective surfaces on furniture that is irregularly placed in the echo room.

Allpass filters can also be used to create filter slopes that have different slopes as multiples of 6 dB. Another use can be in RIAA correction filters used for cutting and playing back vinyl records. This RIAA correction filter has an almost straight 6 dB slope over the whole audio range.

In general allpass filters can be used to create specific effects where different phase shifts of partials will cause specific effects to happen in surrounding circuitry. E.g. allpass filters can be used to make a filter that will create pink noise from white noise, where the filter must have a virtually straight -3 dB slope. Such a filter is also useful to emulate the damping action of humid air on a sound that comes from some distance, a reason why allpass filter are also used in complex reverberation algorithms that take air dampness into account. Many digital synthesizers have a flat spectrum, comparable to a flat white noise spectrum. A filter that can tilt the whole spectrum can filter these digital synths in a way that their audio spectrum tilts down towards the high end with e.g 1 to 3 dB per octave. This will greatly enhance the spatiality in chorus sounds without creating the sense that the high end is filtered. There is a good psychoacoustic reason why this works, which has to do with how the two mechanisms that define sense of direction in the mind work. One of these mechanisms works below roughly 3.5 kHz and uses time delays and amplitude differences between the two ears, the other mechanism works above this 3.5 kHz and uses the combfiltering effect of the pinnae of the ear, the reason why it is possible to pinpoint the direction of a sound with only one ear. Just a little turning of the head will immediately give the correct sense of the direction of the sound, as long as it has some partials above 3.5 kHz. The human mind expects a certain balance in volume for the two audio ranges these two mechanisms work in, as the mind combines information from both mechanisms in defining the direction of a sound. If the volume balance is not like how it is expected from sounds in nature the mind gets confused and actually starts to refuse to generate a sense of direction at all. E.g. pink noise creates a strong sense of spatiality, but white noise does not. So, to make the spectrum of a digital synth more like the pink noise spectrum will in fact enhance spatial chorused and reverberated sounds. Analog synths are less perfect as digital synths and need less treatment or no treatment at all. In general these corrections are made when an experienced mastering engineer is making a master mix. But only few can afford the services of a really good mastering engineer, so it is good to know about these psychoacoustic principles and being able to make these corrections oneself. Conclusion is that the apparently lowly allpass filter is in many cases exactly the main secret in what makes the difference between something that sounds and something that sounds good and agreeable, and isn’t lowly at all.

12 dB State Variable or multimode filter

This type of filter has three simultanious outputs; a 12 dB lowpass output, a 6dB bandpass output and a 12 dB highpass output, which makes it a very versatile filter. This filter will basically split the audio range into three different bands. It allows for resonance on all three outputs, the amount of resonance can be set by a resonance knob. The internal architecture of the filter is quite simple, two 6 dB building blocks are used in series and are fed back with an inverted signal. If the feedback is exactly unity gain and the filter is exited with a small pulse it will start to oscillate on the cutoff frequency and oscillate forever. By allowing for slightly more feedback than unity gain the filter will in fact turn into a sinewave oscillator. This amount of feedback is set in the filter design itself, so the designer of the filter can define if the filter will allow for oscillation or not. To reduce the resonance, the signal at the point between the two filter building blocks is used as a second feedback signal. Which means that this filter type actually has two feedback paths that control its resonance behaviour, one fixed to set the maximum amount of resonance and a variable one that in fact reduces the resonance. The three outputs are taken from the points in the circuit after the two building blocks and after the inverted feedback point. The filter is cheap to build and not very critical. It is found on many of the cheaper analog monosynths from the eighties. The slope of 12 dB on the lowpass output gives reasonable filtering action but is not steep enough for all applications. In contrast the resonance can be very pronounced with a slightly whistling character. This makes it a filter that is a bit difficult to work with from a sonic point of view, it needs careful tweaking. The true power of this filter is in how the three outputs can be combined to create a huge range of filtering effects that are simply not available on other filters, but most of these effects are subtle. In general the filtering effect is not strong enough to be the module that is solely responsible for shaping the timbre, still it is very useful in combination with a waveshaping technique that does its part of the timbre shaping.

Combining the three output signals in a mixer creates a whole range of possible filtering functions. Mixing equal or differing amounts of the high and low outputs allows for a filtering curve with a deep notch. When the amounts of highpass and lowpass output signal are exactly equal, the notch will be at the cutoff frequency and the resonance knob will have not much audible effect anymore. This is because the resonance peak at the HP output is phase reversed to the resonance peak at the LP output, which suppresses the resonance fully when these two outputs are mixed equally. If the amounts of LP and HP are unequal the filter has an elliptic response, which also has a notch but not at the resonance point. Also, so some resonance will reappear. If there is more lowpass signal the resonance frequency will be lower than the notch frequency and when there is more highpass signal the resonance frequency will be higher than the notch frequency. By subtracting the highpass output from the lowpass output all audio is passed through, but a strong resonant peak can be created if the resonance is raised, creating a peak filter. This is because the subtraction will phase reverse one of the outputs, bringing the resonance peaks on the outputs in phase again. Subtraction is simply done by mixing the lowpass output with the phase reversed or inverted highpass output. This will need either an inverter module or a mixer which can invert its inputs.

The following filter curves are possible when mixing the three outputs of the filter in certain amounts:

Table 1: Possible state variable filter curves

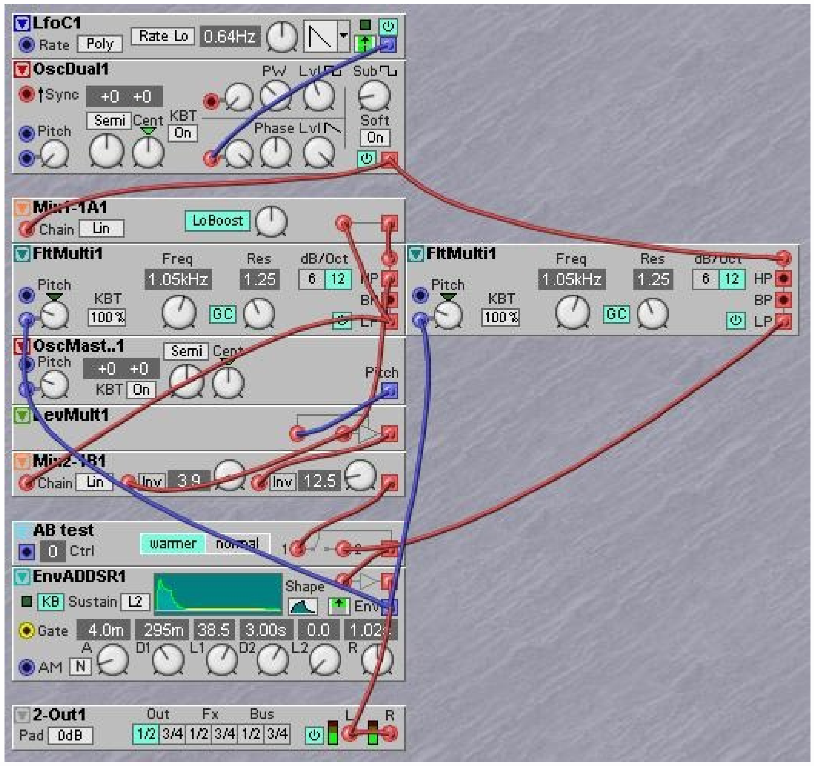
Type	Amount of LP	Amount of BP	Amount of HP
12 dB LP	100% or 0 dB	-	-
6 dB LP	100% or 0 dB	100% or 0 dB	-
6 dB BP	-	100% or 0 dB	-

The elliptical modes are useful for subtle filterings where a certain frequency needs to be removed or emphasized. In the case of an elliptic LP mode the cutoff can initially be set to a certain frequency and then by opening the HP mixer control the notch is created. The notch is swept in from very high at 1% HP signal to the cutoff frequency at equal amounts of LP and HP signal. If the notch is set to about an octave above the cutoff frequency the filter sounds steeper than when only the 12 dB output is used. Somehow this elliptic mode seems to sound best when the notch is on a relatively higher frequency when the cutoff frequency is low, when raising the cutoff frequency the notch should get closer to the cutoff frequency. To get this effect the amount of HP mix will depend on the cutoff frequency, but not with a simple formula. So, adjustment is best done by ear. Another effect of mixing in a bit of the HP output is that the resonance gets slightly reduced until it disappears at an equal mix of LP and HP output, which was explained on the previous page. Together with what appears to be a steeper slope due to the notch, this reduced resonance effect can subtly change the sound of the filter. This filter mode works well on rich and bright input signals, like thick and bright unison string sounds and on the cymbals in drumloops. Slow modulation of the cutoff can give e.g. a static drumloop a much more lively character.

There is quite a lot of phaseshift of all the frequency components in the output signal in respect to the input signal. This effect can greatly add to the apparent warmth of the sound when the filter cutoff is swept. The more the filter is set to an elliptic mode, the more the effect of phase shifting becomes apparent. The very subtle phasing effect this creates can give some nice warmth to the sound. However, this might be less desirable when filtering external audio sources, like recorded tracks with vocals, but again it can be a great effect on drumloops.

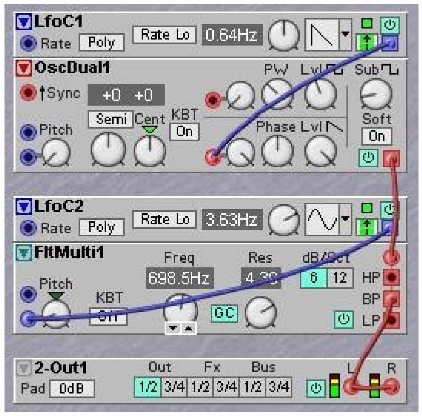
When the resonance is set to selfoscillation the sinus waveforms on the outputs have a phaseshift of 90 degrees for the bandpass output and 180 degrees for the highpass output in reference to the signal at the lowpass output. When additionally the bandpass output is inverted there are four sinewave signals available with respective phaseshifts of 0, 90, 180 and 270 degrees. Most 12 dB filters can resonate at very low frequencies if the cutoff is set to its lowest value and additionally a negative control value is patched to the frequency control input. When these signals at these very low frequencies are fed to the control inputs of four VCAs that feed four amplifiers and speakers, a sound that is fed into all VCA inputs can be made to rotate quadraphonically around the four speakers.

An additional feature of the 12 dB filter is that some part of the LP output signal can be fed back to the filter input. This will slightly boost the lower frequencies giving the sound more ‘beef’. However, this will also offset the two basic filter elements (the poles) used in the 12 dB filter which slightly changes the cutoff frequency and reduces the resonance. This will make the filter sound closer to a pure analog filter, as the filter elements in analog filters are never the same due to component tolerances. About –12 dB seems a good value to make the filter sound warmer and the resonance slightly less whistling. When increasing the feedback to about –1.5 dB the filter will start to sound distinctly different, often a thick and fat sound with a slight distortive character will result. But this can quickly make a timbre sound ‘muddy’, so careful adjustment by ear is again necessary.



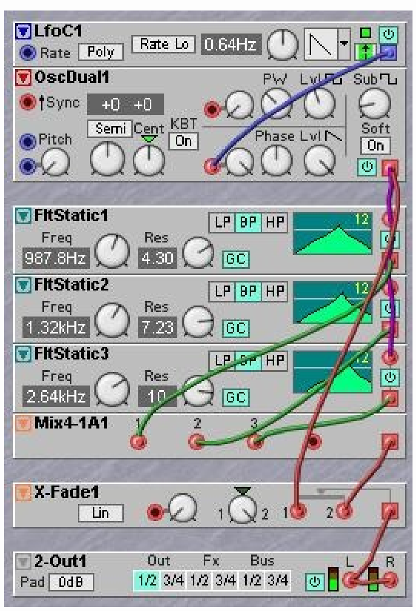
Peak filter

The peak filter is a special case. If there is no resonance the peakfilter acts as a simple allpass filter without much audible filtering effect. But when the resonance is raised to a fairly high value a strong resonant peak is introduced that can be swept through the audio range. This can be very useful to breath some more life in sounds from drum computers and sampled loops. On the G2 the peak mode is already pre-wired in the 12dB multimode filter. It can be used by setting the filter to the 6dB mode and using the bandpass output. With this setting the bandpass output is actually an allpass filter with a resonance peak when the resonance control is opened.



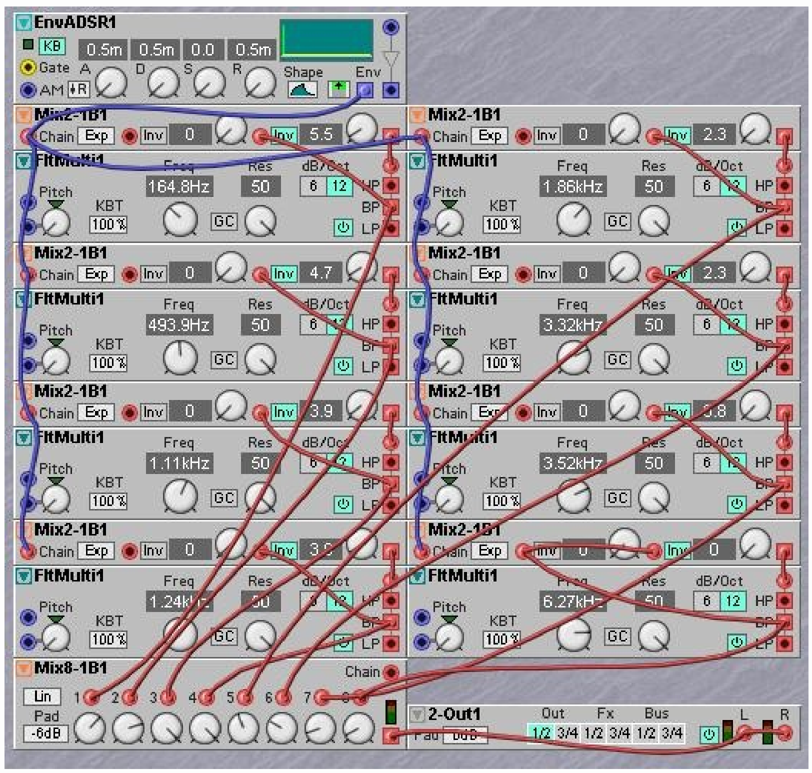
Combining two or more state variable filters in bandpass mode

Another interesting variation is to put a few 12 dB filters in parallel and use the bandpass outputs to create a fixed or controllable resonant structure with several resonance peaks. By simply mixing the outputs of three such filters in a 3-input mixer, the ‘formant filter’ module of some vintage analog modular synths can be conveniently recreated with about the same DSP-resources as needed for a single 24 dB filter. A crossfader can be used to crossfade between the input signal that goes into the filters and the output of the mixer than mixes the BP signals to control the depth of the effect.



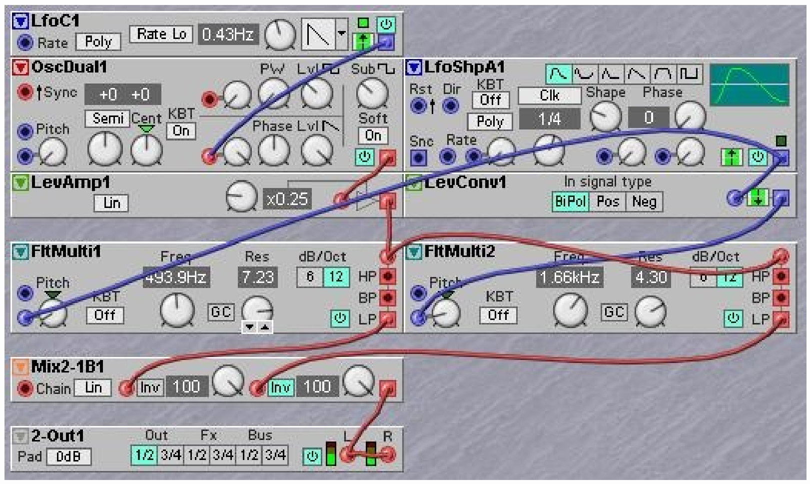
Modal synthesis

Strongly resonating bandpass filters are often used in a synthesis technique named modal synthesis. With this technique a number of bandpass filters are excited with one single pulse, causing them to ring for some time. When these bandpass filters are each tuned to a partial in the sound to be synthesized, quite convincing emulations of certain groups of acoustic instruments can be created. It is important to have very detailed control over the ringing time. An extra degree of fine tuning for the resonance can be accomplished by feeding back a phase reversed portion of the bandpass output to the filter input with an extra mixer module at the input, while the resonance of the filter is set to maximum. If the mixer used uses exponential control the ringing time of the filter can be set more precisely by careful adjustment of the feedback mixer knob.



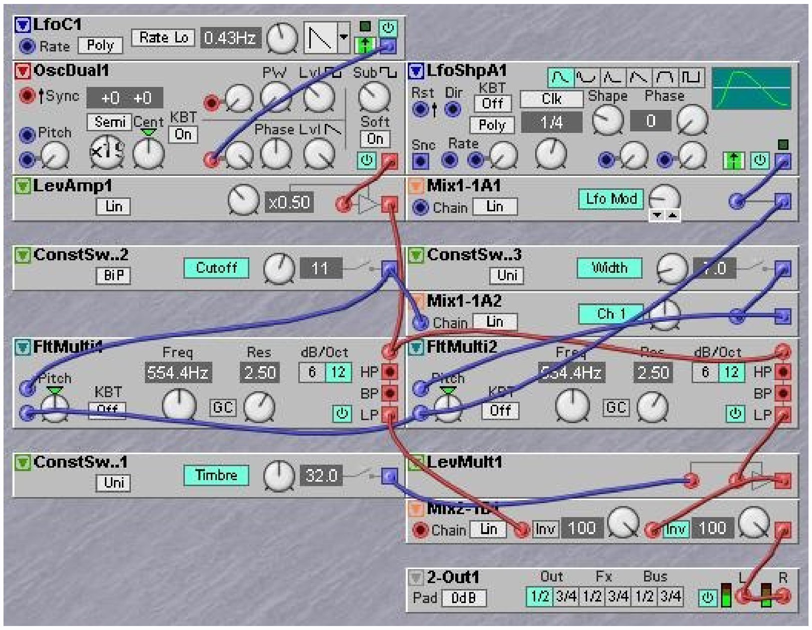
Variable width bandpass filters

Another interesting filter variation is to use only the LP outputs of two parallel 12 dB filters, invert one of the outputs and mix them together in a mixer. In this configuration, either the resonance settings must be kept equal on both filters or the Gain Control must be set to off (GC button). When the Gain Control is off there should be an extra signal attenuator before the inputs of both filters, to prevent possible clipping on high resonance settings. If the mixed output signal amounts are equal and the non-inverted filter is set to a higher cutoff frequency a bandpass filter is created, as only the band between the two cutoff frequencies is passed. This can be explained by imagining that the output signal from the lower tuned filter is also present in the output of the higher tuned filter, so by subtracting the output of the lower tuned filter from the output of the higher tuned filter only the difference between the two output signals is left, and that is the band between the cutoff frequencies of the two filters. This bandpass filter has two slopes of 12 dB and can be set to a variable width by control of the respective cutoff frequencies. By raising the resonance of the filters two resonance peaks will appear at the corners of the band. A filter with a variable bandwidth can be a great tool to mangle vocals, creating telephone sounds, etc.



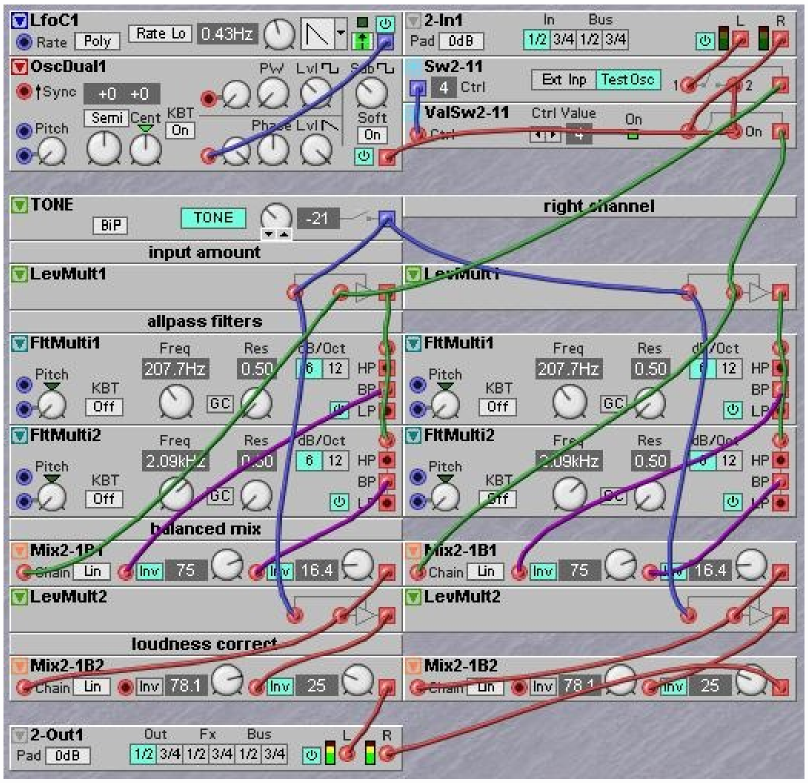
Morphing from lowpass to bandpass

A variation on this filter is when the lower tuned filter output is controllable in level by a knob. This will create a filter that is smoothly controllable between a lowpass filter and a variable width bandpass filter. Instead of controlling the amplitude of the lower tuned filter the higher tuned filter can be controlled. This creates a filter with a controllable slope between lowpass and variable width bandpass, but it will go through a range halfway where the filter combination is in elliptic mode. Especially this last mode is interesting as it can make the filter appear more steep when the higher tuned filter is tuned to about a half octave above the other filter and its output is at a value somewhere around –6 dB. By carefully listening while adjusting the level the effect on the slope can be heard very well and tuned to a sound you like. When the higher tuned filter is tuned to some two to four octaves higher the filter combination will appear to have more ‘spit’, again adjustment is best by ear. Important is that the resonance settings on both filters should be exactly the same, as this gives the most interesting timbre shaping ranges for this filter combination. Whenever one of the filters is retuned the level knob of the highest tuned filter should be adjusted as well. This parallel combination is more interesting as putting two filters in series to get a steeper filter. The sonic difference between paralllel and serial becomes clearer when the resonance is raised to a fairly high level. In paralllel mode both resonant peaks come out much better in the mix and have more timbre shaping power, which makes the filtered sound more interesting.



An 'overeasy' tone control

Many times it is necessary to have a simple control over the tonal presence of a sound. Presence of a sound in a mix does not only depend on its volume, but also on the overall tonal balance. Digital sounds often have too much energy in the very high of the spectrum. Using just a LP cutoff filter to temper the high is not enough. What works much better is to give the sound the same kind of filtering that is used to filter pink noise from white noise. This means that the whole sound spectrum should be tilted by a staight line. When using two 12dB filters in allpass mode this straight downsloping curve can be approximated sufficiently to create just the right amount of psychoacoustic tone filtering to increase the apparent presence of a sound. The idea is that two allpass filters are tuned about a decade apart, around 200 Hz and 2000 Hz are good values. The outputs are added together in a fixed balance and this sum is either added to or subtracted from the input signal by a controllable amount. The frequency dependent phase shifts caused by the allpass filters will cause the specific slope when it is combined with the input signal. The only disadvantage is that there is some changes in overall outputs level, but with a simple trick these can be corrected for. The result is a tone control with one single knob that corrects the loudness levels in the frequency bands in a way that the attention will shift to e.g. the melodic parts of the signal. And it is exactly this shift of attention that improves the presence of the instrument in the mix. This filtering is very useful to give digital instruments a warmer sound, but can also be used to either boost the bass or boost the extremely high end of the spectrum.



Spectral crossfader

A very interesting property of the state variable filter is that it can be designed in a way that it does not have three outputs but has one output and three inputs. From one input only the low part of the spectrum will be passed through, another input will only pas a small band and the third output passes only the high part from its input signal. In this particular design the filter can act as a spectral crossfader between two different input signals. An example is when a drumloop is fed into the LP input and another drumploop is fed into the HP input. This will result in a mix that has the low drums from one loop and the high cymbals from the other. By turning the frequency cutoff knob fully from one side to the other there will be a crossfade from one drumloop through the other, fading in one signal from the low while the other signal is pushed away into the high. There has been only one commercial instance of this filter made by Modcam and designed by Nyle Steiner, who is also known for inventing the Steinerphone, and electronic windinstrument that would later become the Akai EVI windinstrument. In the next chapter, that is all about building your own filters with simple filter building blocks, an example of a spectral crossfader patch based on a G2 24dB filter patch will be given.

The 24 dB classic ladder filter

This type of filter was invented by Robert Moog in the early sixties and used on all the Moog synthesizers. The filtering action is quite pronounced and it has a very nice sounding resonance. The most peculiar thing about this filter is that when it is used on an unisono sawtooth sound created with e.g. three slightly detuned sawtooth oscillators, it tends to give the sound a distant spacial character. It is particular the sawtooth waveform that has this effect, when using pulsewidth modulated waveforms the reverberant character seems to disappear and the sound sounds much closer. As the effect of the filter on sawtooth waves is experienced as very beautiful by most listeners, the filter has gained an almost legendary status amongst musicians. The filter is perfect to create lush padsounds and roaring basssounds as it has a great timbral shaping effect in this sort of application. Technically, the filter is designed as two symmetric cascades of four transistors with capacitors between the transistors. By driving a variable current through the transitor cascades the cutoff frequency of the filter is controlled. The filter needs exactly the same type of exp/lin converter as an analog oscillator to be controlled. The filter has gained its nickname ladderfilter from the schematic of the filter circuitry, where the two transistor cascades look a bit like a ladder. The Moog filter was patented by Moog around 1965 and for a long time other manufacturers could not use the design. At the moment when the design could be used by others, chips had become common and 24 dB filters were made with voltage controllable operational amplifiers in a chip. One of the last synths that was equipped with a true ladder filter was the Roland TB303, although Roland made a little change in the design. This design change gives the TB303 filter a completely different, much more agressive sound, with a lot more spit when compared to a Moog filter. The TB303 was initially a marketing failure, but years after production was ceased it suddenly became very popular in Techno music styles. This caused many synth manufacturers in the nineties to suddenly started building TB303 clones, some with the ladder filter as made by Moog and others with the TB303 modification.

Other 24 dB filters

At the end of the seventies two chipmanufacturers started to produce complete 24 dB filters in a single chip solution, Solid state Music and Curtis Electromusic Specialties. The chips made by these manufacturers have been used extensively in the analog polyphonic synthesizers of the eighties. Production of these chips has long ceased, but their sound remains in the massive amount of analog polysynths still in operation today. These days manufacturers of analog modular synths tend to use standard components again, simply as it is hard to get hold of a bunch of these chips from the eighties.

24 dB filters can be emulated very well digitally and manufacturers of digital modular synthesizers have a choice of how to program filters and what their qualities should be. It is hard to say something in general about these digital implementations, as each manufacturer seems to use one of their own little tricks to give their filters a good sound. Musicians in general tend to think that digital filters sound less warm than analog filters. This can be partly explained by the fact that the signal from digital oscillators tends to be brighter than that of analog oscillators. Additionally, analog circuitry can have small nonlinearities, which cause a little harmonic distortion which slightly colorates the timbre. The issue is very subjective and in general it doesn't really make much difference in a final mix, as long as the mix itself sounds good. In practice digital filters do a very good job and have the additional advantage that, because of their exactness, combinations of filters can be made that are very difficult to do on analog modular synthesizers, unless the analog filters are really top quality.

All 24 dB filters work very well with the earlier descibed exciter/resonator synthesis model. When resonance is set to a medium to fairly high level the flanks in sawtooth and pulse waveforms create very nice resonance effects. It is not uncommon to add a little saturation distortion after the filter, to create some extra character. In general the 24 dB filter is easy to tweak as it basically does only lowpass and resonance. But small changes in the cutoff frequency can have quite an effect, because the filter cutoff slope is steep. So, it is important to tweak the filter carefully while listening well to the minute effects, especially when additional saturation is applied.

Using filters when doing audio processing

The four audio inputs on the Nord Modular G2 can be put to good use to process all sorts of audio material, like sampled loops, drumcomputers, audio from a computer, etc. The way audio processing modules can be freely patched in any order makes the G2 an ideal EFX machine. There are three basic types of audio processing available on the G2; filterings, distortions and time displacement effects. Filterings and distortions are used to lively up static samples or give sounds more beef. Time displacement effects are all effects where a time delay of the audio is used to create a special effect, like echo delays, reverbs, flangers, etc. Due to the modular nature of the G2 all three types of audio processing can easily and conveniently be combined to allow for a virtually unlimited range of effects. It is also possible to extract control signals from audio, examples are the loudness contour and a signal that can make an oscillator track the pitch of an incoming monophonic signal.

Filtering can be an effect in itself, but it can also be put to very good use to support other effects in a subtle way. As when processing audio it is many times needed to first filter parts of the sound, with the purpose to have an effect in only a selected part of the audio range. A common example is to have a distortion effect work only in the lower or middle range and not on the higher frequencies, with the purpose of giving this distortion a warmer and grungy effect.

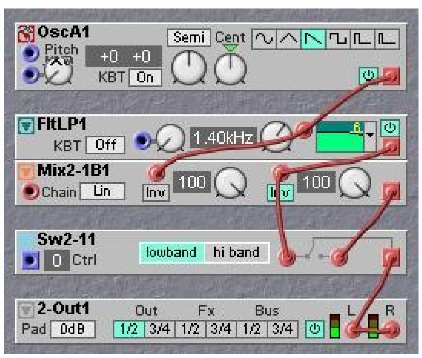
Crossover filters

The exactness of digital filters allows them to be used to do 'arithmetic' with the passbands. Remember that when the 12 dB filter was descibed it was explained how a variable bandwidth bandpass filter could be made by subtracting the output of one filter from the output of another filter. This can be done equally well using 6dB to 36dB non resonant filters and resonant filters that are set to exactly the same resonance setting. The first possibility here is to create a crossover filter. Such a filter splits the audio spectrum into two bands, and when these two bands are added together again the exact same sound should return. If the output of a lowpass filter is subtracted from the input signal, the part that is thrown away by the filter is regained from the input again. This signal will basically be a highpass signal, although the cutoff slope of this highpass signal is not equal to the cutoff slope of a proper highpass filter. So, if the lowpass filter is 24 dB the regained high will not have a low frequency rolloff slope of 24 dB. In practice this is not much of a problem, as the mind perceives the effect of a highpass filter in a different way as the effect of a lowpass filter. But our main purpose here is to create two bands that when added together again will recreate exactly the original input signal again.

A common use of crossover filters is to give each frequency band a different effect and then add them together again. A good example is when a thick chorus is applied to a sound, this will make low frequencies muddy and the very high frequencies can become like a high pitched buzz making the sound loose definition and causing conflict with sounds that have by nature a very defined high. In many cases chorus works best if it is applied to the range between 500 Hz and 4 kHz. This is about the range of the ear where the sense of spatial direction takes place. A chorus or unison effect in this frequency range will give a sense of a natural space.

When the spectrum is split into two parts with a crossover filter, one of the bands can again be split by another crossover filter. This way a multiband crossover filter can be made. These kind of filters are commonly used in filterbanks, multiband compressors or multiband special effects.

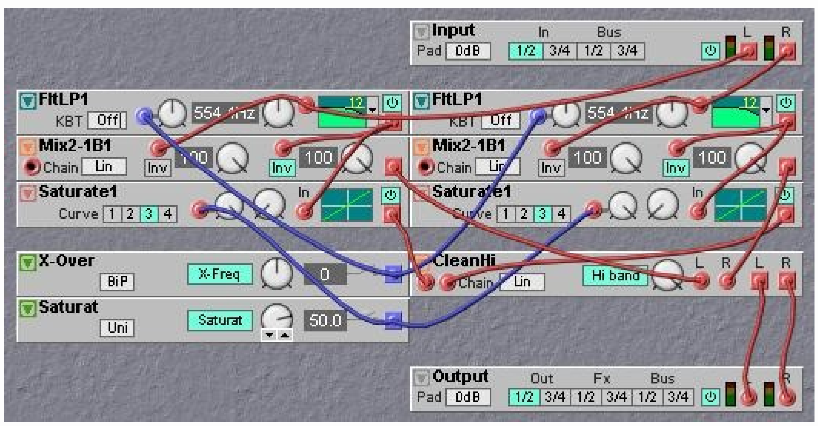
A typical crossover filter has one input and two or more outputs. When the outputs of a crossover filter are mixed together again, this mix should preferrably give an exact replica of the original signal. This last property is sometimes referred to as phase linearity. There is no dedicated crossover filter on the G2, but making a crossover filter is actually quite easy to do. The basic idea is to recover the high part of a signal that is thrown away by a lowpass filter. This is done by extracting the difference signal between the input and the output of the lowpass filter. Imagine that the ouput of the filter is mixed in reverse phase with the signal that goes into the filter. As the low part of the signal is in reverse phase with the low part of the unfiltered signal, these two parts will be cancelled out in de mixing. But as there is no high part in the filtered signal anymore only the high part of the input signal will be present at the output of the mixer. There is one requirement for this to work properly, and that is that the filter must have a gain of exactly one (be at unity gain) in its lowpass band, so it does actually cancel out exactly in te mixer. Pure analog filters are not precise enough for this use. In fact, crossover filters is an application where digital filters have a definitive advantage over analog filters, as they can have exactly unity gain in the passband. A good filter to use on the G2 is the non-resonant lowpass filter named FltLP. The advantage of this filter is that there can be no possible coloration of the timbre due to resonances in the filter. The FltLP can be set to slopes of 6dB per octave up to 36dB per octave in 6dB steps. Additionally it has a totally flat lowpass band at exactly unity gain, so it is the ideal candidate to be used to make crossover filters.



Have a look at the illustration above. A sawtooth waveform from an oscillator is fed into the input of a FltLP lowpass filter. The sawtooth is also fed into the input of a Mix2-1B filter. The output of the filter will be the 'low band' in the crossover filter. This low band is fed into the second input of the mixer. The signal on this second mixer input is brought into antiphase by activating the Inv button. In practice this means that the signal on the second input is not added, but instead subtracted from the first input. On the output of the mixer is the difference signal between the input and the output of the filter, which is 'all but the low band' or whatever the FltLP has thrown away. The advantage of splitting the audio range with this technique is that there are no nasty phase shifts that would occur if both a lowpass filter and a highpass filter were used in parallel, which perhaps many would think to be the obvious way to do it. In this simple crossover filter the high band has a slope that is not as steep as the lowpass filter, e.g. when the lowpass filter is set to 24dB the highband does not have a slope of 24dB. In practice this is not a problem as highpass filters do in general not have to be as steep as lowpass filters to have the right sort of sonic effect. Experimentation and good listening to different settings is necessary to get a good idea of what is happening when tweaking the filter cutoff or changing it to another cutoff slope. Notice that lowpass filters and highpass filters psycho-acoustically work out totally different on sounds. An interesting example is a small sixties transistor radio. The little transistor radio can physically not reproduce frequencies below some 200 Hz, it is just too small for that. In fact, its speaker is an effective highpass filter set to some 200 Hz. Still, when listening to such a radio there is a definite sense of bass notes. Psychoacoustic research has found that if there is only a hinge of harmonics present that belong to a low pitch that is too low to be reproduced by a speaker, a person will reconstruct the sense of this low pitch in the mind. Which actually applies to any highpass filter. This basically means that whatever a highpass filter throws away the mind will try to reconstruct. The mind does not do this with a lowpass filter. Which means that the issue of slope steepness works out totally different on lowpass and on highpass filters.

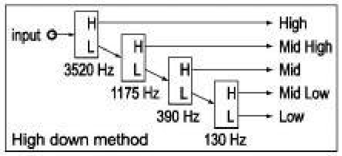
When the two-pole A/B switch is swapped for a crossfade mixer it is possible to smoothly fade between the low band and the high band. If the crossfader is in its middle position the original signal is heard, as this is an equal mix of the low and the high bands. When the crossfade position is moved it will cause one of the bands to be attenuated while the other gets more emphasized. This is very similar to a shelving equalizer where the zero dB point is exactly at the cutoff frequency of the lowpass filter. But this simple patch has several advantages over a standard shelving EQ. First is that the cutoff frequency can be controlled and modulated between extremely low and high frequencies, second is that the balance between the bands can be controlled and modulated between full lowpass and highpass slopes. When used on an audio loop it is possible to fade a loop away in either the low or away in the high, depending on the setting of two knobs, the knob on the crossfader defines if the fade is going into the low or the high and the cutoff frequency of the filter does the actual fade. Third and most desirable advantage is that before the bands go into the crossfader both bands are separately available and can have different additional effects inserted at this point. These effects would have their effect in this band only.

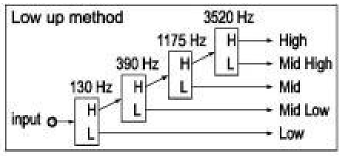
Crossover filters are always good to use before a distortion module. The idea is that distortion often generates a lot of high harmonics which might conflict with the high harmonics already present in the signal. Distortion is often preferred to happen only in either the lower or in the middle ranges of the audio range. By preventing higher frequencies to go into the distortion module much less high harmonics are generated, but the original high is lost as well. By using a crossover filter and mixing the distorted low with the undistorted high, there is suddenly an extra timbre control on any type of distortion. Note that the main distortion will appear to happen in the ranges just above the cutoff frequency of the lowpass filter in the crossover.



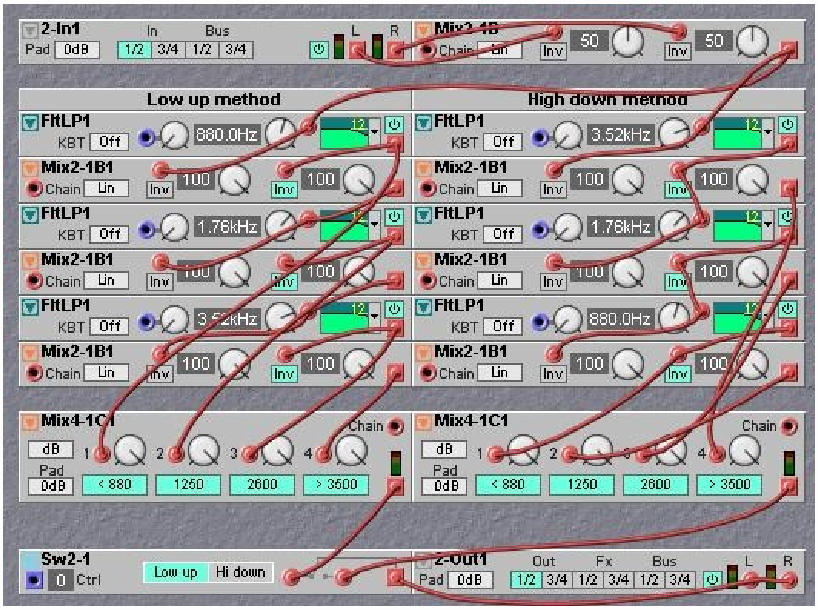
Multiband filters

If more than two bands need to be processed, only one extra filter for each band is needed. So if four bands are needed only three filters are actually necessary. Such an array of bandpass filters can be build from either the lowest band up or the highest band down, which in practice does make a difference on how each band sounds. The idea of this cascade of crossover filters is to split a signal in two bands named L and H, then one of the two bands is split and one of them is split, until there are enough bands. The illustration shows the idea better than words can describe it.



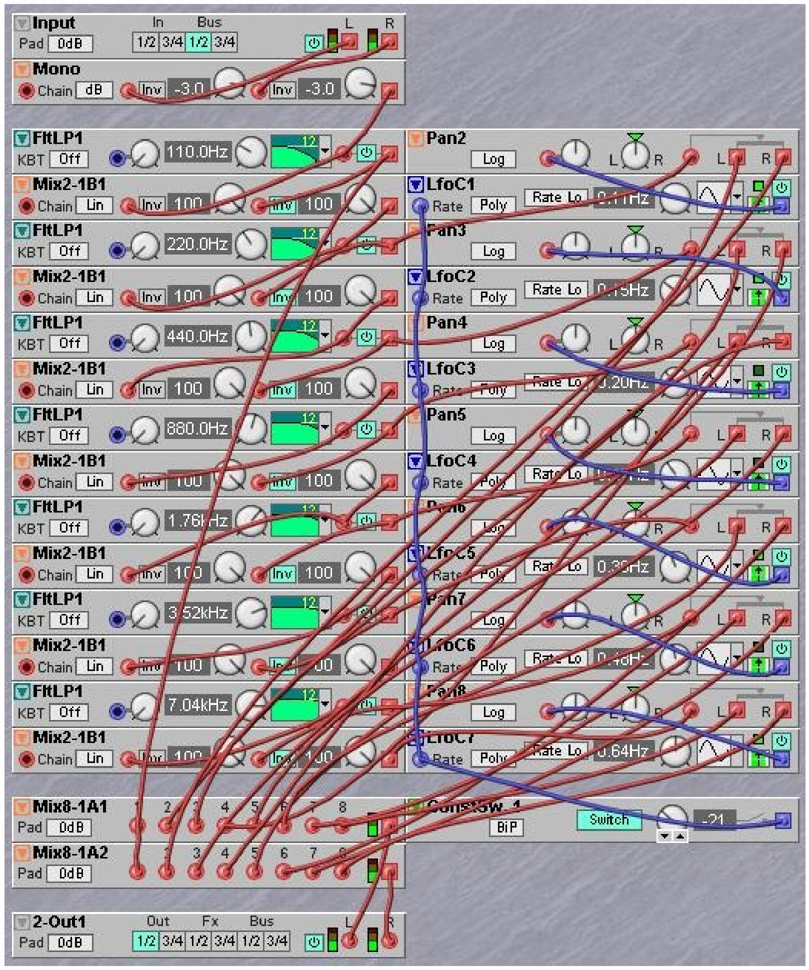


When using this method, the filtering accumulates through the array of crossover filters. If the bandfilters are build up from the low band upwards the lowpass slopes of a band will appear less steep and the highpass slopes appear steeper, while when building from the highest band downwards the bands appear to have steeper lowpass response and less steep highpass response. When listening to only one band in the middle, the low to high method appears to give a bit a higher tuned band than with the high to low method. The choice which method to use is basically a matter of personal taste and might also depend on the sort of audio that is filtered. It is handy to make two similar patches, one with the low up and the other with the high down method and check out which one best serves a particular situation. You will notice that the effect of the bands can be quite subtle, making this type of multiband crossover filter well suited for subtle tonal control on instruments in a mix. This is probably not the type of bandfiltering one would use to create a mastering multiband equalizer, as the bands appear quite shallow. But for the purpose of applying selective distortions and mixing all the bands together again in the end, this bandfiltering works out pretty well. The big advantage is that the crossover frequencies can be freely set to define the width of the bands and virtually any number of bands can be created.



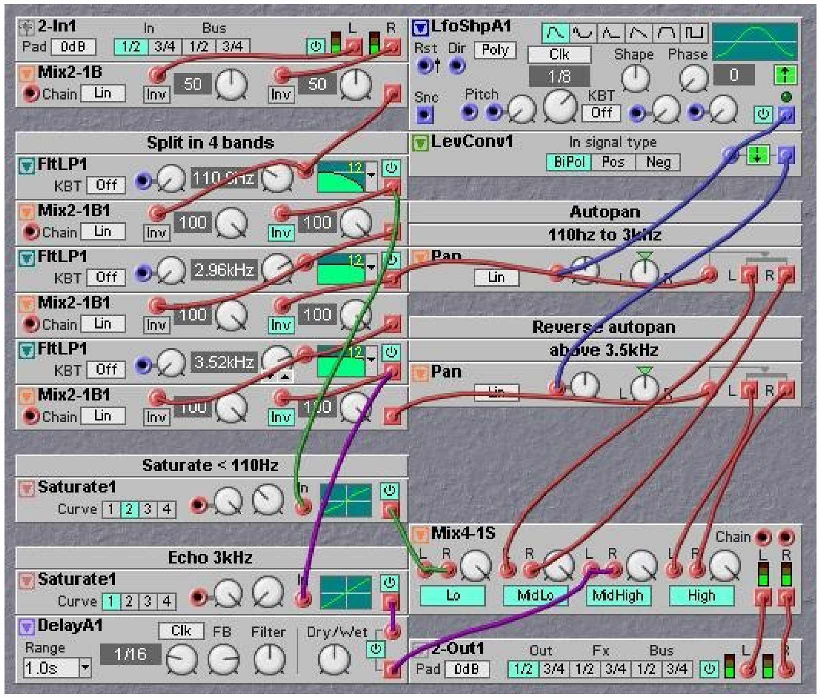
Multiband panner

An example of an interesting stereo effect on a mono signal is to split the audio range into e.g. eight bands and then use separate panners with their own lfo on each band. This will transform a static mono signal into a lively moving stereo signal.



Grunge in low with echo in a small mid band and some panning

The next patch illustration can be used on e.g. a sampled drumloop. There is saturation on the kick, an area around 3kHz has a MIDI synced echo. The area between 110 Hz and 3kHz plus the highest band above 3.5kHz are autopanned in reverse with a MIDIsynced Lfo.



Resonant filters

Introduction

This chapter is about the type of filters that are commonly used in the individual voices of an analog synthesizer patch. To deepen your practical understanding of filters the Nord Modular G2 will be used to build your own filter types.

Background

The original idea behind the classic VCO->VCF->VCA ‘subtractive synthesis’ patch, is actually based on how acoustic instruments work. In this ‘mother of all analog synth patches’ an oscillator signal goes into a filter and then into a controllable amplifier controlled by an envelope signal. Let’s first look at a real-world example; an acoustic string instrument. This instrument will have a resonant body and strings will be attached onto this body in a way that the body can be made to resonate. The resonant body is very important, without it there is only very little sound. What happens in this string instrument is that the vibration of the string is transfered onto the resonant body and the resonance will give the sound its volume and timbre. It is the kinetic energy of the vibrating string that excites the resonant body, and the resonant body transforms the energy from the string into the ‘body of the sound’. The strings will define the pitch and the shape and material of the resonant body will define the overall timbre. In general a resonant body will easily resonate in some parts of the sound spectrum and less easy in other parts. This will favour some frequency bands in the spectrum and these frequency bands are named formant areas, as they ‘form’ the timbre of the sound.

It was exactly this principle of an exciter, which controls the pitch, and a resonator, which shapes the timbre, that inspired early synthesizer designers like Bob Moog to use a resonant filter driven by an oscillator to create sounds. The oscillator acts as the exciter and the resonant filter shapes the timbre by creating a formant effect at the resonant frequency of the filter. Waveforms with sharp edges, like a sawtooth wave, can strongly excite the resonance in a filter, turning the filter into the electronic equivalent of an acoustic resonant body. A good example of a filter that lets iteself be excited very well is the resonant ‘four-pole ladder filter’, designed and patented by Moog in the early sixties. This filter turned out to be só useful musically that many people still considered it the best type of filter around. These days’ digital modular synthesizers come with several filters and undoubtedly one of them is an emulation of the ladder filter. But the digital emulations have one disadvantage over analog filters, which is that they are often too perfect. The Moog ladder filter was in practice far from ‘theoretically perfect’, due to wide component tolerances and small non-linear imperfections in the components. Apparently the imperfections in the Moog filter add to the sound in a positive way. It is a good idea to try to build a filter yourself to understand better why certain filters can have a certain sound. With the DIY filters descibed in this chapter you can explore the inner workings, deliberately add imperfections and make any possible variation on filtertype and slope steepness within the concept of the typical four-pole filter.

The ladder filter

The Moog ladder filter is made by cascading four basic filterblock sections that each have a cutoff slope of 6dB per octave. An individual filterblock section is commonly named a pole, a name derived from a parameter when doing calculations while designing a filter. A pole is not really an actual discrete electronic circuit, but as many manufacturers have used the word pole for so many years it has become common to talk about two-pole filters, four-pole filters, multi-pole filters, etc.

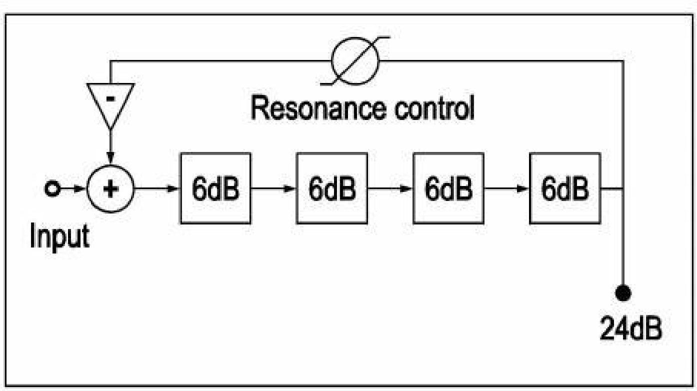
Patching your own filters

When a digital modular synthesizer has a couple of 6dB lowpass filters present it is actually possible and quite easy to patch filters like the four-pole filter yourself.

In a resonant four-pole filter the poles are cascaded in series and the output signal at the end of the cascade is fed back to the input of the first pole to create a feedback loop. Feedback is very important as feedback is always necessary to create resonance. Each of the four poles will cause a very short delay on the signal. This delay is only 1/8th of the length of a single cycle of a waveform tuned to the pitch that is equal to the cutoff frequency of the filter pole. The four poles together will cause a total shift of four times 1/8th, so one half of this waveform cycle. If this delayed signal is additionally reversed in phase this 'inversion' will create an additional 'phaseshift' of another 180 degrees. This causes the delayed and inverted waveform at the output of the four poles to appear to have a delay of exactly a full cycle of the waveform, and so lag one cycle of the waveform behind in respect to the input signal. But note that it is only a sinewave component at a pitch that is equal to the cutoff frequency of the filter that will have this exact one cycle delay. Partials in the sound that are not at this cutoff frequency will have different phaseshifts. If the delayed and inverted signal is fed back to the input of the four poles it will reinforce the input signal and thus create the wanted resonance. In practice the input signal is mixed with the feedback signal before entering the first pole and output is taken from the output of the last pole. This will give a cutoff slope of 4 times 6dB, so 24dB. The resonance is defined by the amount of feedback. An important thing to remember in your experiments is that to make the filter resonant the feedback signal must always be inverted or phase reversed to be able to cause resonance, without this phase reversal the filter won't resonate at all. This also applies when using two, three or more than four poles in your filter design. By using a G2 two-input mixer module with an invert button on its inputs, the phase reversal can be simply done by pressing the proper invert button.

Basic schematic

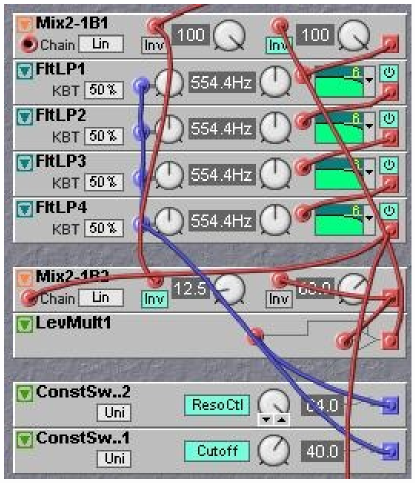
Following is a schematic of the classic four-pole filter.



Basic patch

On the G2 a basic DIY 24dB filter patch looks like this:

A sawtooth oscillator is used as the sound source. The input to the filter is the first input on the mixer, the knob on the second input of the mixer is the resonance control. Note that the second input has the Inv button light up to show that it inverts its input signal. So, this invert button causes that phase reversal that is necessary in the resonance feedback loop. When patching this little patch and experimenting with the resonance knob there are some 'flaws' that can be easily noticed. First observation is that the overall volume drops slightly when the amount of resonance is raised. The apparent drop in volume when the resonance is opened is in fact a property of any basic 24dB lowpass filter. In practice this is often compensated by adding some extra input signal when the resonance knob is opened. The second observation is that the resonance is only shallow at 100% feedback. This means that adding extra amplification in the feedback path must increase feedback over 100% here. Theoretically the resonance feedback signal must be amplified by 12dB for full resonance, as each 6dB filterblock attenuates –3dB at the cutoff/resonant frequency. A third observation is that it would be nice to make the resonance modulatable, which can be done by inserting a multiplier module in the resonance feedback path. These three features can be accomplished by adding only two modules, a two-channel mixer and a multiplier.



Adding options

The two-channel mixer is inserted after the fourth pole and the output of the fourth 6dB filter is fed to the Chain input of the mixer. The output of the mixer is connected to the input of a multiplier and the output of the multiplier is connected to the resonance feedback input on the first mixer. It should be obvious that the multiplier can now be used to set the resonance by a control signal. The simplest way to raise the gain of the feedback loop is to raise the gain of the second mixer beyond unity gain by connecting its output to the second mixer input and feeding a little bit of the mixer output back to this input. This connection of a mixer output to one of its inputs will change the mixer into what is named an integrator circuit. To understand what this connection does you must realize that an input on a G2 mixer module will see its own current output sample as a previous sample. Take in mind that the DSP calculates the patch 96000 times a second. When in a new calculation a module input is taking a value from the modules' own output, the output value to use must have been calculated in the previous calculation round, simply as the output value for this new calculation round is not yet finished, as in fact the module is just about to be calculated. What this means is that this feedback connection has a time delay of exactly one sample, and the output value is what is named the Z-1 sample value. So, the output of a mixer on the G2 system acts as what is named a Z-1 sample to its own input. The effect of an integrator is that it will create a small high frequency damping and act on an audio signal as a soft lowpass filtering. An integrator is often used to make a circuit with a feedback loop more stable in the high frequency ranges. In analog circuitry it is often used to prevent radio frequencies to leak into the audio circuit. In digital systems it is used to prevent feedback systems to start resonating on half the sample rate with the side effect that the sound also sounds a bit warmer and more like analog circuitry.

Filter warmth

What the integrator feedback connection means for the four-pole filter is that in this mixer module with its feedback connection, the lower and middle frequencies are boosted more than the higher frequencies in the resonance feedback loop. This causes a small high frequency damping that in practice makes the resonance more stable at high cutoff settings and the additional advantage of giving the DIY filter the warm sound of an analog filter.

Tuning the maximum resonance

To tune the resonance range first open the resonance feedback knob on the first mixer fully and feed a value of +64 to the multiplier control input. Then close the second input knob on the second mixer before making the mixer feedback connection. When the connection is made start to slowly open the input knob. You will hear the resonance becoming more pronounced. When the knob display is around a value of 75 the resonance has become sÛ high that it will make the filter go into selfoscillation. Turn back the knob slightly until the display shows 68. At this setting the resonance is quite pronounced and musicalkly very useful, but the filter will not yet go into selfoscillation. Now tweak the control signal on the multiplier control input between 0 and +64 and you hear that the patch has proper resonance control.

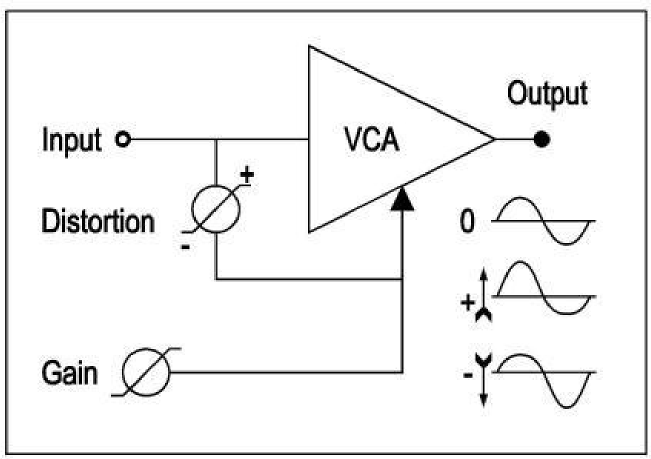
Balancing the output level

The drop in volume at a higher resonance can easily be corrected by feeding a little bit of the input signal to the second mixer. Note that as the output of the second mixer is inverted at the second input of the first mixer the input signal fed to the second mixer must be inverted as well to work correctly. When the resonance is increased a little bit of the input signal is added through the gain control that sets the resonance. More resonance will also add more input signal and correct the output level. How much the mixerknob for this extra input signal must be opened can best be set by ear. Tune it in a way that the apparent loudness at high resonance settings is about equal to the loudness at no resonance.

This 24dB filter design is a good base for doing all sorts of filtertricks. In the example the four 6dB filters are tuned to the same cutoff frequency, just like in the original ladder filter. But it is fun to set them to slightly different frequencies and/or give them different keyboard tracking settings. A common trick is to detune the fourth 6dB one to two octaves below the other three. This will slightly change the cutoff slope and detune and temper the resonance. With this setting a resonance sweep will appear less wobbly when gliding through the harmonics of e.g. a sawtooth wave.

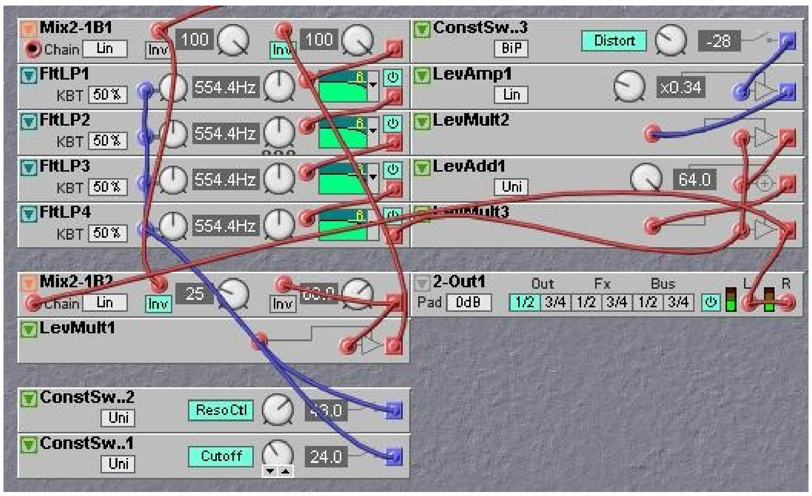
Adding some slightly grungy even harmonic distortion

Next step in out filter design is to add a little distortion to mimic the non-linearities in an analog filter. The choice is between either even harmonic distortion or odd harmonic distortion. Even harmonic distortion is more interesting as it can make a filter appear both steeper and less steep. The idea is that even harmonic distortion in fact produces all harmonics except the fundamental of the distorted signal. These new harmonics can actually be subtracted from the undistorted signal, which makes the filter slope sound steeper. And by adding them to the filter output signal the filter will appear less steep. There is a simple trick to produce even harmonic distortion that works on all synthesizers that have a linear VCA and a means of inverting a signal. The idea is to use a fixed control voltage on a VCA control input in such a way that the VCA amplification is exactly one or unity gain. Then a little bit of the input signal into the audio input of the VCA is mixed with the fixed control voltage. When the input waveform is positive the gain of the VCA will increase slightly while when the inputs signal is negative the gain will be slightly reduced. It is like the signal is compressed when negative and expanded while positive. When set this way the VCA will generate the second harmonic of each sinewave partial that is present in the input signal. The amount of distortion is set by the amount and the polarity of the signal that is added to the fixed control value for the VCA. Beware that the maximum level of the modulation signal is about half the fixed voltage or control value, exceeding this value might cause some true analog VCA's to stop working.



On the G2 even harmonic distortion is very easy to patch, the audio signal is fed into a multiplier module and the control input receives a control signal from a LevelAdd module that is set to +64. The +64 value will cause the multiplier to pass its input signal unaltered at unity gain. By feeding a bit of the input signal to the input of the LevelAdd module the multiplier will start to produce even harmonic distortion. Another multiplier, controlled by a knob control module that can be set between –64 and +64 units, controls the depth of the effect. This knob controls the amount of distortion and also if the generated harmonics are mixed in phase or in anti-phase. The useful range for the knob control is between –32 and +32. On the G2 the control value must be set to a negative value if the LevelAdd is set to +64 units, or the control must be set to a positive value if the levelAdd is set to –64 units, to create the proper type of even harmonic distortion.

The even harmonic distortion can be inserted in the filter resonance feedback path after the last 6dB filter. This has the advantage that the output of the filter can be the output of the harmonic distorter, so that both the filter and the resonance feedback can take advantage of the same distortion. This will make the filter slightly more grungy. The patch looks like this:

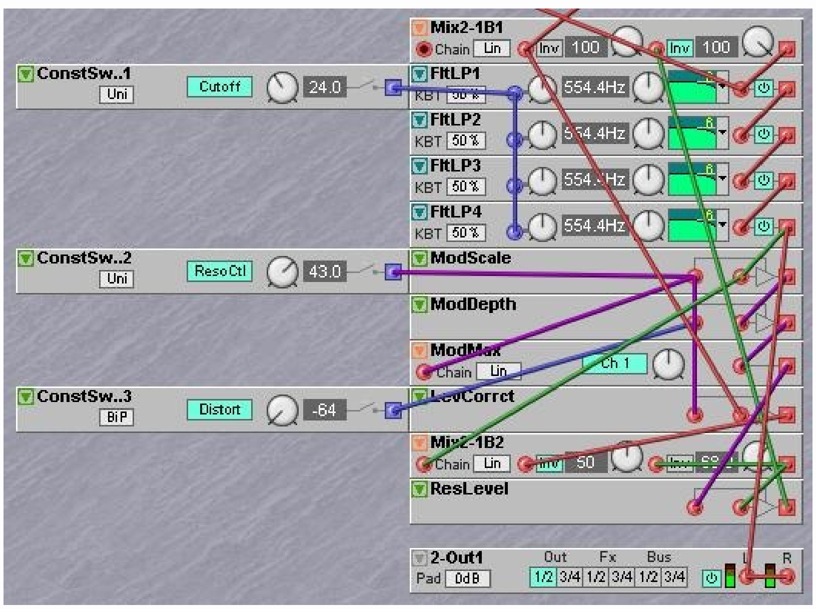


Effects of even harmonic distortion

An interesting effect of even harmonic distortion is that it can make a filter appear to be a little bit steeper. Imagine that a single sawtooth wave is filtered with a filter set equal to the pitch of the sawtooth wave. The second harmonic will be suppressed by 24 dB, as it is one octave higher as the filter cutoff frequency. The first harmonic is basically passed through unaltered. The even harmonic distortion will produce a second harmonic signal from the first harmonic. If this second harmonic happens to be in reversed phase with the original second harmonic in the input signal, this generated second harmonic can be tuned in level in a way that it totally cancels out the bit of second harmonic that was still passed on by the filter. If the second harmonic generated by the distorting is in phase with the original second harmonic the filter will appear to sound less steep and give a slightly buzzy character to the filtered sound. So, with this even harmonic distortion one can go two ways in changing the timbre of the filter.

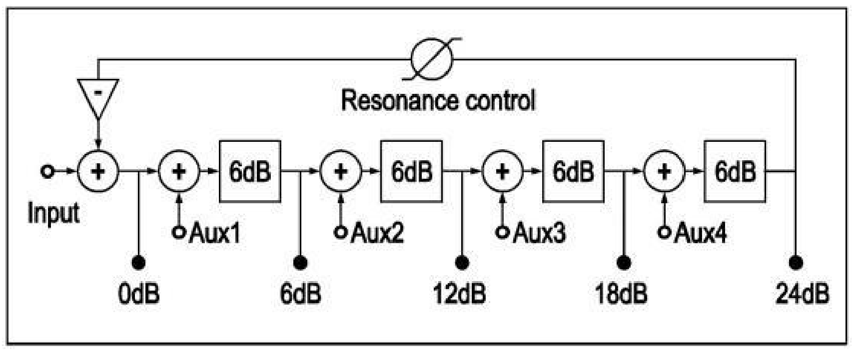
Another way to implement the distortion

Another and simpler way to get the same type of grungy filter distortion is to make use of the gain controller in the resonance feedback path to create the asymmetric distortion. By adding the audio output signal of the last pole in the filter to the control signal of the resonance level gain controller the same type of distortion is produced. To accomplish this the filter output signal must go through two additional gain controllers, one that sets the depth and polarity of the resonance modulation signal and the second one to scale the modulation signal to the resonance depth. This scaling must be done to prevent the modulation signal to prevent the resonance modulation signal to have an effect when the resonance is low or zero. After the two gain controllers the modulation signal is added to the resonance control signal in a simple two input mixer. The green cables show the resonance feedback loop signal flow, while the purple cables show the resonance modulation signal flow.



More filter inputs and outputs

What few people realize is that a basic 24dB filter actually has several inputs and outputs. The outputs are obvious; they are the direct outputs of the individual 6dB filters. Additionally there is a fifth output, which is at the output of the first mixer where the feedback signal is mixed back into the loop and just before the first pole. The inputs are less obvious, but you must realize that to get a signal into the basic filter a mixer had to be used before the first 6dB filter. By adding additional mixers right before the 6dB filters and after the outputs of the previous filters, four extra inputs are created. This works, as when resonating, the filter is basically a closed loop, and a signal can be inserted at any point you like. So, a 24dB four-pole filter with all the possible insertion and output points looks like in the following schematic:



The pole outputs

As you can see the outputs are named 0dB, 6dB, 12dB, 18dB and 24dB. The trick to get a filter with a different passband characteristic, like a bandpass or a highpass filter, is to combine two or more of the outputs, using a mixer module to add or subtract (=add in antiphase) certain amounts of the output signals. There is one thing of importance to realize, the 24dB lowpass is the steepest slope possible, any mix of output signals will always create a lowpass slope that is less steep. E.g. a bandpass filter can have a whole range of possible slopes, but the HP slope of the bandpass response will eat away steepness from the LP slope, e.g. it is possible to have a bandpass filter with both 12dB HP and LP slopes, as that adds to a total of 24dB, but it is not possible to make a bandpass response with both 24dB HP and LP slopes. In this last case four more poles should have to be added to add to a total of 48dB.

Peak filter

The 0dB output always includes the clean input signal on its output. If the resonance is set to zero the 0dB output will simply pass the input signal unaltered as no signal comes back through the resonance feedback loop. But if the resonance is raised, a strong resonance peak will appear in the output signal, added to the input signal by the resonance feedback loop. This makes this type of filter similar to a sharp peak EQ filter with a modulatable peak frequency. This is a bit similar to a Wah filter with a very pronounced resonance. This 'peak' filter can be used musically to boost one single frequency in an audio signal, e.g. to create a superloud basskick in a sampled drumloop. Or a sharp whistling in a noisey wind sound effect.

Highpass filter

To make a highpass filter an extra mixer module is needed, as one has to subtract the 24dB output from the 0dB output. Note that this type of mixing is a bit similar to simple primary school adding and subtracting, but instead of numbers it is frequency bands that get added or subtracted. The 0dB output passes the whole frequency range, so when the lowpass frequeny band is subtracted from the whole range on this output, it is the high frequency band that remains. In fact, the output of the extra mixer will regain what the lowpass output threw away, plus the resonance peak. The level on the HP output appears to be much louder as the LP output. The reason is simply that higher frequencies have more sonic energy. This can be solved by lowering the level of the input signal by some -6dB.

Bandpass filter

Subtracting the 12dB output from the 24dB output makes a bandpass filter with two 12dB slopes. The BP output will appear to have a relatively low level compared to the other filter outputs. This can be corrected somewhat by tuning the third and fourth pole about an octave lower. This will widen the bandpass bandwidth a little and lower the resonance frequency by about a fifth note.

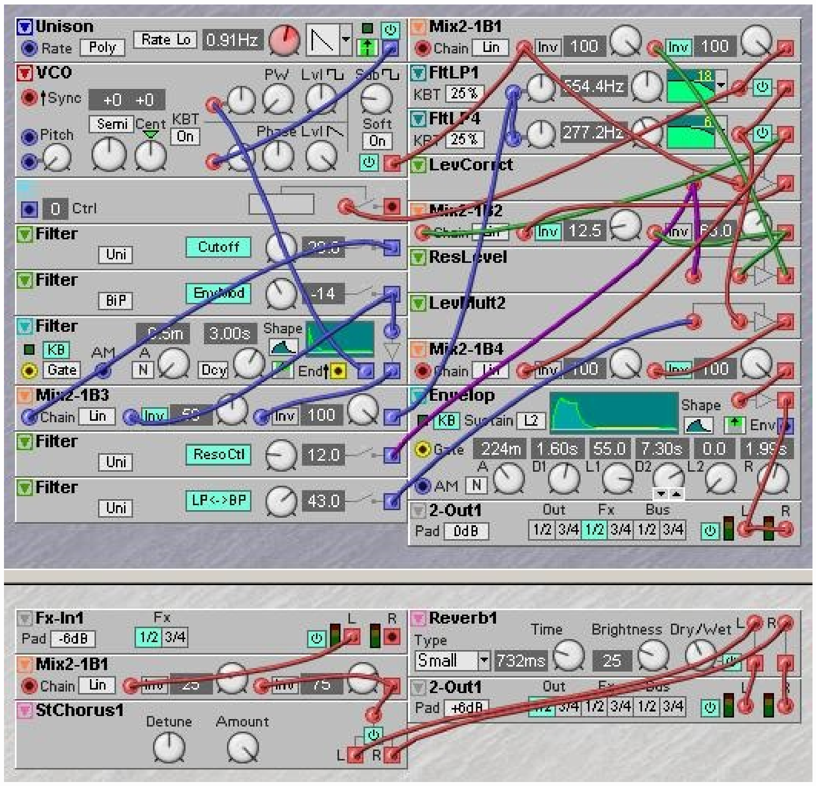
By adding about 2/3 of the clean filter input signal to the BP output the filter will become a notch filter. The notch filter will appear to have a loud output signal, similar to that of the HP output.

Asymmetric 6dB/18dB bandpass filter

An interesting alternative for the 12dB/12dB bandpass response is to subtract the output signal from the third pole from the LP output. This yields a bandpass response with a 6dB highpass slope and a 18dB lowpass slope. This bandfilter tends to sound more pronounced, the reason is that the low roll off slope doesn’t have to be that steep for the ear to give a definite highpass effect, while the steepness of the high roll off slope is much more significant to the ear. The output level however seems rather low and the fourth pole can best be tuned one octave lower to gain some more level.

Morphing the outputs for more responses

By patching a five input mixer with invert options on each channel, manual ‘morphs’ between an unlimited amounts of curves could be made, but as all curves except the 24dB LP curve will be ‘less than 24dB’ the practical effect for much settings is minimal. It is better to just pick four options and use a four-position switch to select one of these options. However, there is one morph that does work out very well and this is the morph between a 24dB LP and a 12dB/12dB BP curve. This morph is achieved by gradually mixing the inverted 12dB to the 24dB output while the 24dB output stays at full level. This LP->BP morph can give a nice buzzy sound to the filter, depending a bit on the audio material that is being filtered. Alternatively, you can use the 6dB/18dB bandpass filter and subtract a controllable amount of the 6dB output from the 24dB output to create a morph for the filter curve. The following patch is an example of this 6dB/18dB filter, used here to create a classic bright string ensemble sound.



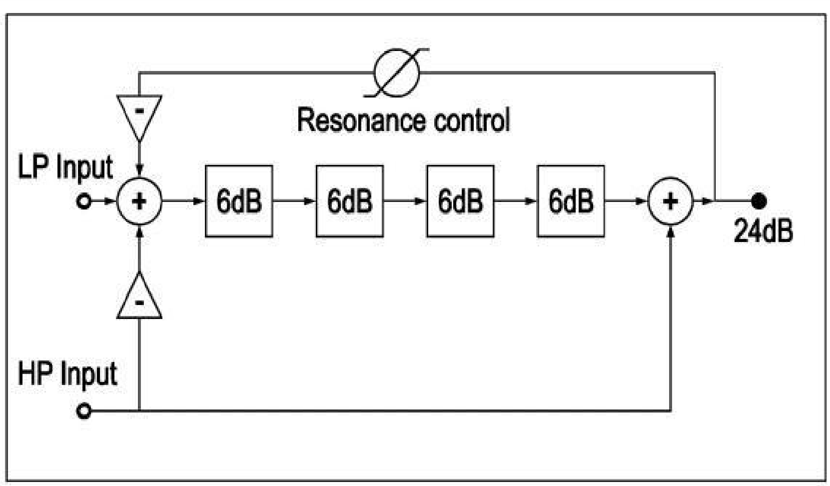
By detuning the 6dB curve you can set the width or aperture of the bandpass response to your own taste. Making the filter wider will decrease the resonance range slightly but also give less wobbly resonant filter sweep. It is best to set the bandwidth by ear to give just the effect that works out best on the filtered audio material. In the following example patch a four-position switch is used to select either a peak filter with clean feed if resonance is zero, a 24dB HP filter, a morphable BP/LP filter and a notch filter. The poles are equally detuned over two octaves to temper the resonance and widen the BP a bit. The detuning of the four 6dB filters causes the filter to not track the keyboard exactly anymore, which adds slightly to the analog feel.

Multiple input filters

The Aux insertion points in the basic filter schematic open up the possibility to mix signals in unusual ways, like mixing the HP part of one signal and the LP part of another signal and crossfade from one to the other through the audio spectrum. Just like different filter curves can be made by mixing two or more output points, different inputs can be made by routing one or more input signals to two or more insertion points.

Two input LP/HP spectral crossfader filter

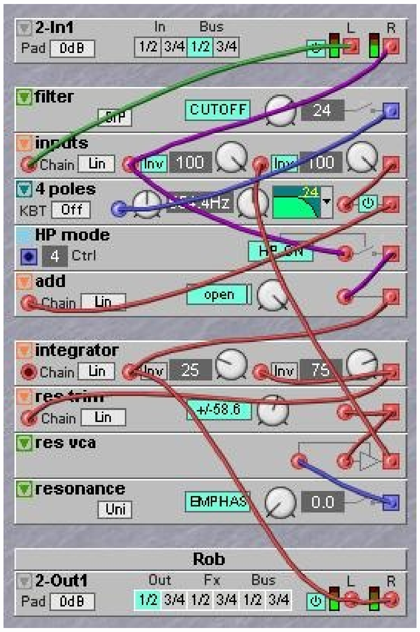
A filter that has both a lowpass and a highpass input can be used to crossfade between the signals on the two inputs. Sweeping the cutoff frequency from very low to very high over the whole frequency range will fade one signal in from the low end, while the other is pushed away into the high end. If the cutoff is set to the highest maximum cutoff frequency of the filter virtually all of the signal on the lowpass input will be present on the output, but the signal on the highpass input will be almost fully suppressed. If the cutoff is set to the lowest possible cutoff frequency the signal on the lowpass input will be fully suppressed and the signal on the highpass input will be available on the output. When the cutoff is set to e.g. 800 Hz the signal on the output below 800 Hz will come from the lowpass input and the signal above 800 Hz will come from the highpass input. An interesting use of this sort of a filter is to e.g. replace the kick and bass in a drumloop sample by a kick and a bass from another drumloop sample. And all with only one filter. Another fine use is when two different waveforms from two detuned oscillators are fed into the two inputs. The filter envelope control signal will now also crossfade between the two waveforms through the sound spectrum. In this case it is often best to use a waveform with little harmonics, like a triangle wave or the output signal from a modulated FM oscillator, on the highpass input, and a bright waveform like a saw on the lowpass input. The block schematic of such a filter looks like the following diagram:



As you can see this filter is not much more complex as the standard 24dB lowpass filter.

Spectral crossfader patch

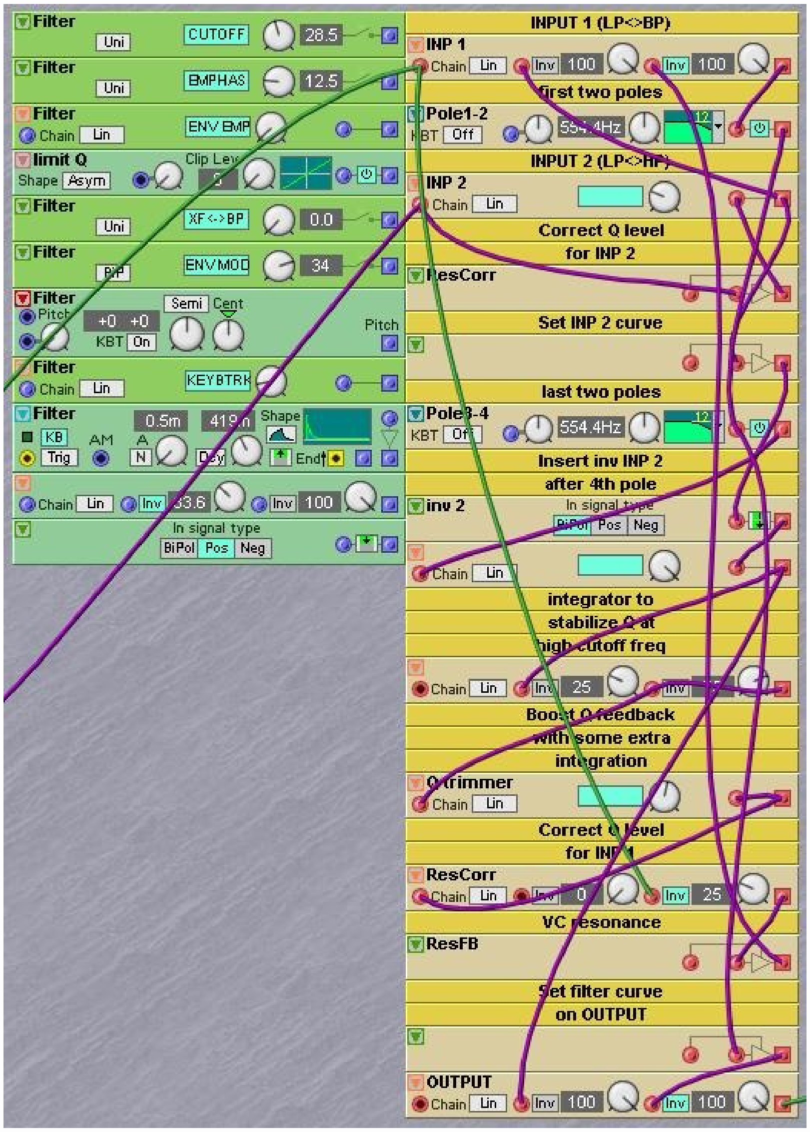
On the G2 this filter is quite easy to patch. A basic patch looks like this:



The green cable goes to the lowpass input, just like on the standard 24 dB filter that was described earlier. Extra is the purple signal that has two connection points in the patch, one goes to the input mixer where it is inverted and the other connection is made with an extra mixer right after the four poles and before the resonance feedback loop. The output of this mixer is also the output of the filter. There is an extra toggle button module named HP mode. This toggle button can switch the signal to the second insertion point on or off. If this switch is off the HP input becomes a LP input as its signal will only go to the input mixer. But when this switch is on the signal will also go to the second insertion point and the input will work like the HP input.

More options for the spectral crossfader

This two input filter can be be expanded with psycho-acoustic signal level correction when the resonance is raised. Additionally several output points can be mixed to change the filter curves of the two outputs. In the next patch example the signal levels are corrected for higher resonance settings plus a single control is added to change the curve of both the lowpass and the highpass inputs into two equal bandpass curves.



By using two additional panner modules on the input signals, where one panner fades in the opposite direction of the other one, two input signals can be 'inversely' crossfaded between either the highpass or the lowpass input. This creates a very interesting fader for two streams of audio material, where the two audio signals can be crossfaded in the traditional way or crossfaded through the audio spectrum. In a keyboard patch the filter can be used to mix two oscillators and thus create an interesting morph between two waveforms through the spectrum. In this last case it works best if the oscillator waveform on the highpass input has relatively little very high harmonics, like a triangle wave or a waveform created with a FM oscillator.

Noise, randomness and chaos

Noise

By definition noise is an audio signal that consists of an accumulation of sinewaves of all the possible frequencies in the hearing range and with all possible amplitudes and phase relations. Musically a noise signal can be seen as the opposite of a sinewave signal, as a sinewave signal contains only one single frequency component while noise contains every possible sinewave component. When used as a sound source for subtractive synthesis, noise has some interesting properties. Consider the thought that by filtering noise every possible sound might be created, as that sound should be hidden somewhere in the noise. This idea can easily grab the imagination and in the fifties of the last century composers started to experiment with audio processing of noise signals, trying to destill specific sounds from the noise signal. It soon became clear that there are two reasons why it is virtually impossible to filter every possible sound from noise. The first is that this would require filters with a quality that simply does not yet exist. The second and more important reason is that when the frequencies are filtered out correctly the amplitudes still vary wildly, making it virtually impossible to create steady tones. Still, filtering noise does open a whole range of sounds that often have a spatial, almost eerie nature, just because of those wildly varying amplitudes. Much of the electronic music of the late fifties and early sixties is characterized by sounds made by processed noise signals, e.g. by tuning a couple of sharp bandfilters to chords and use these bandfilters in parallel to filter the noise. Noise is still a very important sound source for creating sound effects and to add 'breath' to sounds that mimic wind instruments or enharmonic sounds like cymbals and percussion.

A typical example of how noise can be generated is by amplifying the thermal noise in the semiconductors used in an analog circuit. Thermal noise is inherently present in every analog electronic device, e.g. the faint noise that is heard when fully opening the volume control of an amplifier while there is no sound at the amplifier input. In this type of noise each partial appears and disappears in no apparent order. Many noise generators in analog synthesizers use a semiconductor device like a diode as the part that produces the noise. By driving a small current through the diode, thermal quantum effects appear at the junction of the diode device, creating a very faint noise signal which is then heavily amplified to bring it to line level. This type of analog noise sounds very lively as the amplitudes and phase relations of each partial change all the time.

Noise has no apparent pitch. Now, pitched sounds can be described exactly by the momentary frequency and harmonic content. But unpitched sounds don't let themselves be described that easily, just as there would be too much partials to describe. Instead, noise signals are descibed by either how much sonic power is in certain frequency ranges or by the statistic probabilities for certain frequencies to appear. The simplest case is when each possible frequency has an equal chance of occurrence, which in statistical terms is named an equal distribution. When a noise signal has an equal distribution, it sounds like a very bright hissing sound where the higher parts of the audio spectrum are clearly much more pronounced as the lower parts. The timbre of noise is often reffered to by a colour and equal distribution noise is commonly named white noise. Still, there is no real physical relation to white light, the association with a colour is just to give a name to the timbral character of the noise signal. The reason why white noise sounds very bright is because the equal distribution of possible frequencies is on a linear frequency scale, but the ear hears pitches on an exponential scale. Remember that each higher octave means a doubling of the frequency values, so transposing a pitch by three octaves means in this case an increase of the frequency by 2^3 or making the frequency eight times higher when expressed in Hertz or cycles per second. What this implies is that in white noise the sonic power in each octave band in the hearing range also doubles for each higher octave band, just as there are twice as much frequencies in that higher octave. So, there will be much, much more high frequencies in the highest perceivable octave band that in the lowest perceivable octave band, which accounts for the bright hissing timbre. When noise is processed by filters to create specific sounds it is often more useful to reduce the sonic energy in each higher octave bands to prevent higher pitched sounds to be overly loud compared to the lower pitched sounds. This means that when the frequency doubles, the possible range of amplitudes should become smaller. Basically this can be done by filtering white noise with a lowpass filter set to the lowest audible frequency and amplifying the filtered signal until it is at line level again. When using a 6dB lowpass filter he timbre of this type of noise is named red noise, as it sounds quite dark. Like white noise sounds overly bright, red noise sounds overly dull. To the ear, the noise timbre that is perceived as neutral is somewhere in between white noise and red noise and is named pink noise. Pink noise sounds very pleasant, it appears to have a timbre like the sound of a distant ocean surf. When analyzing the spectrum of pink noise it turns out that the sonic power decreases by 3dB in every next higher octave. Statistically this means that a frequency has a probability of 1/f. The interesting thing is that there are many natural phenomena that occur with a statistic probabililty that also exhibits this 1/f or -3dB per octave curve. Not only our ears, but nature itself seems to like pink noise rather well.

Pink noise can theoretically be made by filtering white noise with a lowpass filter with a 3dB cutoff slope where the cutoff frequency is set to the lowest perceivable frequency, so the cutoff slope is a straight -3dB declining line over the whole frequency range. Technically it is not at all that easy to make a filter with a perfect 3dB cutoff slope, as there is no one singular filterblock known that exhibits such a slope. In practice it turns out to be very complex to generate a perfect pink noise signal. Which means that in general a method is used to approximate as closely as possible the -3dB curve, where perfection is traded off with complexity. Musically the easiest way to make good sounding 'pinkish' noise is to use two parallel allpass filters tuned to around 200 Hz and 2kHz, where a certain mix of the two filter output signals with the original signal roughly estimates the timbre of pink noise.

There are more types of noise that received the name of a colour to quickly grasp their basic tmbre. Blue noise has a slope of +3dB, in contrast to pink noise the higher frequencies in blue noise are amplified instead of attenuated. Blue noise appears to the ear to have no low frequencies at all. 'Brown' noise is a very dark sounding type of noise and is actually derived from what is named Brownean motion, a signal that is made by randomly adding or subtracting a small fixed value to the previous value. This is also referred to as a random walk signal, as each new value is 'the length of one step' away from the previous value. Brownean noise is often used to generate slowly varying control voltages.

The spectra of white, pink, red, blue and Brownean noise all have a smooth slope that is either horizontal or slightly tilted. Next to these common types of noise signals there are some more specialized types of noise. These special types are characterized by spectra that in fact contain a number of well defined frequencies, but these are tuned in a way that the human brain can not determine a dominant pitch. A good example is a type of noise that is named bronze noise or metal noise. This type of noise contains strong partials that sound very much like cymbals. This noise is created by mixing a couple of oscillators with harmonic rich waveforms. The oscillators are tuned in a way that the common fundamental of their mix lies below 16 Hz, so the mind cannot perceive this fundamental anymore. As the ear cannot hear pitches that are below about 20Hz, the mind has no reference to imagine what such a pitch would sound like, similar to how the mind cannot imagine what infrared or ultraviolet light looks like. It is basically the harmonics of the oscillator waveforms that create a sound with a pitchless sensation, but with a distinctly different timbre from white noise or pink noise. A good example is the metal noise used for the cymbal sounds in the Roland TR808 analog drumcomputer. In this drumcomputer the noise is generated by a mix of six squarewave oscillators tuned to roughly 210 Hz, 320 Hz, 380 Hz, 540 Hz, 550 Hz and 800 Hz. The lowest common denominator of these frequencies is 10 Hz, which means that the fundamental pitch of the mix is at 10 Hz. The pitchless sensation is enhanced by highpass filtering with a 12dB highpass filter, a cutoff at around 1600 Hz seems a good value.

Curiously enough it is not easy to produce good sounding noise with digital techniques. The simplest way to create digital noise with discrete digital chips is with a circuit based on a shift register. A shift register is sort of a pipeline that can hold a sequence of bits. New bits can be shifted in at the left side of the pipeline causing the bits that are already in the pipeline to shift their positions one location to the right by command of a clock pulse. The status of the new bit that is shifted in at the left can depend on a rule of what to do when two or more bits in the shift register form a certain combination. By combining those bits with an exclusive OR function, a rule that produces a zero value if both bits are equal and a one value if the bits are unequal, a sequence of numbers can be generated that appears to be random. All the bits in the pipeline together form a binary number that is used as the output. As the number of bits defines the possible amount of numbers that can be represented, e.g. eight bits can produce 2^8-1 = 255 numbers, the sequence will eventually repeat itself exactly. Real randomness never repeats, so this shift register algorithm is aptly named a pseudorandom number generator and in fact it produces a periodic signal. But when the pipeline uses 48 bits that are clocked at a samplerate of 96kHz, one period lasts somewhat over 92 years, so only few of us will live to hear it repeat. The disadvantage of the shift register method is that the random signal seems to have little motifs, short arpeggio-like figures caused by the delay in the shift register before the bit that shifts into the register will actually influence the feedback signal. When a system has a multiplication possibility another algorithm named the linear congruential method can be used. This method gives slightly superior results over the shift register method, as it lacks the short arpeggio-like figures. The linear congruential method uses the formula x' = (a * x - b) mod c, where a and b are well-chosen prime numbers and c defines the range.

The advantage of pseudorandom number generators is that they can be preset to a certain number defining the startposition in the sequence. From there on they will proceed in a well defined way, but so complex that the sequence will appear to be random to out ears. Another advantage is that every possible number will appear only once during one period of the sequence with the parameters for the algorithm are properly chosen. This means that the numbers have an equal distribution and the timbre of the sound produced by the algorithm is equal to pure white noise. An algorithm that produces pink noise is much more complex to implement, basically because noise is not about generating random numbers but generating an unlimited amount of frequencies that all have a chance of appearing depending on their frequency. As mentioned before the chance of a frequency f to appear is 1/f for pink noise. Most algorithms to generate digital pink noise algorithm are very complex, too complex to be easily implemented in a current state of the art digital synthesizer without eating away all the computational resources. The difficulty is in the fact that it is not the statistical distribution of numbers that defines the timbre, but the statistical distribution of frequency partials. Just like with analog pink noise it is best to approximate digital pink noise by the same methods, e.g. by applying the earlier mentioned dual allpass filtering method on a properly chosen pseudorandom number generation algorithm pink noise can be created that is of sufficient quality to be used for musical purposes.

To filter specific sounds from noise the basic colour of the noise used as the sound source is very important for the final effect. In practice it turns out to be very difficult to judge the timbral quality of the noise source by ear. In everyday life there are noises all around us and many of these noises are unpitched. Examples are natural sounds like the whistling and hissing of the wind or the flow of a stream of water. Or man made sounds like the noise of cars running down the road. These sounds have specific characteristics by which they can be instantly recognized. The human mind has a tendency to suppress these sounds from our awareness after a while and somehow listening to the sound generated by a noise generator seems to slightly confuse the brain. The effect is such that when steady noise is heard it is difficult to remember how noise of another colour actually sounds like. This is a psychoacoustic phenomenon that might have to do with how the brain tries to suppress background noises, with how in normal life a continuous noise sound, like e.g. the ocean surf heard from a medium distance, would quickly be suppressed in the human awareness. To judge the sonic qualities of noise there is a simple trick that involves a vocoder. When a noise signal is fed into both inputs of a vocoder the noise sound will be transformed into a sound that is immediately associated with rain or a running stream of water. Pink noise and a good quality vocoder will give a very convincing sound of a running stream, in fact the more convincing the better the quality of the vocoder. While the vocoder produces the character of running water or falling drops of water, the colour of the noise will define the association with a certain natural phenomenon. White, pink and blue noise might sound like rain, a fast running stream from a short distance, a slow running stream from a long distance, a quietly babbling stream, a high pressure jet of water, etc. Red and Brownean noise will sound more like thunder from a distance or an earthquake. Just as the association is quite clear it makes it a lot easier to judge if the noise is suited for the particular use in mind. So, the vocoder will reveal specific qualities that are hiding in the unprocessed noise and the sound from the vocoder can give some guidelines on what to expect when doing other filterings. It is also a good trick to judge the quality of a vocoder.

Randomness

Randomness is closely related to noise. Noise is an audio signal, but randomness has many times to do with musical events, like when certain notes are being played or how a control signal doing a modulation develops. Basically randomness is about making choices, e.g. a musical event like a note is about to happen and for this upcoming event a choice for a specific parameter must be made out of a limited or unlimited amount of possible choices. Note that possibilities can always be clearly defined by a set of rules. This is a bit the difference between noise and randomness, noise is audio with a certain static timbre that can be processed, but randomness involves making active choices from a certain set of alternatives. Next to random choices, where the choices are made on discrete moments in time, there is the possibility of a continuous control signal that develops in an uncertain way. With such a continuous random signal the choice on how to proceed the development is made continuously. There is some subtlety involved in the definitions here, some might say that they don't make choices as they simply use a module that produces something random and that's it. But randomness always implies the possibility of choices according to rules and choosing not to make choices is in itself already a choice that is part of the possible choices. It is just the simplest choice one can make.

On an analog synthesizer there can be an output on a noise generator named random voltage. The signal that comes from this output is a slowly varying control signal and is in fact a subsonic noise signal. In general this noise signal is lowpass filtered from white noise or pink noise in a way that there are no more frequencies in the audio range present, but all frequencies that are present are actually in the subaudio range. It is comparable to the signal from a low frequency or modulation signal oscillator, but instead of producing a well defined periodic waveform the random voltage signal just seems to wander aimlessly between a minimum and a maximum limit. There is some sense of speed in the signal, it can slowly change in a relaxed rate or variate wildly at a fast rate. A musically important property for a random control signal is how the apparent rate or the slope of the signal accelerates and decelerates. The reason is that acceleration and deceleration of tempi can be important means to give music a human feel. Acceleration and deceleration can also give modulations a sense of going somewhere, which might increase tension in an abstract electronic music composition. When a continuous random control signal is created by filtering noise with a lowpass filter there is no specific acceleration and deceleration as a parameter to be controlled. It is fully up to chance how the signal will develop. Brownean noise is an exception, as with this type of noise the momentary signal depends on the choice of how the previous state changes by either adding or subtracting a specific number. The value of this number can be manipulated depending on certain rules and this opens up the possibility of controlling acceleration and deceleration by a parameter. This is very complex to do with analog electronics, but a computer can easily run an algorithm where a controllable acceleration parameter is implemented.

When random choices have to be made on specific moments, like varying the initial phase relation of two unisono oscillators on a keypress to give each played note a slightly different character, a random value has to be picked from somewhere. The common way to do this is to use a S&H; (Sample & Hold) module that can measure the momentary value of an input signal on a clock pulse command. The value is stored internally and presented on the output as a fixed value that stays fixed until the next clock pulse command is given. A S&H; module is best seen as a memory cell that can measure and memorize an analog value. But next to being a memory cell the S&H; is also an important synchronizer module, as every store operation is synchronized to the clock signal that commands the S&H; to memorize and store the input value. When a S&H; receives a series of clock pulses at a certain rate there will be statistic properties in the sequence of output values. And these statistics depend fully on statistic properties of the input signal. When using a noise signal to be sampled the output values will be a series of unpredictable random numbers. Although there is a big difference in sound between white and pink noise, there will not be such an apparent difference in the generated series of numbers when sampling either white noise or pink noise with a S&H.; Much more apparent is differences in how the amplitude of the input signal deviates around zero. When using a S&H; the rule of thumb is that the distribution of momentary amplitude values is of more importance than the distribution of possible frequencies in the signal to be sampled. There is some sense to this, a S&H; samples amplitude values and not frequency values.

To get a predictable behaviour it is good to start with a signal where it is known that every possible amplitude value has an equal chance of appearing. The shift register pseudorandom number generator is a perfect choice. The maximum length of the sequence it can produce is 2^n-1, where n is the number of locations in the shift register. The produced value will be in the range from 0 up to and including 2^n-1. E.g. when the shift register is seven locations long the sequence will be 127 steps, and each integer value in the range from 0 up to and including 126 will appear once. To get this range the shift register is fed back with a XNOR function combining taps six and seven from the shift register. There is one combination of bits that can never appear as it would stop the production of new values, this state is if either all bits are zero when a XOR function is used or if all bits are one when a XNOR function is used. Initially the bits will be all zero, so when using a XNOR function one never has to worry about this issue as the number where all bits are one is simply never produced. Let's assume that there is a pseudorandom number available that will do this 127 steps and produce 127 values. This would be quite convenient to play notes when each value represenst a note. Or to produce values that will be used for velocity or a midi CC# to be send to some device, just as midi CC#'s can only handle 127 values. But the pattern that is generated repeats every 127 steps and so doesn't really appear random. Still there should be 127 different sequences possible and when it is possible to sequence through these sequences the total length would become 127 times 127 is 16384 steps before the sequence would repeat. The way to do that is to scramble the order of the basic pseudorandom sequence as generated by the shift register. There are several ways to go about increasing the sequence length, the most obvious is of course to increase the length of the shift register. But as on an analog modular system the shift register is often made by cascading a number of S&H; modules, and there might be just a limited amount of S&H; modules in the system available, it is interesting to look at other options. One option is to make use of the principle of interference. The idea is that the output of the shift register is sampled with an extra S&H; that runs at another clockrate. The frequency ratio between the clock used on the shift register and the clock used on the extra S&H; will define how the sequence gets scrambled into a new sequence. Basically the original pattern and the output of the extra S&H; form an interference pattern. In normal situations one would want the extra S&H; to be clocked by the masterclock that syncs everything in the patch and variate the clock that clocks the shift register. If the shift register clock is faster than the extra S&H; clock there will be a differnt value on every S&H; clock pulse, but if the shift register clock is slower the values will hold for one or more clock pulses. The extra S&H; doesn't need to be clocked by a continuous train of clock pulses like those coming from a tempo masterclock, the clockpulses can also come from e.g. the keyboard gate. This will produce random value on each keypress. because the relation between the moment of the keypress and the momentary shift register value is pretty random the pseudorandomness of the shift register is changed in a real random value, but with the statistical property that each possible value has equal chance to appear.

Sometimes it is wanted to change the statistics of the equal distribution, meaning that the chance for a certain number to appear must be greater than another number. An example is when one wants a sequence of only the notes E, F, G and Bes, but want the statistics to be that the E and G have three times more chance than the G and Bes. In such cases the easiest way is to use a lookup table. If this lookup table has eight locations to store values and three locations are filled with a value that will produce an E note and three locations are filled with a G note value and the resting two locations are filled with the value for the F and the Bes, it suffices to use an equally distributed random number to choose a location in the table to get the right statistics from the table. A synthesizer module that is able to work as such a table is a voltage controlled sequencer. Such a sequencer is not stepped to a next step by a clockpulse, but a control voltage input makes it switch to a certain step. The knob that belongs to a certain step sets the lookup value and the voltage level on the control voltage input will select the value set by the corresponding knob. This type of sequencer usually has eight or sixteen steps on an analog system. On a digital system there might be much more steps available. When using a programming language on a computer a lookup table, or array as it is named in many computer languages, might have many thousands of locations to store lookup values. Lookup tables are a very convenient way to change statistics of a range of values and often works better than trying to figure out some mathematical formula and trying to patch such a formula with mixers and multiplier modules. The output of a lookup table can be used to lookup a value in another lookup table to define complex rules. A use might be to define possible chord progressions. The output of a table can also be used to lookup a value in the same table again, which is in essense equal to a technique named cellular automata.

Chaos

There are many dynamic processes or systems in nature where it can be verified that every current state develops from a previous state and an initial state defines how the whole process will develop. A wellknown example is named 'The Butterfly Effect', or how the movement of the wings of a butterfly in the Amazon Rainforests could start a chain of events that eventually could cause a storm to happen in Oklahoma. In the last thirty years there has been a lot of research on such systems and this research has shown that many of these systems can have several stable states. When in such a state the system is in balance until some influence gets it out of balance and it develops into another stable state until it is disturbed again. These kind of systems are known as chaotic systems, there is definitely a certain order in the system, but the order is many times so complicated that it is simply impossible for a human to grasp how it develops and so it is designated as chaos. Still, the stable states might be well recognized. Chaos generators are of musical interest because they can produce sonic source material that is quite different from the sounds produced by oscillators or noise generators. Don Buchla pioneered the field of chaos generators by designing the Module 265 'Source of Uncertainty' for the Buchla Music Box analog modular system. A more recent chaos generator module loosely based on the Buchla design is the 'Wogglebug' made by Wiard. These modules produce chaotic random voltages and randomly gliding tones. An analog circuit that is truly chaotic is known under the name of Chua's circuit, developed by professor Leon Chua. Chua's circuit is an example of a simple non-linear feedback system where the nonlinearity in the feedback path will create chaotic behaviour. A cimilar circuit has become known as the Cracklebox, developed by Michel Waisvisz and marketed as a little wooden box with a few touchpads. When placing the fingers on the touchpads the box will start to make chaotic crackling noises that to some extend can be influenced by the fingers.

A chaos generator will have attractors that reveal themselves as a short repetitious pattern or sequence. When the generator produces such a repeating pattern it is in a stable state. Such a repeating pattern forms one cycle of a more or less randomly shaped waveform. Only a small variation in a controlling parameter will disturb the stable state and the generator will produce a series of apparently random values until at a certain moment it will get caught in another repeating pattern. It gets literally attracted to that new pattern, hence the name attractor. So, basically the attractor is the pattern the chaos generator will eventually adopt and not a parameter to be tweaked. Byt the tweakable parameters will define to which attarctor the pattern will evolve to.

Building a chaos generator

A Sample and Hold module is at the core of a chaos generator. The output of the S&H; is processed by some modules that must exhibit some non-linearity and the output of these processing modules is fed back into the input of the S&H.; Let's assume that the S&H; is initially filled with some value. This value is changed into another value by the processing modules and as long as the S&H; outputs this initial value the final output value of the processing modules is stable. When the S&H; receives a clock pulse it will sample this final output value and use it on the output of the S&NH; module as a new value to be processed. The processed new value is sampled again, and on every sample clock to the S&H; the value on the output of the S&H; will change. If the processing modules together form a function that is by nature chaotic, a repeating pattern will eventually be produced, the pattern actually depending on the initial value in the S&H.; There are quite a few simple mathematical fucntions that can be easily patched and have the non-linearity that will create chaotic behaviour. The simplest and most well known is the function X' = 4 * X * (1-X), where x is the current value in the S&H; and X' is the result of the calculation that will be sampled in the S&H; on the next clock pulse. The initial value must be between 0 and 1 and the output will always be between 0 and 1 as well, so it is fitted to be the new input again. The whole trick of a chaos generator is to insert the initial value or seed value. To do this a controllable two-pole switch can be used that switches the input of the S&H; between the output of the modules that form the function and a constant value or knob that defines the seed value. The switch must point to the constant for exactly one clockpulse only and a special circuit named a one-and-only-one can be used to generate the single clockpulse. Instead of a controllable two-pole switch a voltage controlled crossfader can be used, but a one-and-only-one module is most probably not present on an analog modular system. Digital modular systems like the Clavia G2 do have all the modules on board to create chaos generators based on a non-linear function.

The attractor or stable pattern a chaos generator will eventually adopt is defined by both the seed value and the non-linear function. Instead of inserting a new seed value on a clock pulse command the non-function can be slightly modified. When the function is modified the pattern will evolve over a short time into another pattern. The easiest way to change the function is to reduce the feedback a little. It shouldn't be reduced too much or the generator might stop to produce new values. The properties of the function should be that the generated patterns should be sufficiently long to be of musical interest, a couple of hundred to a couple of thousand values is convenient. Additionally it should take some time to evolve into a new pattern, again some hundred to a few thousand values is of interest. When the chaos generator is run at audio rates these sequence lengths can produce very characteristic sounds. When it is run at lower rates to create melodic patterns one might go for functions that produce shorter lengths. A function that produces patterns and attractor transitions of sufficient length and additionally produces bipolar values between -1 and +1 is the third Chebyshev polynomial X' = 4 * X^3 - 3 * X. This function is quite easy to program, but if it produces the value 0 it will hang as an input value of zero will produce an output value that is also zero. A zero value is however quite easy to detect and a good moment to automatically insert a new seed.

Another method to produce chaotic sounds is to feed the output of a squarewave oscillator into a lowpass filter and feed the output of the lowpass filter back into a LinFM modulation input of the oscillator. If a LinFM input is not available a pitch control input can be used as well. Without filtering the feedback loop the oscillator would switch between a very fast and a very low frequency, which would cause the oscillator to produce a narrow pulse on its output. The filtering slows this process down in a way that the oscillator can come into a chaotic state. Tweaking the filter cutoff and resonance and the modulation index will produce sounds that are in between the original square wave through a range of semi-random pulsations to a noise signal.

Phase modulation oscillators are also very good to create chaotic patterns, especially if the pitch can be set to zero Hertz, converting the oscillator into a sine function. In this last case a S&H; module is placed between the PM oscillator output and the PM input. If the oscillator is set to a pitch of zero Hz the oscillator changes into a sinewave function. Each clock pulse on the S&H; clock input will put a fixed value on the PM input and the output will be a value that is the sine of that PM input value. As the oscillator is actually stopped by setting it to zero Hz, the output value of the oscillator wil be fixed until the next clock pulse. This patch could produce output values of zero, which would hang the process. To avoid this a fixed value must be added to the oscillator output before it enters the S&H.; The chaotic pattern can be disturbed by small changes in the fixed value. As long as this fixed value is not zero the patch will produce chaotic patterns with attractors and transition periods between two attractors when the feedback is disturbed.

The output of the chaos generator is a stepped signal, but it can be changed into a linear gliding signal by adding a few modules. The idea is that when the S&H; is clocked by the flank of a sawtooth waveform and the output of the chaos generator is fed into a shift register, that is also clocked by the sawtooth flank, a modulatable crossfader can be used to create linear glides between two adjacent outputs of the shift register. The sawtooth signal is used to control the crossfader position. It works like this; when a flank in the sawtooth triggers the S&H;, and so a new value is generated, the output values will shift one position to the right in the shift register. The crossfader will on the flank of the sawtooth immediately crossfade to the previous output value that is now one position to the right, and then crossfade to the new output value that is to its left in the shift register. On the next sawtooth flank this will repeat and so the crossfader will smoothly crossfade between the previous value and the new value, creating glitchless linear glides. This signal can be used as a random glide signal that follows the chaotic pattern of the chaos generator. When e.g. an eight output shift register is used seven crossfader can be used to create seven glides, each glide being a delayed replica of the crossfader that uses the crossfader to its left. These glide signals are very useful to control and modulate all sorts of parameters in a patch. When the sawtooth signal that drives the chaos generator, the shift register and the crossfaders is synced to the tempo clock a whole range of tempo synced glides are created in what is much like a canon. Of course, the chaos generator can be replaced by a sequencer module, a S&H; sampling any waveform or a another type of clocked random signal generator.

Dynamic processing of signal levels

Introduction

An important property of the G2 is that every module can handle every type of signal, such as dynamically varying signals like audio signals, slowly changing control signals like those from low frequency oscillators and envelope generators and static signals that have a fixed value that might have been set by a panel knob. Such a fixed or static value is named a Level signal, just as the value has a certain level that remains fixed to the value it is set to. There are many signals that are static by nature. A good example of such a static signal is the note value in a monophonic patch, after a key is pressed the note value of that key will be static, until a new key is pressed. Gate signals are also static, as long as a key remains pressed the keyboard gate signal stays in the fixed mode on that is represented by a static level of +64 units. And when the key is depressed the gate goes in the fixed mode off, producing a static level of 0 units. Such a gate signal can be routed into any module and several modules can sometimes do unexpected sensible things with the gate signal. An example is when a gate signal is fed into the audio input of a filter. When the filter is set to lowpass the flanks of the gate signal will not rise sharply anymore but become smoother. And when the filter is set to highpass there will be a short click on the output of the filter every time the gate changes state. These clicks can be used as an audio signal, to produce rhythmic clicking sounds where the filter shapes the timbre of the click. But the clicks can also be used as a very fast envelope over another sound to produce short blips. The general point of interest is that static levels can be processed in many ways to serve a multitude of musical purposes.

Algebraic operations

A very interesting use of static levels is to do simple algebraic operations. A simple musical example is transposing an incoming MIDI note by one octave and retransmitting it to another MIDI device. To transpose a note all that is necessary is to add a static level to the note value. The value of the static level is the amount of transposition that will be given to the incoming notes. Adding a positive value will transpose the notes up, while adding a negative value will transpose them down, adding a static level of +12 units will transpose the note up by one octave. But there is much, much more that can be done. At the end of the fifties and first half of the sixties there was a device that looked remarkably similar to the analog modular synthesizer. This device was named the analog computer and was used to do arithmetic computations. This analog computer hasn’t been used for long, as it was very soon replaced by the pocket calculator. But the interesting thing about analog computers was that they were modular, just like a modular synthesizer, only the modules did arithmetic operations like addition, subtraction and multiplication instead of the waveform generation and waveform processing done by a sound synthesizer. Next to algebraic modules there were modules that could provide a level with a value that could be set with a knob. Patchcords were used to connect inputs and outputs, and many of the circuits used in the analog computers were also used in the first modular synthesizers. These analog computers could be used to patch a mathematical function, setting variables by the value dials and then read the result on a meter at the output. The G2 has many of the modules that were present on those analog computers available in the Level and Mixer tabs. And these modules can be used to do many things that are absolutely unavailable on other keyboard synthesizers. Addition of two values is simply a mixing function, when two values are fed into two inputs of a mixer there will be a value at the output that is simply a sum of the input values. Similarly, inverting one of the mixer inputs with the invert button next to the input does a subtraction. And the Level multiplier module is in fact a multiplication function. This means that there is also a type of mixing possible that can implement some sort of user definable function. Such a function can be used for algorithmic composing purposes, but can also be used to process audio signals. Mathematical functions can actually do musically interesting things to a sound signal. A good example is when a sine wave is fed into both inputs of a G2 Level multiplier module. When this is done, there will also be a sine wave signal at the output, but this new sine wave will have twice the frequency of the input signal. Mathematically the sine is raised to the power of two, so the sine becomes a sine^2 signal or the quadrature of the original sine.

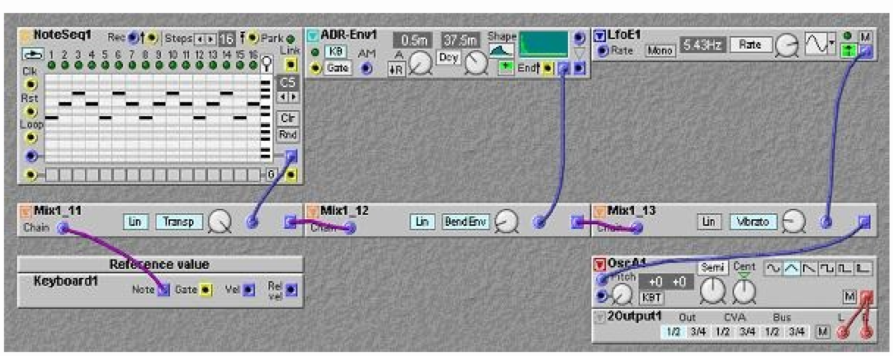


Figure 1 - Sine wave and its quadrature at double frequency

Figure 1 shows how this frequency doubling effect comes about. The straight line is the original sine wave and the dotted line is its quadrate. From the picture it becomes clear that the frequency doubling effect is caused by the fact that multiplying two negative numbers will have a positive result, so the negative half of the sine wave is flipped up to a second positive half, creating two positive peaks instead of the original one positive peak.

Doubling a frequency is of course of great musical interest, as it will produce a signal with a pitch that is exactly an octave above the pitch of the original signal. Which is in fact the second harmonic for the original sine wave. So, here is a clue to generate second harmonic distortion. Regrettably, frequency doubling only happens reliably under specific circumstances, so what works very well with a sine wave of amplitude 1 may give strange results with other waveforms of varying amplitudes. In general the patching of functions and applying them to waveforms will work very well for some waveforms and could totally mess up others. Meaning that this is not a general technique, like filtering that can be applied to any signal with predictable results. But for those instances where it works out well, there is no reason not to explore this technique. In fact all waveshaping and distortion techniques can be translated into some simpler or more complex mathematical function. Many modules doing distortions have some sort of function internally programmed into their programming code. The details of this technique need a chapter by itself, right now it is important to realize that, next to sound processing, the G2 can do computations on signals as well. As an example, by using the G2 as some sort of analog computer, incoming MIDI information can be processed in simple or more complex ways and then retransmitted with the MIDI out modules to other MIDI equipped gear. And on the audio level, when the quadrate of the output signal of a Phase Modulation oscillator is taken by feeding the output signal into both inputs of a Level multiplier module and the output of the Level multiplier is fed back into the PM modulation input, the oscillator will start to generate a signal with only odd harmonics. Such a signal sounds like the typical ‘hollow’ sound of a square wave oscillator. Increasing the amount of feedback can increase the brightness of this odd harmonics sound.

Headroom issues

There is one drawback in using the modules in the Level and Mixer tab to do algebraic computations and that is the headroom level. As the limit of the system is –256 and +256 units, and these units scale to the arithmetic numbers minus and plus four, any computation in a function that would exceed the arithmetic values –4 and +4 would be clipped and mess up the result, making the function probably useless. But most musically interesting functions do compute nicely within these limits. Whenever you would decide to explore this territory, a careful choice of functions and/or scaling is necessary. E.g. some intermediate results in a Bézier curve function would not fit, but a cubic spline function actually does fit completely within the G2 signal headroom. These two functions, which are commonly used in graphic design software to draw smooth curves, could e.g. be used to smoothly distort audio waveforms as well.

Conclusion

This subject of doing calculations is definitely an advanced subject. It is mentioned here simply because there is more to Level modules than just some convenience modules to now and then do the odd job. To summarize, the Level modules can be used for both control signal and audio signal processing and additionally for computations on steady levels. When used for computations, many times a few modules are used in a group or sub-patch, doing some function for which there is no dedicated module. In a way Level modules can be the building blocks to build your own user defined modules in the form of a sub-patch, should you ever have the need to do so. There is tremendous power behind these deceptively simple looking modules, and in time you will certainly appreciate all the extra and musical possibilities they can offer.

Waveshaping and distortion

Introduction

Audio waves can be plotted in two dimensions on paper or on the computer screen. In such a plot the horizontal axis represents time and the vertical axis represents the momentary value of the air pressure at the particular point in time that is marked right below on the horizontal axis. In an electronic system the vertical axis commonly represents a voltage level or a number. The series of values follows as close as possible how the air pressure will fluctuate when the electronic signal represented by the graph is fed to a quality speaker system. Since the introduction of harddisk recording software for computers it has become more and more common to look at waveforms like graphs, as in this software each track can be shown on screen as the long graphic track strips that in essence show how the air vibrates when that track is played back. This type of graphic representation of sound implies that sound is actually a two-dimensional phenomenom. The first dimension always represents time. The second dimension represents the momentary air pressure at the place where the microphone is located. The air pressure can be expressed as a numerical or a voltage value. This also shows the limits of what can be done to sound once it is in the electronic domain, as there are only two possible directions into which the sound can be altered. A momentary level can be changed or transformed to another value on the vertical axis and a momentary level can be pushed forwards or backwards in time on the horizontal axis. Everything that can be done to a sound will be based on one of these two possibilities or a combination of both. A smooth repetitive compression or expansion on the horizontal axis is named frequency modulation, while smoothly varying changes in the vertical direction are named amplitude modulation. It is also possible to make jumps on the horizontal time axis, which creates a displacement in time which will delay the audio. Techniques like oscillator synchronization, echo delays, granular synthesis, but also techniques like filtering, are all based on creating displacements in time and how the time delays caused by these displacements are handled.

The reason that there are so many possible ways to process sound by electronic means is based on the notion that sound consists of wave patterns that span a certain amount of time. These wave patterns have their own properties, like harmonics and partials, each with their distinct frequencies, and a volume envelope. Certain processes have specific and well defined effects on each of the individual partials, e.g. odd harmonic distortion will create a series of odd numbered harmonics out of each partial that is present in the sound, and all these newly created partials will be added to the original waveform.

Waveshaping and distortion are techniques where the original waveform is manipulated on one or both axes in a way that the basic pitch of the sound is left unaltered. If the pitch is left unaltered there must be a change in either amplitude or timbre, or both amplitude and timbre. In general, waveshaping includes all techniques where the waveforms are changed on the graphic level. On the other hand, distortion includes all techniques that work on the individual partials in the sound. Waveshaping techniques in general create a lot of new and often very high harmonics, resulting in a very bright and fuzzy sound. Examples are wavewrapping, clipping and soft clipping. The individual levels of the partials do not matter much, as there is no clear relation between the individual partials present in the original waveform and the partials that the waveshaping process generates. The advantage of the more elaborate waveshaping techniques is that they can create distinct formant areas in the processed sound that can give the effect a very pronounced character. Distortion on the other hand can be much more subtle. Newly generated harmonics depend on the individual partials that are present in the unprocessed signal. This means that if there are little high harmonics in the original sound, most distortion techniques will have a grungy character in the mid of the sound spectrum. When applied with taste distortion can enhance the apparent presence of instruments in a mix instead of creating ‘a distorted sound’. Distortion works out particularly well if it is applied in only a small portion of the overall sound spectrum. Distortion can work very well on already recorded audio signals that contain chords and enharmonic signals like percussion and cymbal sounds. In fact, a certain amount of properly applied distortion is highly desirable in synthesized sounds. In contrast, waveshaping doesn't work out very well on audio material containing chords, etc. It is rather used on the oscillator or single voice level, e.g. to produce more character in the separate voices themselves in a popyphonic sound. Very often waveshaping is used right after a single oscillator and before a filter, while distortion works very well after a filter. So, although in general both waveshaping and distortion do not change the apparent pitch of sounds, they do have their own specific fields of application.

Distortion is always inherently present in any analog electronic device, although modern electronics are so good that the artifacts produced by this distortion usually fall below the treshold of hearing. Loudspeakers also inherently distort, and the distortion figures of loudspeakers can be quite serious for the cheaper ones. There are three basic types of distortion, even harmonic distortion, odd harmonic distortion and total harmonic distortion. Even harmonic distortion appears in radio tubes. Tubes have an amplification curve that is slightly bend like an exponential curve, though not as extreme as a true exponential curve. The effect is that the negative part of a signal is amplified slightly less as the positive part of the signal. This asymmetrical amplification will cause even harmonic distortion. An example of odd harmonic distortion is the saturation effect of magnetic recording tape. Recording tape has a limit to the strength of the signal it can record, similar to soft clipping. The more the signal strength approaches this limit the more the tape will resist to record at that strength. This effect is the same for both the positive and the negative part of a waveform, so it is a symmetric effect. This effect will cause odd harmonic distortion. Analog VCA circuits also exhibit this effect and when overdriven will cause odd harmonic distortion. Even harmonic distortion is said to sound more clean and natural compared to the more grungy sounding odd harmonic distortion, but qualifications like this are actually quite subjective and depend a lot on how distortion is applied. When simulating even harmonic distortion it is much harder to keep the effect in check as odd harmonic distortion, as the asymmetric effect can cause the positive part of a signal to quickly reach headroom levels and cause clipping. Even harmonic distortion can make a filter sound more steep, a technique that will be explained later, but it is a tricky technique that needs attention to prevent the mentioned possible clipping. Simulating odd harmonic distortion is less accident prone as both the positive and negative parts of a waveform are attenuated more as the signal level rises, so it can actually prevent signals from clipping. In fact, on many analog synthesizers the VCA circuit that comes after the filter circuit is allowed to be overdriven to reduce jumps in signal levels when the filter is set to a very high resonance value. This way the overdrive effect acts as sort of a signal level limiter. In practice analog electronic components have a limit to their working range, e.g. it is impossible to amplify a signal to a level that would exceed the power supply voltages. When a level is close to a power supply voltage the amplifier starts to refuse to amplify further which creates a saturation effect. As this is a symmetrical effect devices like radio tubes do not only create even but also odd harmonics. In this case it is common to talk about total harmonic distortion.

When there are chords or enharmonic sounds in the audio material that is distorted the partials start to interact with each other and create intermodulation or IM distortion. Note that both even and odd harmonic distortion also create IM distortion. As distortion is always the result of some nonlinear effect, the new partials produced by the distortion will have frequencies that are the sums and differences of the frequencies of the partials in the original signal. This means that of a pure quint is played with pitches at 220 Hz and 330 Hz, a partial at 110 Hz will be produced, as 110 Hz is the difference between 220 Hz and 330 Hz. This 110 Hz partial will start to act as a subharmonic that gives a low bottom to the sound. Electic guitar players use this IM distortion principle almost unconciously. The main reason why guitar amps and speaker cabinets have relatively high distortion figures is to have the IM distortion produce a grungy low bottom end in the guitar sound when chords are played. It is quite important to have the instrument well tuned, as when not properly in tune the IM distortion also creates a strong and probably unwanted beating at a low frequency. An important thing to note is that IM distortion is like a recursive process, meaning that the partials produced by the distortion will also immediately intermodulate with the original and newly created partials. This effect increases exponentially when the distortion depth is increased. E.g. if the quint from the previous example was tuned at 220 Hz and 331 Hz there would be a new partial at 111 Hz (331-220). This partial would also intermodulate with the 220 Hz and create a new partial at 109 Hz (220-111). And the new 111 Hz partial would intermodulate with the new 109 Hz partial to produce a beating at 2 Hz (111-109). A guitar player can tune his guitar strings to get just the right effect for the relatively few chords used in most pop songs. He can also correct the tuning of chords by bending some strings and even use the beating as an expressive effect. But on common synthesizers this type of individual voice bending control is lacking. So, distortion should be used with care on synthetic sounds to prevent unwanted strong beating effects. E.g. a lot of distortion on a chorused or unisono sound will in general sound very nervous, as the distortion strongly exaggerates the subtle beating that is already present in the unisono effect. Deep distortion on a reverberated sound is considered pretty aweful by most people, and is indeed hardly useable, even as a special effect.

The trick to applying distortion is to apply it only to specific frequency bands. To do so the sound must first be split up in different frequency bands by using crossover filters. It hardly pays to apply distortion to the frequency band above 2.5 kHz if the sound is already quite bright. But when used with care it can freshen up a dull sound, e.g. aural exciters are based on adding subtle distortions to the very high ranges of the sound spectrum. Subtle distortion in the range between 500 Hz and 2.5 kHz can greatly enhance the presence of a sound in a mix and can be an important method to improve the overall sound. Distortion below 500 Hz can easily make the bass range sound muddy, so it should be used quite conciously. It depends a lot on how the bass and the kickdrum work together. In general it is best to apply distortion separately to the bass and the kick before they are mixed together, to prevent strong IM distortion between the bass and the kick.

Transfer function

All types of waveshaping and distortion that work by manipulating the momentary amplitude level can be drawn in a simple graph that shows the transfer function in a graph. The horizontal axis of the graph spans the range for all posiible input values and the vertical axis spans the range of all possible output values. In most cases this graph will have linear scales on both axes. To work with the graph a momentary value that is found on the vertical axis of the earlier mentioned waveform plot is drawn on the horizontal axis of the transfer function plot. The transformed value can be found on the vertical axis of the transfer function plot and this value will substitute the value in the original waveform plot. If the line on the transfer function plot is curved or has sudden corners the plot is nonlinear, as if the line would have been straight there would only be a linear amplification or attenuation depending on the angle of the straight line. Basically any function that produces a curved or cornered line will produce some waveshaping or distortion effect. If the input is a sawtooth waveform that spans the full dynamic range the resulting waveform will have the same shape as the line in the graph. This particular case is very useful to understand what actually happens in the distortion process. First observation is that as the sawtooth contains all possible harmonics with smoothly decaying amplitudes for the higher harmonic numbers, and so is free of formants, the new waveform will have some harmonics enhanced and others attenuated. So, the new waveform will have formants. These formants will have a place in the audio spectrum that is relative to the pitch of the waveform, increasing the pitch will also shift the new formant areas up in the audio spectrum. Another observation is that if there are corners in the transfer function the new waveform will also have corners, and these will contain much sonic energy in the highest part of the audio spectrum. So, if the graph is not smoothly curved the resulting wavefrom will sound fuzzy. But if the graph is a smoothly curved line the new waveform will have a more grungy character, meaning that the harmonics just above the fundamental will be enhanced, and little energy is added in the very high parts of the audio spectrum. This means that the energy in the melodic part of the audio spectrum is enhanced, which can increase the perceived presence of the sound in a mix without having to boost the overall volume of that sound. This effect is mainly based on psychoacoustic principles, or how and where the mind tends to focus in a mix. Needless to say this is an important technique in mixing and mastering. Still, using distortion to improve a mix is a subtle and delicate art that needs quite an amount of practice. First it must be determined where in the mix more presence is needed and then a proper technique must be applied with subtlety to get a result that is not overdone but leads to just about the right balance. As only the presence should be increased and the effect should not sound distorted. It is impossible to give recipes that always work, as different material will probably need different treatment. There is a lot of intuition involved here. The only available tools to judge the final results are your ears. Meaning that careful listening is highly recommended.

Waveshaping

On an analog oscillator with multiple waveform outputs the waveforms are internally derived from one basic waveform by means of a technique named waveshaping. In most cases the oscillator itself generates a sawtooth waveform. As the input to the waveshaping transfer function is a sawtooth, the graph of the transfer function is equal to the new waveform to be created from the sawtooth waveform. On a digital system a lookup table can be used, the momentary sawtooth value will in this case be the index to get a value from the lookup table. By describing the new waveform in the lookup table a sawtooth waveform can be transformed into virtually any new waveform. This technique is sometimes named wavetable synthesis. If there are more lookup tables stored in the system dynamically changing waveforms can be created by smnoothly crossfading between the results of two or more lookup table transfers. Instead of lookup tables specific functions can be used to get specific waveforms, e.g. using the momentary sawtooth value as input for a sine function will generate a sine wave. Analog systems will use the specific properties of certain electronic components like diodes or devices like opamps or comparators to create the transfer functions to transform the sawtooth waveform into other waveforms.

Following is a description of common methods to cretae the more common waveforms found on analog oscillators. The pulse waveform is derived from a sawtooth by comparing the current level of the sawtooth to a constant value. When in the comparison the current level is greater the pulse output will be positive. And if the current value is less the pulse output will be negative. The transfer function plot will show a straight vertical line. Every input value that is left of this line will transform to the maximum negative value and every value right to the line will transform to the maximum positive value. Varying the compare level by e.g. a slow triangle waveform will achieve pulsewidth modulation. Basically the vertical line in the transfer function will be shifted from left to right and back again. A triangle waveform can be derived from the sawtooth waveform by folding down the upper halve of the sawtooth waveform. Alternatively the upper quarter of the sawtooth can be folded down while the lower quarter of the sawtooth is folded upwards, until their ends meet. The triangle waveform can be changed into a sinewave. On an analog oscillator this is often done by feeding the triangle through a device that has a voltage dependent resistance. On a cheaper system two diode components are used, although diodes will not produce a very pure sinewave. This method also needs careful trimming to get the least harmonic distortion in the sinewave. On a digital system a much more pure sinewave can be created by either using a lookup table that describes a sine wave or by using a mathematical function based on what is known as a Taylor series evaluation. This last method can be computed quite efficiently and can produce a very pure sinewave without having to use a long sine function lookup table stored in memory.

The mentioned techniques are used to create the waveforms that are commonly used on analog synthesizers, but those waveforms can be manipulated further to create more waveforms with certain desirable sonic properties. The basic waveforms, except for asymmetrical pulse waveforms, all have a harmonic series that falls off smoothly, meaning that there are no strong formant properties in the timbre. To create more characteristic timbres waveshaping should introduce formants, and in most cases it will.

A common approach to creating suitable transfer functions for waveshaping is to divide the input range into two or more segments. In the transfer function graph these segments show on the horizontal axis. The angle of the transfer curve line differs for each segment. The graph lines for each segment do not necessarily have to join, if they do not join it will create a sharp vertical transient in the final waveform when the input value crosses the border between the two segments. If the segment lines do join, a corner is created in the final waveform. Technically the segments can be created by using one or more voltage or level comparators that control a set of switches. Each switch passes on the input signal with a controllable amplification factor plus an additional variable level offset. The offset levels can be set in a way that the line segments in the transfer function graph join ends to suppress unwanted transients. There are several variations possible on how the comparators and switches can be set up, which is up to the synth designer. The G2 system offers a module named a control sequencer which is a very convenient setup that divides the input range into sixteen equally spaced segments. This module can be set to interpolate between sixteen slider values that provide the parameters for each segment. If the input signal amplitude varies between 0 and +60 units a very flexible waveshaper is created. The input waveform oscillator that drives this waveshaper can e.g. be a shaper oscillator set to the waveform that morphs between a triangle and a sawtooth. The basic waveform can be set by ‘drawing’ the waveform with the sliders and then the timbre can be dynamically altered by modulating the triangle<->sawtooth input waveform with e.g. a low frequency oscillator.

Clipping

Clipping clips off the top or both the top and the bottom of a waveform. Depending on the original waveform the effect can be from subtle to quite extreme. The transfer function plot is divided into three segments. The middle segment shows a straight line at an angle of 90 degrees going through the centre or origin of the plot. When the graph line reaches the left and the right segments the line makes a corner and becomes horizontal in both outer segments. Clipping can produce a lot of high harmonics and often works best on a raw waveform before it is fed into a filter. When a moderate amount of clipping is used on a sawtooth or a triangle waveform it will increase the presence of the fundamental in the waveform, giving the final sound a bit more beef without destroying the basic character of the sawtooth or triangle waveforms. In most cases the clip levels are controlled by a fixed value and can not be modulated. But an interesting modulation effect is created by adding a slow triangle waveform to the audio waveform before it enters a clipper module. The sonic effect is that of a lively change in timbre that sounds related to pulsewidth modulation. Note that clipping only works on waveforms that have smoothly rising or falling slopes, e.g. on square and pulse waveforms clipping doesn't have any sonic effect at all.

A disadvantage of many clipping modules is that when the clipping levels are changed, the overall amplitude of the output might be set to the same levels. This means that the volume can drop significantly when clipping levels are set to more extreme values. There is a very simple way to overcome this by actually using the clipper in a feedback loop in a mixer. The idea is that even with great amounts of feedback the clipper will always clip the feedback signal to the set levels. So, by setting the clip levels to fixed values and controlling the amount of feedback, the output amplitude will remain constant and it will become easier to work with clipping. Of course this only works when both the top and the bottom are clipped. By using a three input mixer both the amount of clipping and the clipping modulation can be conveniently controlled. The output of the mixer is fed into the input of the clipper, while the output of the clipper is fed back into one of the mixer inputs, the second mixer input receives the audio signal while the third input can receive a slowly varying modulation signal. Final output is taken from the output of the clipper module. Instead of using a second slowly varying waveform a second audio waveform can be used as well. Tuning this second waveform to a just pitch ratio, e.g. to 2:3 or 3:4 can create thick and bright sonic results. Adding even more audio waveforms and using enharmonic detune ratios can give thick metallic timbres useful as basic material for bright and metallic percussive sounds like metal can hits, gong sounds, etc.

An alternative for the G2 Clipper module is to use a modulatable crossfader module that crossfades between two fixed values. The advantage of the modulatable crossfader is that the clipping action takes place on the modulation input, if the input value on the modulation input exceeds either -64 or +64 the crossfader will stay fixed to either the A or the B crossfader input. This means that when e.g. a triangle waveform is fed into the crossfade position modulation input the crossfader output will vary between the two fixed levels on the A and B inputs. The input modulation level input will set the clipping sensitivity while the values on the A and the B input set the minimum and maximum levels the waveform will clip to. As these A and B values can be set to any value this clipper setup can force a waveform to be shifted into a clearly defined amplitude range that it can never exceed. So, the two fixed A and B values define where the boundaries between the outer two segments and the middle segment are positioned, while the middle segment line always joins at its two ends with the horizontal lines of the outer two segments.

Soft clipping

The transfer function for soft clipping is almost similar to the transfer function of a clipping module. The difference is that as the middle line segment in the clipper transfer function approaches the minimum or maximum limits it starts to smoothly bend towards the limits, so the corners are softened. Soft clipping produce much less energy in the very high parts of the audio spectrum, making it sound less fuzzy. Soft clipping is often more useful as straight clipping, unless a very large amount of very high harmonics is needed. Many times soft clipping is created by using a slightly curved line that is derived from a simple exponential function. This method is computationally simple and quite effective. An even better result is achieved by using a sine function where the range between -90 degrees and +90 degrees fills up the middle segment. This will result in a nicely grungy soft clipping effect where the newly produced harmonics are well balanced and sounding a bit more organic as when using exponential functions.

Wavewrapping

Wavewrapping uses similar folding circuitry as is used to derive a triangle wave from a sawtooth wave. But wavewrapping offers the possibility to dynamically fold the top and bottom in a way that 'multiple folds' can be created. When applied to a triangle waveform, the wavewrapping amount modulation creates an effect that is sonically very similar to using hardsync on a triangle wave, meaning that it results in a strong sweeping formant effect. The effect on other waveforms can be quite harsh, as it can create even more high harmonics as clipping does. The amount of wavewrapping works very well on both triangle and sawtooth waves when the amount of wrapping is set to a fixed level and a slow triangle waveform is mixed to the audio waveform before it is fed into the wavewrapper module. The amount of the low frequency modulation signal is best set to only one third or less of the fixed amount of wavewrapping. This will create a lively effect that can be enhanced by using an extra chorus. Setting the overall envelope attack rather slow and using long note decay times will result in characteristic padsounds with a slightly ethereal sound. Replacing the low frequency oscillator with an AD envelope generator can create a characteristic attack.

Nonlinear waveshaping

Virtually any mathematical function that uses one input and one output value can be used to produce nonlinear waveshaping. Such functions can use extra control values that dynamically alter the transfer function. In computer graphics smooth curves can be drawn with functions named Bezier curves and B-spline or qubic spline curves. For smooth graphic curves these functions are used in two dimensions for curved line segments or three dimensions for curved surfaces. They can also be used in one single dimension and then can be used directly to modify a waveform into another waveform. The idea behind such functions is actually quite simple, imagine first that there are two control values and a crossfader that fades between these two values. The audio input signal is used to control the position of the crossfader. What will happen is that the waveform on the output of the crossfader is a copy of the input waveform, but its minimum and maximim amplitude value will be equal to the two control values. By changing the two control values the waveform amplitude is attenuated and shifted up and down. Basically, if the crossfader is at one end only that control value will define the output and if the crossfader is at the other end the other control value will define the output. In the middle the effect of both control values will be fifty-fifty. Now imagine that the crossfader is adapted in a way that in the middle the effect of both control values is only 25%. But a third control value is added in a way that its effect in the middle is full but at both extreme ends it is zero.The curve for this third value must be smooth, like a bell shape. The three curves for the control values can be chosen insuch a way that if the end values are +1 and -1 and the middle value is zero and the input waveform is a sinewave the output is also a sinewave. But when the three control values are set to randomly chosen values the input sinewave will be shaped into another smooth waveform. The curves for the three control values are named blending curves and define how much effect the corresponding control value will have on the output value for a certain input value. The more control values the finer the control on the resulting waveshapes.

Harmonic distortion

Introduction

The purpose of harmonic distortion is to generate new partials from an existing audio signal and add those partials to the original sound. If these partials are harmonic to the partials in the original sound they will blend with the original sound and subtly change the timbre. The effect of harmonic distortion is different to the effect of filters, as filters tend to take away and emphasize existing aspects of a sound while harmonic distortion can create and add new aspects to a sound. All analogue circuitry does to some extent exhibit harmonic distortion, but most circuitry, e.g. HiFi amplifiers, are designed to generate as little harmonic distortion as possible to get a faithful reproduction of the original signal. In the digital domain, after a signal is digitized by an analog to digital converter, the computer code instructions that act on the stream of numbers representing the sound do not add any harmonic distortion, simply as the operations initiated by the instructions are strictly linear.

The amount of harmonic distortion is expressed as THD (Total Harmonic Distortion). THD is measured by subtracting the original input sound from the distorted sound in a way that a signal with only the distortion is generated. Then the average energy of the distortion is compared with the average energy of the original sound and this ratio is expressed as a percentage. When the THD of an amplifier is below 0.1% THD it is considered to be HiFi. Below this value the amount of THD is hardly noticed by the average listener. When distortion is wanted for musical purposes the THD is exaggerated by design to figures that might go way up to 30%, which results in a severely distorted sound. But the THD value doesn't say much about the sonic effect, as it does not specify which harmonics are generated and in which range of the audio spectrum. So, two harmonic distortion devices from different manufacturers can both have a measured THD of 30% but still sound completely different. E.g. one may have a muffled grungy effect, while the other might add a bright fuzzy edge to a sound.

When a monophonic single pitched sound is distorted using harmonic distortion, either only odd harmonics or a mix of odd and even harmonics will be added to the sound. These odd and/or even harmonics are created from every partial present in the original sound. When instead of a single pitched sound a chord is distorted, an extra effect can be noticed which is caused by intermodulation of the harmonics that are generated from the different pitches in the chord. This effect is named intermodulation distortion or IMD. These extra partials might be harmonic or enharmonic and can have pitches below the lowest pitch in the chord. When such a low-pitched partial is harmonic to one of the pitches in the chord it is commonly named a subharmonic. These subharmonics will add a grungy bottom under the chord. Tuning becomes essential here; a just tuning will sound better as an equal temperament tuning. The faster beating in the equally tempered chords will be strongly exaggerated by the harmonic distortion, which sounds uneven and in general not very good. In contrast, the very slow beating in just tuned chords will enhance the effect of tension in the sound, giving a sense that the sound is going somewhere. IMD also points to the new partials created from the partials already created by the distortion, most of these will be enharmonic. Because of the possible enharmonic products of harmonic distortion designers of recording and mixing equipment consider harmonic distortion a little devil that must be fought fiercely. Rock musicians on the other hand discovered that harmonic distortion boosts the impact of e.g. the rock guitar sound tremendously, and deep harmonic distortion is sort of the trademark of styles like heavy metal. Another early example of use of harmonic distortion is the heavily overdriven Hammond organ sound, often combined with Leslie speaker cabinets, as used in the psychedelic music of the late sixties in the twentieth century.

Analogue distortion devices make use of nonlinear properties, e.g. saturation effects, in a suitable component to create a distortion effect. An example is a property that when a voltage over the component is increased the component's electrical resistance will gradually decrease. When such a component is used in an amplification circuit it can result in a transfer curve where a higher input voltage value will be amplified less than a lower input voltage. Examples of suitable components that exhibit this behaviour are the germanium diode and analogue VCA circuits based on OTA chips (Operation Transconductance Amplifier). These OTA chips can have a THD percentage that can be around 10%. Magnetic recording tape exhibits a similar property named tape saturation, which is the point where the tape refuses to magnetize deeper when the recorded signal is increased in amplitude. These three examples are just a few of the many options that an analogue electronics designer can use to create a harmonic distortion device. Main characteristic of saturation distortion is that both the positive peak and the negative peak are gradually compressed to a certain maximum signal level. When both the positive and the negative signal peaks are compressed by equal amounts this type of distortion is named symmetrical. Symmetrical distortion will generate only the odd harmonics of a single sine wave input signal or of each partial in the sound. When one of the polarities is compressed slightly less than the peak of the other polarity the distortion is asymmetric, which will result in the generation of extra even harmonics in addition to the odd harmonics. The compressive effect is an important property that can be put to good musical use. Note that this compression effect is instantly and is different to how a studio compressor works; studio compressors use the envelope of an audio signal to slowly control the compression rate, while odd harmonic distortion uses the audio signal itself for immediate compression.

Vacuum tubes have a slightly more complex nonlinear behaviour, as the transfer curve of a tube is slightly asymmetric. This means that positive peaks are compressed differently than negative peaks. This will create both odd and even harmonics, though the odd harmonics will in general have a stronger presence. Designing a tube amplifier is an art by itself as the amplification curve of a tube is bent with a complex curve. A HiFi amplifier designer will try to find a part of the curve that is virtually linear. In contrast, the designer of a guitar amplifier will in general have the tube work in a range in the curve that is highly nonlinear and probably drive the tube into saturation as well, which accounts for the typical character of a distorted guitar sound when a tube-based guitar amplifier is used. Different types and brands of tubes can have different amplification curves, meaning that the type of tube used can make quite a difference to the sound. Often a tube is set to a range where peaks of one polarity are heavily compressed, while the peaks of the other polarity are strongly expanded. The expanded peak will quickly 'explode' to a very high voltage and create lots of very high harmonics, an effect that is sometimes named 'tube screaming'.

Emulation of analog circuitry

The best way to recreate analogue types of distortion by digital means is to use a technology named ACE (Analogue Circuitry Emulation). ACE is similar to physical modelling of acoustic instruments, but instead of modelling the physical aspects of an instrument the physical aspects of a certain analogue component or an analogue circuit is modelled by an algorithm in a piece of computer code. Keep in mind that the basic instructions in a computer chip do not have the quirky properties of analogue components and these properties must always be recreated by writing the proper computer code. ACE is all about how to write such code. It is also possible to patch ACE models on an analog modular synthesizer, in which case the modular synthesizer is used in a similar way as one would use an analogue computer of the late fifties and early sixties of the twentieth century. Mixer modules in combination with signal inverters do the additions and subtractions, while ringmodulator modules and VCA modules are used for multiplications. Fixed voltage modules will provide the necessary parameters.

ACE concentrates on two important aspects of an analogue component or a circuit, 1) the transfer curve and 2) the effect on the frequency spectrum. A discrete analogue component like the germanium diode has a transfer curve that is fixed for the whole audio range. But a more complex analogue distortion circuit can have different transfer curves for different ranges in the audio spectrum. Meaning that to emulate such an analogue circuit a whole lot of transfer curves could be needed. In general this is not much of a problem, as it is often the same curve that simply tends to become more linear in the higher frequency ranges. This has as a result that the higher pitched partials in a sound produce less distortion as lower pitched partials. When only a moderate amount of distortion is used this tends to give the sound a bit more body in the mid range of the spectrum without making it brighter, as most newly generated partials will be in the low and mid ranges of the spectrum. This tends to increase the presence of a sound, which is generally perceived as pleasant. Especially on chords or loops it is often important that a harmonic distortion circuit does not produce a lot of extra energy in the highest parts of the spectrum, as this will lead to problems in a mix with vocals and acoustic instruments and/or destroy the sense of spaciousness in the overall sound. Rule of thumb is that the perceived increase in sonic energy in the very high (> 4kHz) should be considerably less as the sonic energy increase in the mid-high ranges around 2.5 kHz.

Note that digital sound generation and processing algorithms that do not take these ACE principles into account in general sound thin, flat and overly bright compared to pure analogue musical equipment. Even if a piece of analogue equipment is trimmed to have as little THD distortion as possible, it still tends to have a fuller sound as digital equipment. But when e.g. the effect of a tube screamer is required the transfer curve of the distorter might become more and more nonlinear when the pitch goes up, until a certain point in the audio range where a lowpass function kicks in to block the generation of very high partials. This will account for a slightly resonant sonic character of the distortion. E.g. the transition point might be around 2kHz, below this frequency distortion increases for higher pitched partials but above this frequency both distortion and amplification decrease very fast.

Transfer curves

Transfer curves can be modelled by two methods, the first is to use a lookup table that simply describes the nonlinear transfer curve, second is to use a formula that approximates as closely as possible a suitable nonlinear transfer curve. The advantage of using a lookup table is that an accurate measurement of an existing component can be taken to fill the lookup table. Disadvantage is that huge tables must be used, e.g. for a 24-bit signal resolution 3MB of memory is needed to store the table. Another disadvantage is that the table is static and that when different curves for different frequency ranges are needed a whole lot of memory is needed to store the tables, plus a method to interpolate in between tables. In contrast, formulas do not need memory to store tables and have the advantage that formula parameters can be manipulated in real time and be made controllable by varying control signals like modulation oscillators or the actual amplitude envelope of the input signal. Except for some vacuum tubes the transfer curves are often simple polynomial equations with only few and straightforward parameters. Still, using tables or using formulas are both valid within the principles of ACE.

The use of formulas opens up additional territory, as basically any nonlinear function can be used to produce harmonic distortion. So, next to formulas that approximate transfer curves of existing components and circuits, different nonlinear functions can be used which emulate 'fantasy components' that do not exist as such in the real world. Here is of course lots of room for experiments and chances for happy accidents.

The important thing to always keep in mind when creating harmonic distortion of some type is that distortion always works on existing audio material as input. This audio material will have a specific sonic character and the only valid assessment on a certain distortion effect is how it works out on the sonic character of the original audio material. Distortion will add some of its own character and this should blend well with the original character of the input material. If it doesn't blend well, the distortion should be tweaked until it does blend well, or perhaps using the chosen type of distortion wasn't such a good idea after all. In general, distortion will almost always be acceptable on a single pitched sound or a single percussive hit, be a little more difficult to apply on chords or percussive loops, and be very difficult to apply on a whole mix, especially when vocals and acoustic instruments are included in the mix. As a rule of thumb distortion is generally applied per instrument and sometimes separate on each voice in a delicate polyphonic instrument sound. On a monophonic 'fantasy' synthesizer sound distortion can in general be applied in generous amounts without doing much harm. But applying distortion during the mastering process of a recording is in general considered not done, although this might depend on the musical genre. It is also common to use a crossover filter to split the audio spectrum into two or more bands and apply different amounts of harmonic distortion to only the lowest and/or the middle bands, but rarely on the highest band.

A VCA-based harmonic distortion element

The element that is to be described here can emulate both germanium diodes and tape saturation. The idea is to create a gain cell that, when no input is applied, is at exactly unity gain. Then, when the amplitude of the waveform at the input increases, the gain cell will reduce amplification with an approximately logarithmic curve. An important property will be that amplification will never exceed unity gain, which will make it an ideal element to be used in a feedback loop of a tape echo emulation, an overdrive-type distortion, etc., as being below unity gain prevents overload or unwanted oscillations through the feedback path.

In its simplest form the distortion curve is symmetric, but it can easily be adapted to produce a variable amount of even harmonics as well. At the core is a VCA or multiplier that receives a fixed control signal that will cause the VCA to amplify at exactly unity gain. This control signal is named the bias signal. The trick is to extract a modulation signal from the input signal that will modulate the bias signal in a way that the VCA amplification curve will become logarithmic. To accomplish this the modulation signal will have to go from zero to negative when the input amplitude increases towards either a positive or a negative peak. This negative 'bias modulation signal' is simply added to the bias signal, so that when the input signal increases the final control signal for the VCA will drop and decrease the VCA amplification. The most obvious analogue way to derive the bias modulation signal is to use a full wave rectifier circuit and negate its output so it has a negative polarity. A less obvious but superior way is to generate the quadrate of the VCA input signal by feeding it into both inputs of a four-quadrant multiplier module or both inputs of a ringmodulator module, and then negate the output. The quadrate of a bipolar signal will always have a positive value and so will act as a full wave rectifier as well. The reason why this is a superior method when creating harmonic distortion is because the quadrant of a sine wave will also be a sine wave, but with twice the frequency, so only the second harmonic of the input sine wave will be generated. A diode-based full wave rectifier circuit, or an 'absolute value' computer code instruction, will also produce a signal twice the frequency, but already with a lot of harmonics added. These extra harmonics will somewhat limit the possibility of having a controlled gradual build up of harmonics, especially the even harmonics. In contrast, the multiplier/ringmodulator will offer ways to gradually build up a harmonic series from each sine wave partial in the input signal. Note that the partials produced from the partials in the input signal in the multiplier are only the second harmonics of the input partials, and with a suitable technique these new harmonics can be used later to create the higher harmonics in an easy to control way. By using the quadrate to modulate the VCA bias signal it is possible to e.g. gradually build up only an odd harmonics series for each of the partials in the input signal. Or add a gradual build up of an even harmonics series. It will also be possible to exactly calculate the level of all the produced harmonic signals, as when this circuit is expressed as a mathematical formula it closely resembles Chebyshev polynomial formulas. But instead of going deep into the mathematics a hands-on approach will be used that is easy to patch or program and can be tweaked by ear for the final sonic result. Only a few guidelines will be given, but make note that these are quite important to get a stable nonlinear gain cell. From here on it will be assumed that a multiplier or ringmodulator is used for the full wave rectification.

Using a multiplier as a rectifier in the gain cell will produce only little extra sonic energy in the higher parts of the audio spectrum, most sonic energy of the harmonic distortion will be in the middle and lower parts of the audio spectrum. Sonically, this means that harmonic distortion will tend to increase the presence of a sound in the midrange without making it specifically brighter or fuzzy. In contrast, most waveshaping techniques, like clipping etc., do produce a lot of sonic energy in the high part of the audio range and add only little, or even reduce, sonic energy in the low and mid parts of the audio range.

The rectified signal will itself increase when the input signal amplitude increases and when the circuit receives a very strong input signal this bias modulation signal might become so high in amplitude itself that it will cause the final bias signal on the control input of the VCA to become negative. This situation should be avoided, so the rectified and negated input signal should be attenuated in a way that when a signal at system headroom level is fed into the circuit the final VCA bias signal should still be positive. When the input of the full-wave rectifier is taken from the output of the VCA, instead of the VCA input, the chances that the modulated VCA control signal becomes negative is greatly reduced. The reason is that the rectifier will use the already slightly compressed VCA output signal instead of the full level input signal. However, this also creates a feedback situation, the feedback signal flowing from the modulated VCA output to the VCA control signal. This means that although chances of a negative bias signal are greatly reduced, the feedback path will increase the chance that an internal oscillation could occur. The oscillation would prefer half the sample rate as its resonant frequency, which is probably an inaudible frequency but it will make the gain cell highly unstable and unpredictable. This simply means that a balanced choice between two evils has to be made in a way that the final circuit is stable under even extreme working conditions (e.g. a square wave signal that alternates between positive and negative system clipping levels). Analogue feedback circuits suffer from the same tendencies to oscillate, although these circuits prefer radio frequencies. To prevent radio oscillations in analogue circuitry band limiting is used in the feedback path of e.g. operational amplifier circuits. In digital circuitry a similar solution can be used, e.g. by inserting a 6 dB lowpass filter with a cutoff frequency set to about 5% of the sample rate. On a 96kHz system this would be about 5kHz. Inserting a lowpass filter at 5kHz would also suppress the generation of harmonics above 5kHz, which is sonically not a bad thing at all.

Increasing distortion depth

The harmonic distortion produced by the gain cell is only moderate in depth. Still, the sonic effect will be that the presence of the mid range of the audio spectrum seems to be somewhat increased, instead of giving a clear sense of a distorted sound. But on a chord the mentioned grungy low bottom will be clearly present. Distortion depth can be greatly increased by placing the gain cell in the feedback loop of a simple mixer module. One mixer input will receive the audio input signal while the other input receives the output of the VCA. Audio output is still taken from the output of the VCA. In essence this means that the distortion cell is placed in the feedback loop of a mixer's output back to one of its inputs. The compressive action of the distortion cell will keep signal levels in this 'outer loop' in check, while the build up of harmonic partials is intensified by this outer feedback loop. There is a lot of room for experimentation here, e.g. placing a carefully tuned allpass filter in this outer feedback loop it is possible to create the sonic effects of e.g. a tube screamer. On the waveform level the slight phase delay caused by the allpass will create an effect that is similar to the slightly delayed effect of a compressor on a percussive hit, which will emphasise the hit of the percussive sound. When experimenting with this kind of technique it is important to judge the sonic effects by ear; when it sounds good, and there seem to be no internal oscillations caused by exaggerated feedback levels, all is fine.

Frequency modulation synthesis

Introduction to FM

FM synthesis is in general considered to be complex, possibly because the wellknown DX7-type synthesizers from the eighties offered a complex model that only very few knew how to handle. Still, FM doesn't have to be complex. It is very well possible to use a hands-on approach that quickly leads to the wanted results. It is not at all necessary to know the math that was used in the past to describe FM, instead it is more worthwhile to experiment with simple patches using only one or two oscillators and building experience from there on.

In essence FM is the modulation of the frequency parameter of an oscillator with a signal in the audio range, meaning that FM can be used on any oscillator that lets itself be smoothly controlled in frequency at audio range. For the FM technique the oscillators must be absolutely stable to get predictable results. Most analog oscillators are not stable enough, so FM is almost exclusively used on digital synthesizers.

Modulating the frequency parameter

The frequency parameter can be modulated in a linear or in an exponential fashion. When using exponential modulation, by using a Keyboard Pitch or V/Oct input, the results are easily enharmonic. Using linear modulation gives much better results, but requires a dedicated FM or V/Hz control input on the oscillator. It goes too far to explain in detail the difference between these two input types, as a rule of thumb just remember that a Pitch input is relatively useless for FM in the audio range and in general the dedicated FM input is used instead. Some digital oscillators have an option to modulate the momentary phase position of the waveform instead of the actual frequency parameter, which can be imagined like shifting the waveform forwards and backwards in time. E.g. on the DX7 it is in fact the waveform phase position that is modulated and not the linear frequency parameter. The main difference is that phase modulation does not detune the basic pitch of the oscillator when the oscillator is modulating itself. If this 'selfmodulation' is instead applied on a true linear frequency modulation input (like on an analog oscillator) it will in fact severely detune the oscillator.

Creating timbres

Creating timbres with FM is based on the priciple that there is a tight relationship between amplitude modulation and frequency modulation. Imagine a graph of a waveform, e.g. the graph of a triangle wave. This graph is a two dimensional picture with an X-axis that denotes time and a Y-axis that denotes amplitude. This picture can be distorted vertically, in which case the distortion is named amplitude modulation. It can also be distorted horizontally, and then the distortion is named frequency modulation. It can also be distorted in both directions, which has no special name. In all three cases the waveshape will change and thus create a new timbre. In general this technique is named waveshaping, creating a new waveform with a different timbre from some basic waveform. The interesting and almost paradoxal thing is that amplitude modulation is in certain cases able to keep the waveform intact and only cause a steady change in frequency. And in certain well-defined cases frequency modulation is able to create new waveforms at the original pitch. These last cases is what FM synthesis is all about.

Linear FM with two oscillators

When one oscillator is used to FM modulate another oscillator the oscillator that gets modulated is commonly named the carrier-wave oscillator or simply the carrier. The oscillator which modulates the carrier is named the modulator. When using one carrier and one modulator there are four factors that define the resulting timbre of the modulated waveform.

The first factor is which waveforms are used on the carrier and on the modulator. Many dedicated FM synthesizers use sinewaves for both the carrier and the modulator. But FM can be done with any waveform for the modulating oscillator and most waveforms for the carrier.

The second factor is the detuning or frequency relation of the carrier and the modulator, which is named the frequency ratio. The frequency of the carrier is the reference frequency to define the ratio, meaning that the ratio can be simply calculated by dividing the modulator frequency by the carrier frequency, while using the values in Hertz for the division. If the modulator is tuned to a harmonic of the carrier, this ratio will always be a whole number that is also the number of the harmonic. E.g., if the carrier is tuned to 100 Hz and the modulator is tuned to 300 Hz the ratio is 3:1 (and 300 Hz is also the third harmonic for 100 Hz). Often the ratio of both the carrier and modulator is not set in relation to each other, but in relation to the pitch of the note played on the keyboard. In this case both carrier and modulator have a separate ratio setting, e.g. 4:1 for the carrier and 6:1 for the modulator. The relation between the carrier and the modulator will now be the ratio of the modulator divided by the ratio of the carrier, in the example 6:1/4:1 => 6:4 => 3:2. If the ratio is a whole number like 3:1 or a simple rational number that happens to be a pure chord interval, like 3;2, 4:3, 3:5, etc., the resulting timbre of the modulation will sound harmonic. But if the ratio is 'more difficult', like 1:3,57342, the modulation will generate so many unrelated partials that the timbre will sound distinctly enharmonic.

The third factor that defines the resulting timbre is the depth of the modulation. The modulation depth is defined by the amplitude of the modulating signal only, increasing this amplitude will 'widen' the frequency sweep of the carrier. In general deeper modulation will create a brighter timbre as it will 'sweep through' more widely spread harmonics of the carrier pitch. The modulation depth can be expressed as the difference between the basic frequency of the carrier and the maximum frequency the carrier can reach in the frequency sweep caused by the modulation. This relation is named the frequency deviation. E.g. if the basic carrier frequency is 1000 Hz and the modulation will cause the carrier to sweep between 600 Hz and 1400 Hz, the frequency deviation is 400 Hz.

The fourth factor is the phaseshift between the carrier waveform and the modulator waveform. If both the carrier and the modulator use the same waveform and are set to the same basic pitch and the modulation depth is constant, the timbre will still change dramatically if the modulator waveform is shifted in phase compared to the carrier waveform. This phase shift is the little devil with FM synthesis. The first three factors can in general be easily and exactly set, but this phase shift can still mess up these three settings, as the phase shift between two oscillators is basically undefined. Simply because both oscillators are independent modules and 'have no knowledge what the other one does'. Some extra vibrato LFO modulation on one or both oscillators can also cause apparently random phase shifts between the two oscillators. The only thing that can be done to get control on this phase shift is to force or reset both oscillators to a predefined phase position on a keyboard trigger, and probably restart the extra modulating LFO's as well on a key press. The simplest way to do this restart thing is to connect the keyboard gate or trigger signal to the hardsync inputs on both oscillators and optionally the reset inputs on LFO's. This will force these modules to reset their waveforms on a keypress and give a predictable sound on each keypress. While doing experiments with FM it is adviseable to use this hardsync trick with the keyboard trigger signal to eliminate the effects of this phase shift factor. In a later stage you can always disconnect one or more hardsync inputs to get a more lively sound, but you will most certainly notice changes in timbre on each new note.

FM tracking modes

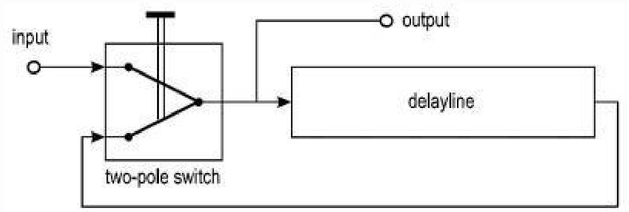
There are two possible modes when a single carrier is modulated by a single modulator. The first mode is to create formant areas in the audio spectrum that will stay on the same spot in the spectrum when different notes are played. In this mode the amplitude of the modulating signal is kept constant over the keyboard range. The second mode is to keep the resulting waveform constant over the keyboard range, just like how a sawtooth is the same shape for each key. This mode also creates a formant structure in the sound, but the formant areas glide along with the pitch. This is similar to the keyboard tracking of a filter, the first mode is like no tracking and the second mode is like full tracking. In the FM Trk mode the amplitude of the modulation signal is scaled to the keyboard pitch, higher notes will increase the amplitude as the deviation must increase, e.g. when the sweep spans 110 Hz for a 440 Hz pitch it must increase to span 220 Hz for a 880 Hz pitch. On the G2 oscillators these modes are named FM Lin and FM Trk, where FM Trk is the full tracking mode. The scaling for the tracking mode is conveniently built in on the FM input on the oscillators. For non-sine waveforms it is often best to choose for the FM Trk mode to prevent an unrealistic nosey effect in the timbre when playing up and down the keyboard. In fact, the FM Trk mode can best be seen as a way to get a freely shapable steady waveform that can later be filtered, just like how one would use the standard waveforms. Technically the difference between FM Lin and FM Trk is 'fixed formant' versus 'fixed modulation index'. But what is more important is that when the modulation depth is increased in FM Trk mode it will increase the brightness of the sound with a pleasing 'buzzy' type of timbral change, turning the modulation depth knob into a control similar to the Cutoff on a filter. The Phase Mod input on the OscPM is always in FM Trk mode.

The only thing that is now left to be chosen is the waveforms to be used for the carrier and the modulator and a suitable detune ratio between the two oscillators. On the modulator any waveform can be used, but for the carrier it is best to avoid waveforms with flanks, like the sawtooth. The sinewave and the triangle wave are good choices for the carrier. Using the pulse wave on the carrier will give a harsh sound, which can actually be quite nice, but should be treated with care. At the modulator side it is especially the pulse wave that is very suitable, as it will have the effect of alternatingly change the slope direction of the waveform, which works especially well on a triangle carrier waveform. PWM modulation on the modulator is also a nice effect that works out very well. Using the OscShape as the modulator and modulating its waveshape gives even more possible waveforms.

Capturing and looping audio

Delaylines

The Nord Modular G2 features audio delayline modules that can be put to good use for on the fly capturing and manipulation of audio loops. The most versatile G2 delayline module for this purpose is the DelayQuad module. It features four tap outputs that can all be set to a different delay time in respect to the input. Maximum delay time is 2.7 seconds, which is equal to about a bar of audio at 90 BPM. Which means that if a full bar must be captured the tempo must be 90 BPM or faster. If the tempo of the bar to be captured is faster than 120 BPM the delayline module can be set to use 2.0 seconds of delay, which saves some delay memory that can be used by other modules, like a reverb or a pitch shifter module. A standard G2 uses four DSP chips, each with its own independent audio memory, so a standard G2 can capture four separate bars of audio at 90 BPM and faster. An expanded machine, which has eight DSP chips, can capture eight separate bars.



Recirculator principle.

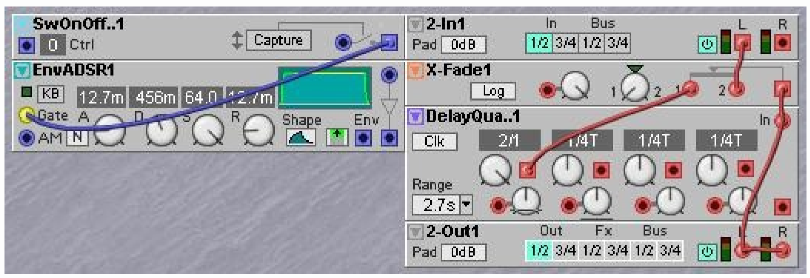
The principle

On a vintage three-head reel-to-reel taperecorder, like e.g. the once extremely popular Revox A77, the tape first passes the erasure head, then the recording head and then the playback head. But it was not uncommon in the experimental tape studio to change the order of these heads by remounting the playback head first, then the erasure head and then the recording head. Mounting the heads in this order offered a new possibility to continuously overdub the audio on the same tape with only a single taperecorder. The tape first passes the playback head and outputs the audio on the tape to the playback output of the recorder, after which the audio is erased on the tape. The tape output signal is mixed with new audio material and then rerecorded on the tape at the record head position, which results in a build up of audio layers on the tape when e.g. an endless loop of tape is used. The method was already in use in the 1950's in the WDR studio's on Köln, where Stockhausen recorded his tape compositions. This same technique was later used in tape echo devices like the CopyCat and the Roland Space Echo. When the idea is to create a loop that must loop at a continuous volume and not die out like an echo does, the technique is often referred to as audio recirculation and a device that can do this is named a recirculator.

A taperecorder is not an ideal machine for this recirculation technique, as on every rerecording of the audio material the signal quality slightly decreases, and the audio will eventually either drown in noise or produce an overly saturated sound. But by using digital memory the audio can loop forever without any degradation. In the preparation for this article a two second loop on the G2 was made to recirculate for two days, and still sounded as fresh and crisp as when it was captured.

The basic recirculator patch

The basic patch uses a two input switch on the input of a delay memory. One input of the switch is connected to an audio source and the other switch-input is connected to the output of the delay line. If the switch is set to the audio source, this source is connected to the delayline input and audio will flow into the delayline. When on a certain moment the switch is toggled to the other position, the output signal of the delayline is connected to its input. This will cause the audio contents of the delayline to loop endlessly, until the switch is toggled to the audio source again. By connecting a 'momentary switch' module to the control input of the two-pole switch module, and assigning this 'pushbutton' module to a G2 frontpanel pushbutton, the capturing can be easily controlled by this frontpanel pushbutton.



Single loop patch.

Setting the loop length

The length of the loop will be exactly equal to the delaytime setting of the delayline. Which brings us to the issue of how to control the delaytime and so the loop length. On the G2 it is easy to control the delaytime automatically with the G2 Masterclock. To do so simply set the DelayQuad module to Clk mode with the Time/Clk button on the module. The Masterclock and so the loop time can now be conveniently controlled from the G2 frontpanel. When a Midiclock signal is sensed on the Midi In connector and the G2 is set to receive Midiclock, the delaytime will automatically adjust itself to the tempo of the incoming Midiclock signal. But this must be an absolutely stable Midiclock signal and not all Midi sequencers and sequencer programs produce stable Midiclock signals. It is more reliable to actually use the G2 as the Midiclock master and sync the other devices to the G2 Midiclock. This will guarantee absolutely stable delaytimes on the G2 delay modules.

When the DelayQuad module is set to Clk mode the delaytime of each tap can be set to a subdivision of the BPM tempo. To capture exactly one full bar of audio at the masterclock BPM setting the subdivision must be set to 2/1, or the knob fully open.

Outputs on a recirculator

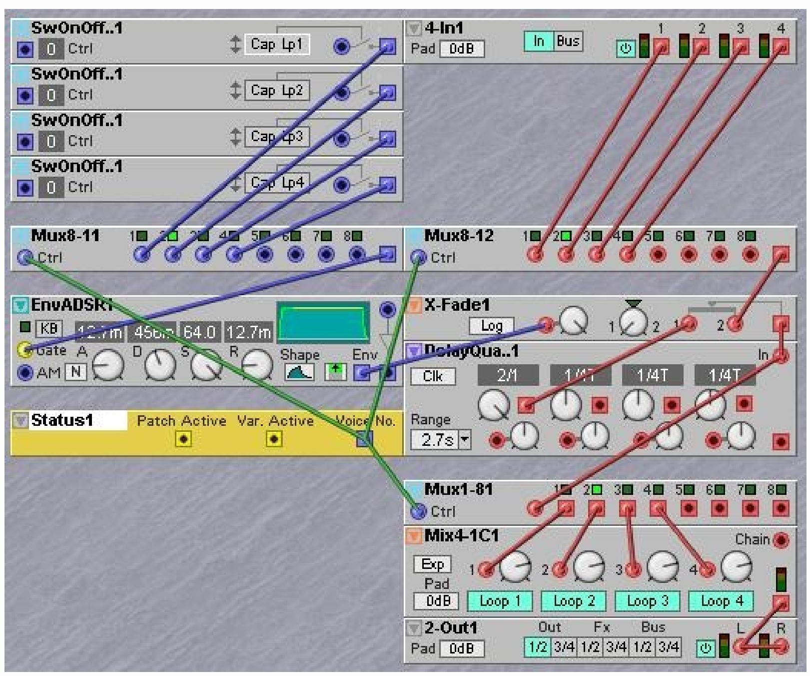
There are two main points where the audio can be tapped from the recirculator, one point is at the output of the two-pole switch and the other is at the output of the delay line. If the output is taken from the switch output the output of the recirculator will immediately output the incoming audio when the loop is set to capture, meaning that when pressing the capture pushbutton you will 'monitor' the input signal while capturing, until you release the capture button. If the output is instead taken from the output of the delayline and the capture button is pressed, you will not hear the input signal until the next bar, as the audio first has to pass the delayline, which takes a while. Many times you would probably want to hear the new audio immediately, so the output of the switch is the most often used output of the recirculator. There is one situation where you will have to use the output of the delayline and not the switch. This is necessary when the output of the delayline is fed into some extra effects and you want to create an extra feedback loop where the loop signal plus the effects are mixed together with the audio input signal. In this case the audio from the currently looping signal can only be taken from the delayline output. This is definitely a more advanced and a bit tricky way of looping, so for now start by using the output of the switch and experiment with the delayline output only when you fully understand the looping mechanism. The other three outputs of the delayline can of course be used to tap the audio as well.

Soft switching the capturing

When a two-pole switch is used on the input of the delay module, the toggling of the switch will most probably introduce a click. To avoid this click it is better to use a crossfade module and control the crossfade position with an ADSR envelope with a short attack and release time and full sustain level. This ADSR envelope can be controlled by a 'momentary switch' module that is assigned to one of the G2 frontpanel pushbuttons. One of the outputs of the QuadDealy module is connected to the first input of the crossfader module and the audio to be captured is connected to the second input. The crossfader knob is set fully to the left, and the ADSR envelope is connected to the crossfader modulation input. If the modulation amount control knob is set fully open, the crossfader will act as a soft switch where the attack and release times set the soft edges of the switching. There is another advantage when using a crossfader instead of a toggling switch, as when the modulation amount control knob is set half open the audio will not be replaced, but instead be mixed to the recirculating output of the delayline and the loop can be made to build up with additional layers of audio.

Multi-track looping

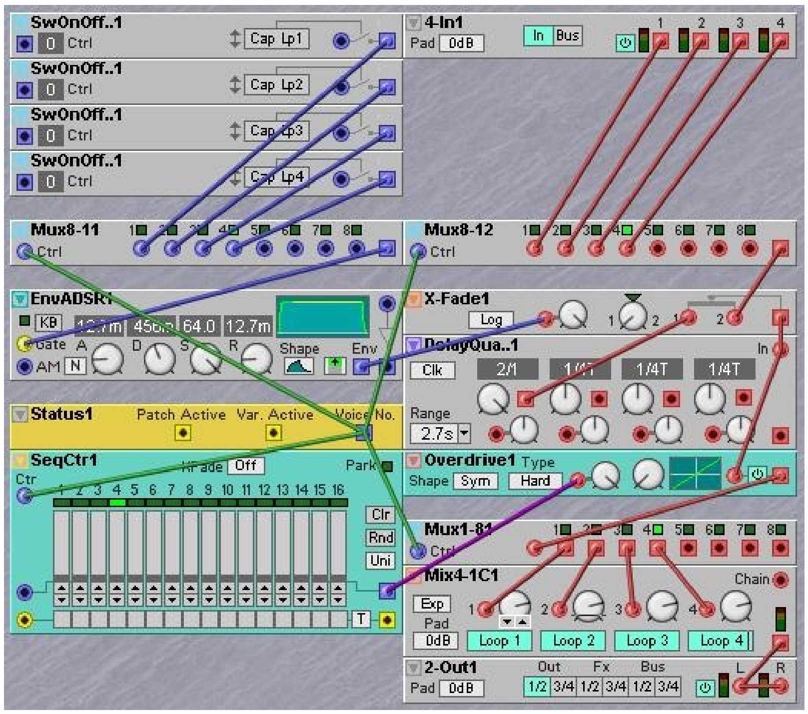
The next issue is how to get more simultaneous loops, like when using the four tracks on a four track taperecorder loop. When the previous patch example is set to four voice polyphony there are in fact four parallel loops, as each voice will in fact be an independent loop. The point is now how to discriminate between the four voices. In other words, how can only a particular voice be forced to capture audio while the other loops just keep on looping. To be able to do this the Status module must be used. This module has an output named 'Voice No.' and this output will generate a value that is different in each voice. This output can be used to directly control the Mux multipole switch modules. The idea is to use a Mux8-1 module at the input of the recirculator and a Mux1-8 module on the output of the recirculator and control the Muxes with the Status module. At the Gate input of the ADSR module is also a Mux 8-1 module and several pushbutton modules are connected to the inputs of this Mux8-1 module, like in the example. On a standard G2 you can make four loops, using the DSP memory of each of the four DSPs for each loop. On an expanded G2, with its eight DSPs and memory, you can make eight paralllel loops. In the example each loop has its own physical audio Line In input. The outputs of the loops are mixed with the four channel mixer, but you could alternatively route each loop to one of the four physical Line Out outputs. It is also possible to get the audio from patches in other slots by using the four G2 internal audio bus lines.



Four-track loop patch.

Individual voice settings

Effect modules can be patched after the mixer in which case all loops will flow through the effect. But it is also possible to use an effect like e.g. a distortion on each individual loop and have individual distortion settings for each loop. In this case only one distortion module is necessary and it can be placed between the output of the recirculator and the input of the Mux1-8 module. To give each loop its individual settings the SeqCtrl module can be used. By controlling the SeqCtrl module from the Voice No. output of the Status module the first four step sliders will actually provide the individual control values for the four voices. Just assign the first four sliders to frontpanel knobs and you have the individual control knobs.



Individual loop settings.

Conclusion

With this 'live sampling' technique of capturing audio into loops using this 'polyphonic recirculator patch', you can actually mix a whole live performance set with audio that comes from other slots in the G2, audio from samplers, drum computers, record players or a CD player, the Mic input or from the instruments played live by other musicians you perform with. It is a powerful live technique that the G2 can do quite well, though you will probably need some practicing to master it.

Creating evolving patterns

Shift registers

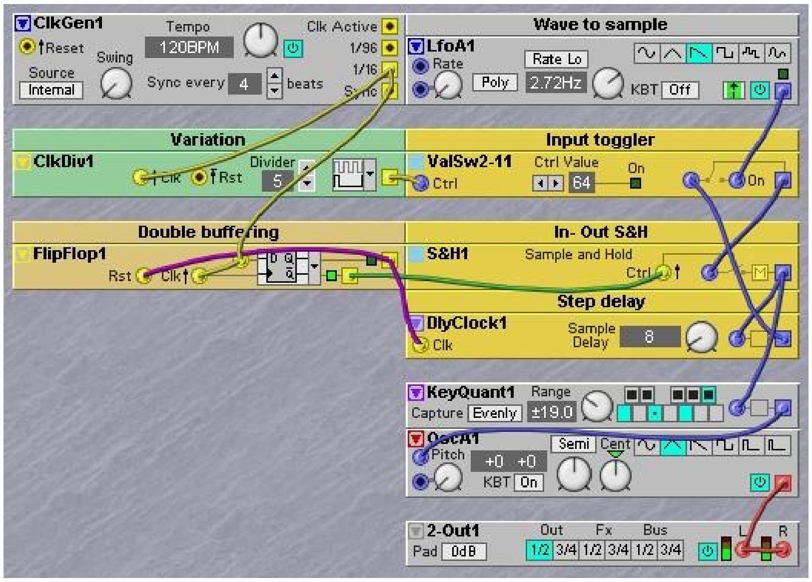
Shift registers are short delay lines using a clock with a variable rate to read and write a value. Imagine shift registers looking a bit like a pipe with an input on the left opening and an output at the right opening. On each clock pulse on a clock input a value will enter the pipe. After a certain amount of clock pulses that input value will appear at the output. So, it is like all input values are shifted in a first-in-first-out order through the pipe. This first-in-first-out order is the reason why shift registers are also named FIFO-buffers.

When using analog circuitry the most basic form for a shift register is a bucket brigade made with a number of capacitors that act as memory cells charged with a voltage level and switches that pass the charge from one capacitor to the next. In the seventies chips became available that held a long string of capacitors and switches on one silicon substrate, with lengths of up to 4096 steps. These bucket brigade devices, or BBD's, have the advantage that they can be clocked with a variable clock rate from as low as 10kHz to a maximum around several hundreds of kHz, which is perfect to create audio delays with delay times of up to several tens of milliseconds. Varying the clock rate creates a smoothly varying time delay, which creates a smooth frequency sweep on a sound with a fixed pitch. These chips were frequently used in flanging devices, chorus units, echo devices, resonators, comb filters and early implementations of pitch shifters. The chips had great disadvantages as well, like being noisy, having quite some signal loss from input to output and generating quite some harmonic distortion. Still, used in a well-designed schematic they could have a very lush sound with a typical 'analog' character. These days bucket brigade devices are replaced with digital memories using an A/D converter on the input stage and a D/A converter on the output stage. There is a very important difference between using BBD's and digital memories as the early bucket brigade devices have a fixed length and a variable clock rate, while modern digital memory designs in general have a fixed clock rate and a variable length. What this means is that delay devices based on digital memory must use elaborate interpolation to fight the aliasing created by the delay time being related to the fixed clock rate. This aliasing is especially troublesome when creating a smooth frequency sweep. If the sweep is very fast the digital delay line will also start to skip memory locations and thus loose information, resulting in degradation of the sound when the delay line output it is fed back to the delay line input. Having to compensate for these skips will make the anti-aliasing routine quite complex, as an average value of the skipped samples must be calculated and taken into account. In contrast, fixed length with variable clock rate delays do not suffer from dataloss as values are never skipped. All in all, both bucket brigade devices and digital memory delay lines have troubles of their own and it is not easy to create a proper digital emulation of analog effects based on bucket brigade chips.

Circular pattern buffers

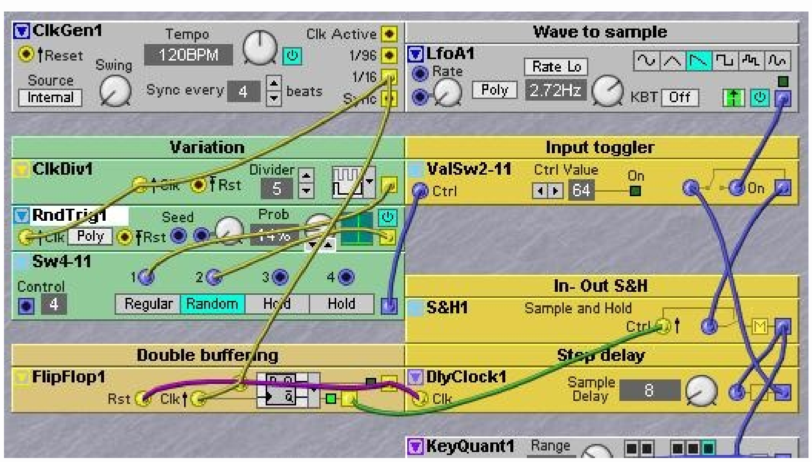
Short bucket brigades can be used as musical sequence and pattern generators that can create evolving patterns. Such pattern generators can be represented as a certain amount of sample and hold modules chained in series. Such a chain is usually named a shift register. Imagine there is a chain of eight sample and hold modules and they are all clocked with the same clock. On each positive edge of the clock signal the value in a sample and hold is passed on to the next sample and hold. To prevent that the value on the first sample and hold is raced to the output of the last sample and hold each sample and hold output must be buffered with an extra sample and hold that is clocked on the negative edge of the clock signal. This is essentially the same as what happens in a bucket brigade, where twice as many men are needed as the amount of available buckets to pass on the buckets. This technique of using the positive edge to clock values and a negative edge to clock buffers is sometimes named double clocking. An alternative approach to double clocking a bucket brigade delay line is to use an electronic multi-pole switch that is advanced to the next position on each clock pulse. E.g. a circular delay line of eight steps can be created when a single sample and hold circuit is equipped with e.g. an eightfold multi-pole switch, which on its turn is connected to eight capacitors. The trick is now to always advance the multi-pole switch to the next position on a clock pulse, but have a mechanism that can prevent the currently selected capacitor of being charged with the voltage on the sample and hold input. This creates a circuit that is equivalent to an eight-step clocked delay line where the output of the delay line is by default connected to the input of the delay line by a toggle switch. You can imagine that it now seems impossible to enter something into the delay line, but when the input toggle switch is toggled to input a value from e.g. an LFO during exactly one clock pulse period this value is entered into the delay line and will appear on the output each multiple of eight clock pulses later.

This circular delay line is a very powerful means to create repeating patterns where now and then one or more values in the pattern are replaced by other values. On the G2 it would look like this:

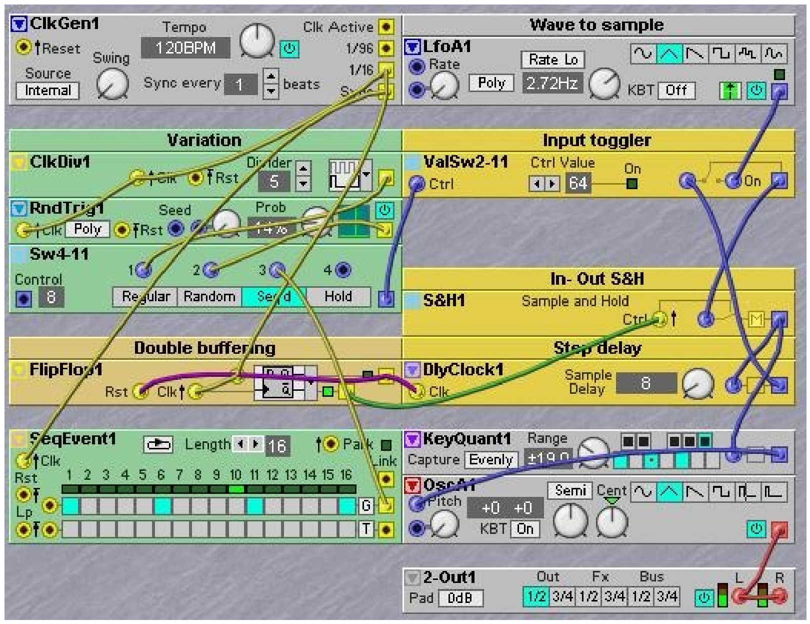


The circular delay line is made with one S&H; module plus a ClockedDelay module. By using double buffering the amount of steps will be equal to the amount of steps set on the ClockedDelay module. The two clock signals are generated by a FlipFlop module, which is used in a way that the S&H; module is clocked exactly one system sample after the positive edge of the clock signal coming from the ClkGen module. The reason why it is patched in such a way is to make sure that the input toggle switch is toggled to the signal input one system sample before the S&H; is triggered. In this way the working of the toggle switch is always reliable. When the FlipFlop receives a positive edge on both its Clk and D inputs it will pass on the high value to the Q output. The Q output will clock the DlyClock module and make it shift the pattern one step. On the next system clock the FlipFlop will find a high value on its Rst input and reset the Q output to low and the Q-bar output to high. Now the Q-bar output will trigger the S&H;, sampling he signal selected by the input toggling switch. If the input toggling switch selects the output of the ClkDelay module the value that is delayed eight steps will be entered into the ClkDelay again. But if the input toggling switch selects the output of the LFO module in the patch it will enter a brand new value into the pattern. The input toggling switch is controlled by a clock divider module that defines after how many clocks a new value is entered into the pattern. What you should do is rebuild the little patch in the example and play with it until you understand how patterns are built up. There are basically three parameters that define the build-up of a pattern. First is of course the length of the pattern, which can be set on the ClkDelay module. Second parameter is after how many steps a new value is entered into the pattern, which can be set on the ClkDiv module. Setting this value to 1 will make the circuit act like a normal S&H; module sampling the LFO waveform. Note that if the ClkDiv module is set to an even number and the dleay length is also an even number the odd numbered steps in the pattern will not be changed, as they will be skipped by the ClkDiv module. This means that in many case you would probably want an even value of steps and an odd value for the ClkDiv, or instead an odd number of steps and an even value for the ClkDiv module. In many cases you also probably want the number of steps fixed, like e.g. eight twelve sixteen or thirtytwo steps and variate the value on the ClkDiv module. The third parameter is the rate, waveform and amplitude of the sampled LFO. If the LFO rate is very slow in comparison to the ClkGen clock rate new values will enter in the normal way, but it will take a long time until these new values actually change to a higher or lower value. You need to develop a feel for how the ratio between the LFO rate and the clock rate works out. As a rule of thumb it is good idea to start with the LFO rate faster as the ClkGen clock rate and control the speed of change of the pattern with the ClkDiv module. But you are free to experiment and find what works out best musically for a certain musical purpose.

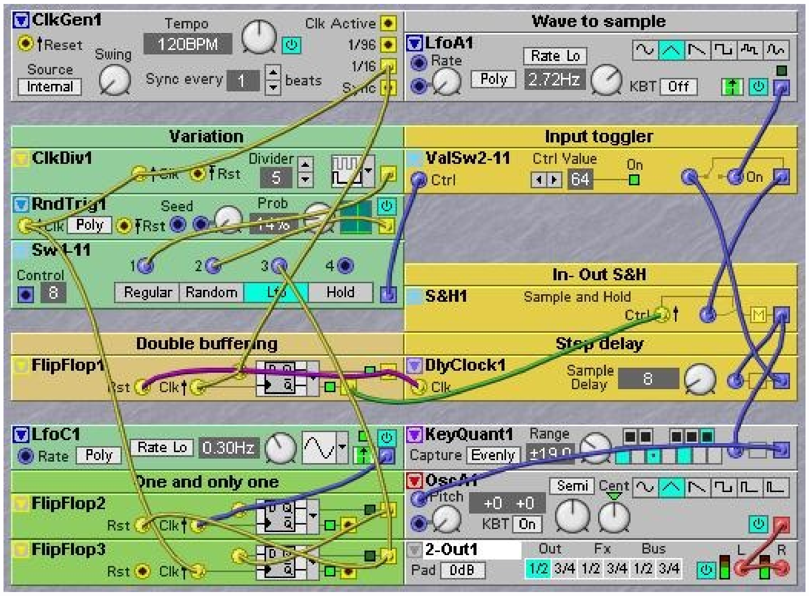
The beauty of this basic patch is that there are a lot of expansion possibilities that create an almost unlimited amount of way to build up, variate and build down sequences. E.g. the ClkDiv can be substituted for a RndTrig module, or a RndTrig is added and a two-pole switch is added to make it possible to select between a regular or a random rate of change in the pattern. By tweaking the Possibility knob on the RndTrig the average speed of change can be controlled between very fast at almost 100% and very slow at 1%. If set to 0% no new values will be entered into the pattern and the pattern will remain as it is. When a four-pole switch is used instead of a two-pole switch the third and fourth switch position can be connected 'to nothing' to cause the pattern to hold as it is. It would look like this:



Another interesting option is to use an SeqEvent module to define when a new value should be entered into the pattern. When the SeqEvent module is clocked from the ClkGen Sync output and the event bar is set to the G-mode it is possible to change e.g. whole blocks of four, eight, sixteen steps, etc. It would look like this:



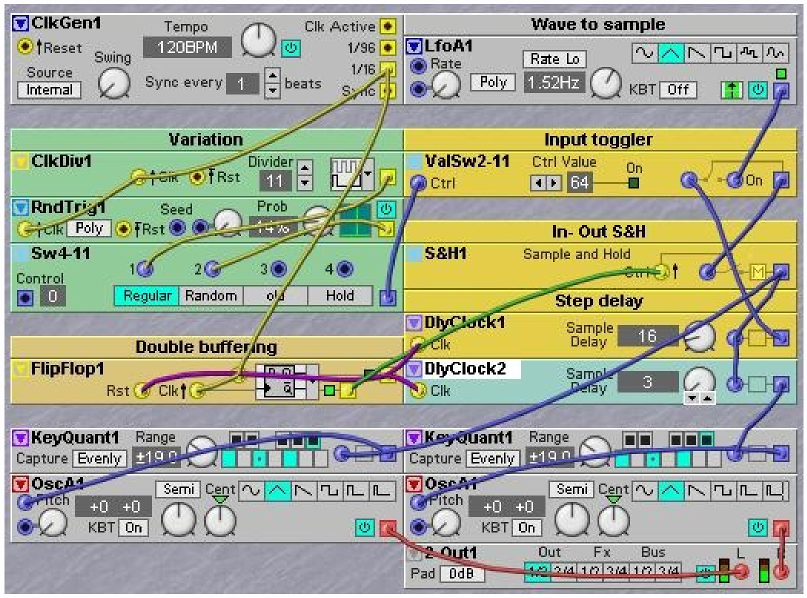
Another variation is to let a LFO decide when a new value is to entered. But this needs some extra facility, a little circuit that generates a pulse that is exactly as long as one step and is initiated by the LFO. Such a circuit is named a one-and-only-one and is made with two FlipFlop modules. The idea is to set a FlipFlop by a positive going zero crossing of an LFO waveform. The Q output of this first FlipFlop is monitored by a second FlipFlop that is clocked from the ClkGen module. If the second FlipFlop clocks a high from the first FlipFlop it will pass this high on to its own Q output. Noteh that this will happen on the positive edge of the ClkGen signal, so be in sync with the ClkGen module. The Q output from the second FlipFlop will also reset the first FlipFlop, so the when the next clock from the ClkGen comes it will clock a low from the first FlipFlop and set its Q output low. So, on the Q output of the second FlipFlop will be a pulse that is exactly as long as one step in the pattern. It looks like this:



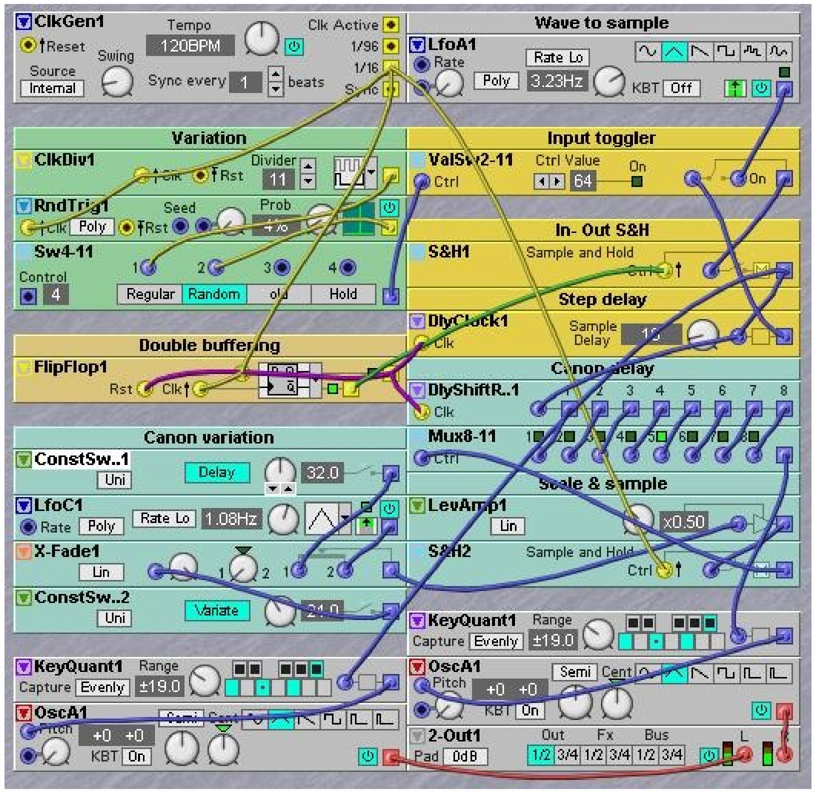
The LFO plus one-and-only-one circuit has the advantage that by modulating the LFO rate the rate of change in the pattern can be modulated in wild ways over a wide range.

Canon effects

This interesting effect is created by adding another clocked delay line in parallel to the DlyClock module. This second delay line will effectively delay the pattern by a number of steps and its output can be fed to a second oscillator module. As you can see in the next illustration it is quite simple to add this effect:



The next step is to create variation in the canon effect. By replacing the second ClkDelay module by an eight output DlyShiftRegister module and using an eight input MUX module the amount of delay steps of the canon effect can be varied with a control signal from e.g. an LFO. To get the change in the canon in sync it is necessary to use a S&H; module between its control input and the LFO output. This S&H; must be clocked from the ClkGen module. There is some extra modules needed to scale the LFO signal down and combine it with a fixed setting to keep it in the MIUX control range between 0 and 32 units. As you can see this is neatly solved by crossfading between a Constant module and the positive only LFO waveform, after which the result is scaled down by a factor of 50%. Then it is synchronized to the ClkGen by the S&H; module. It all looks like this:



The previous examples tried to show how to build up a pattern generating patch that can create varying pattern where there is some sort of a controlled amount of repeat in the patterns. The like of thinking that led from one example to another can be extended into virtually infinity. E.g. instead of sampling a LFO waveform it is also possible to sample sequencer modules with preset sequences. By having a several sequencer modules and dynamically selecting which one to use you can morph preset patterns into each other. It is also possible to add a second circular buffer on the output of the first one, fill this one one a moment that the pattern sounds really interesting and hold it there. Then you can use this second buffer to later fill the first one again with that previous interesting pattern. Just think of how you could patch such a system and try it. It is very well possible that you end up with something that is even more interesting.

By replacing the oscillator modules by MidiOut modules you can let other instruments, like e.g. a sampler, play the generated patterns. Or record the patterns into a MIDI sequencer program. This last option can be very interesting, as whenthe MIDI information has been recorded you can easily delete the less interesting bars and move the more interesting ones around to more fit the structure of a song.

Using the circular buffer principle on audio