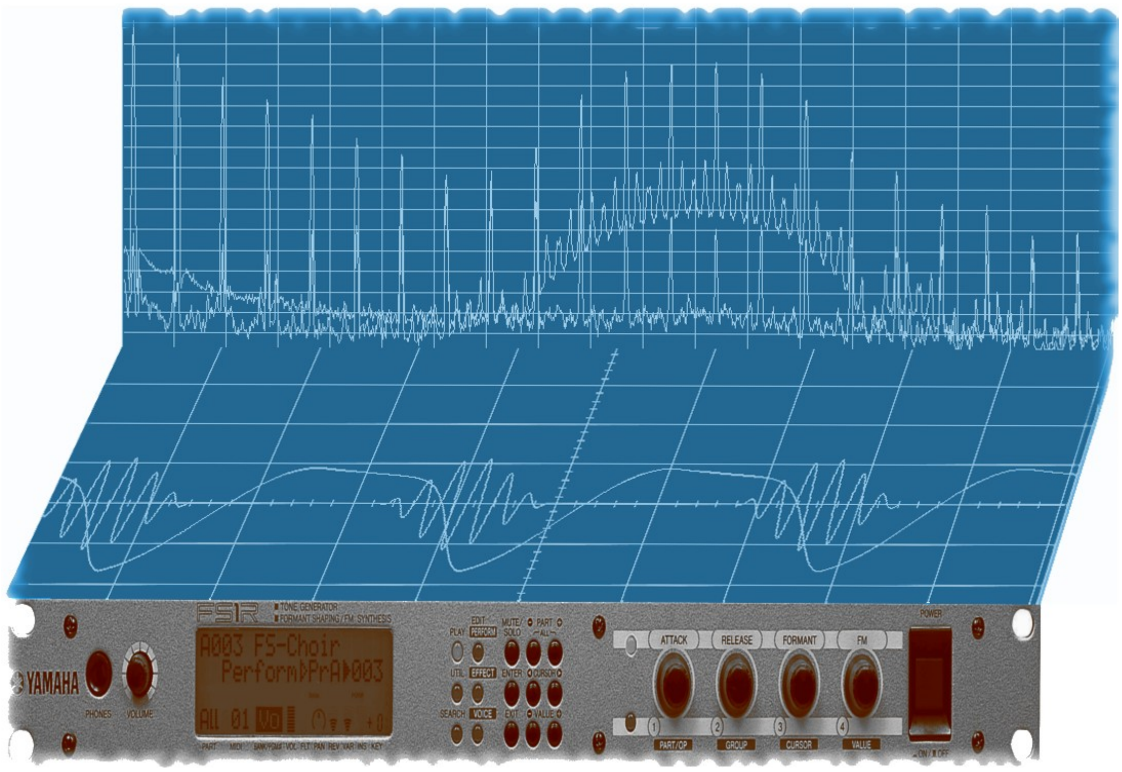
YAMAHA FS1R

Facts & Speculations

by

Michał Wiernowolski

This booklet is dedicated to an amazing synthesizer from the past.
You can treat it as a programming guide.
Basic understanding of sound synthesis and MIDI is recommended.



E-mail: mihalw@gmail.com

Contents

Review

Yamaha introduced FS1R in 1998 as a Formant Shaping/FM Synthesis tone generator. This is a hybrid digital synthesizer that provides formant modeling, frequency modulation and analog modeling. It also includes a digital effects section.

Formant Shaping is a unique feature of FS1R, so let's start with it. The FS synthesis models formants, which are peaks in spectrum of a sound. These peaks play an important role in recognizing sounds, most notably human speech. By generating and controlling formants it is possible to create sounds that mimic speech or singing.

Yamaha decided to provide FS in a frequency modulation environment. There are 8 voiced (pitched) and 8 unvoiced (noise) operators per single voice. The voiced operators can be selected out of 8 so called spectral forms (waveforms) and combined according to one of the 88 FM algorithms. The unvoiced operators are simply added to the mix. Up to 4 voices can be used to form performance in a Sampling&Synthesis fashion. This means that a single patch can provide up to 64 oscillators per note.

Formants are available as specialized FM operators. For this reason FS may be considered as a variant of FM. However FS not only sounds and behaves a bit differently, but it also requires quite a different approach to programming. The synthesizer also provides formant sequences, which is another distinctive FS feature. To conclude, FS should be considered as a genuine synthesis, that somehow overlap with FM, but neither is a subset of the other.

Both FS and FM are additive in nature (in a broad sense). Yamaha was generous enough to add on top of them analogue modeled multi-mode filters, which sound really good. This means basic subtractive synthesis is possible with FS1R. What is important patches with filters can be used polyphonically.

At the end of the signal chain there are three FX processors and an equalizer. There is quite a broad selection of individual effects. Their quality varies, but the reverbs alone make the FX section a valuable addition to the synth. The routing is fairly flexible thanks to send/return levels of individual parts and FX blocks. This gives many possible arrangements especially since many particular effects appear in more than one block. The FX section can be bypassed through the use of individual outputs which supplement the main outputs. It is also possible to direct each part to a separate mono output, which can be useful for external processing.

In general all the components mentioned above fit together really good. The FM concept seems well suited for hosting formant modeling. The filter can nicely polish FM/FS generated waveforms and the effects add the final touch.

MIDI Implementation

From performance point of view there is a lot to like. MIDI implementation is very comprehensive and the synthesizer can be configured to match your controller regarding channels, CC numbers etc.

In a few cases the transition is not smooth enough and one can encounter unwanted digital artifacts. Moreover it seems there is no support for 14-bit resolution which could provide some improvement here. Fortunately these are rare cases mostly noticeable in the FX section.

MIDI clock support is limited to formant sequences. This is quite a pity as many other elements like LFOs or delay effects could benefit from this kind of synchronization.

Almost all synthesis parameters can be controlled using short SysEx messages in real time. You can change for example operator's waveform or frequency of currently playing notes. This is very useful for sequencing and tweaking sounds.

There is also a full featured modulation matrix with multiple sources like aftertouch, modulation wheel and so on. These can be used to control up to 8 parameters from the synth engine.

With the exception of Program Change messages, which take really too much time, the response to MIDI is very fast. This make the synth a good choice for a wind controller for example.

User Interface

With the exception of formant sequences (at the frames level) everything can be programmed from the front panel. There are tons of parameters, so there is a lot of menu diving. Following FM tradition parameter names are occasionally bizarre. The menus are in general well structured, but they reflect somehow the internal architecture of the synth which is complex and confusing.

FS1R received a lot of criticism for its user interface. It is a bad idea to put a synth with so many parameters inside a rack mounted unit of this size. Tabletop design with many knobs and a big high resolution screen would suit it better. Unfortunately this would make the synth prohibitively expensive especially considering the year it had been released.

At some point I've noticed that the lack of a decent user interface is just an excuse for not getting satisfactory results. Let's make it clear - this synth is very difficult to program especially if you intend to build your voices from scratches. If you want fast results and hands-on experience look elsewhere.

Eventually, after working with the synthesizer for quite a long time, I must say that the user interface is not that bad at all, and much better than I initially thought. Finally, it is more of a problem how to make the program itself then how to enter it into the synth.

Sound

In the end it all comes down to the sound. I must say I've been highly impressed from the first time I've heard FS1R and it still amazes me after many years. The range of sounds this unit can produce can rival many synthesizers. And there are sounds hard to find elsewhere.

The synth can sound very smooth and homogeneous across the keyboard (unlike typical S&S synth where you can find cross points between samples). This makes it very good for monophonic (single note) sounds, like basses or leads. On the other hand very rigid harmonic structure suits polyphonic sounds well too. Consequently, and as expected from an FM capable synth, there are excellent electric pianos, organs and chromatic percussion kind of sounds. Pads are also very interesting. Vocal patches sound unique, somehow in the middle between human and machine.

The synthesizer comes with 1408 preset voices divided into 11 banks. Two banks use FS synthesis, the rest of voices is a selection from Yamaha DX7 library patches. (Note that FS1R can load and use DX7 patches.) These voices are used to build 384 preset performances organized into 3 banks. The third bank is dedicated to the guitar converter G50, but all of them can be used with any keyboard controller. At a later time Yamaha made additional sound banks including one dedicated to their WX wind controller series.

In the end it must be said that despite its little imperfections and idiosyncrasies FS1R is an outstanding synthesizer in the full sense of the word.

𝄞

A big thank you to Yamaha and the development team for making the FS1R

and to all users who greatly contributed over the years by providing

freely available software or information regarding this synthesizer

Programming Aids

Patch Editors

There are several editors available. I recommend **FS1R Editor** available from
https://synth-voice.sakura.ne.jp/fs1r_editor_english.html

It is free and really good. There are Mac&PC editions. The Windows version works fine in Linux/Wine.

For converting audio to formant sequences you can try **FseqEdit** available from
http://niff.home.xs4all.nl/fs1r/

Spectrum Analyzers

Oscilloscopes can be very helpful, but spectrum analyzers even more. Even experienced synth reviewers forget about the spectrum and present just the scope. Spectrum plot seems especially important for FM synths, which can easily generate high frequency content. You can even not be aware of potential problems like aliasing.

I highly recommend **Visual Analyser** available from http://www.sillanumsoft.org/
The 32-bit version works fine in Linux/Wine.

For formant analysis you can use **Praat** available from http://www.praat.org or
http://www.fon.hum.uva.nl/praat
Praat is available for many different platforms.

𝄞

Many thanks to all authors of the above mentioned valuable software.

MIDI Controller

A decent hardware controller with many knobs and buttons can be very helpful. Most controllers can be configured to send various CC messages including RPN/NRPN and this may be enough for basic playing. For more advanced use the controller should provide support for SysEx messages, what can substantially expand performance possibilities. This is not a very common feature unfortunately.

I've prepared special patches just for experimentation with 2- and 3-operator FM. The controller allows for direct manipulation of the most important parameters like waveform, frequency, level, bandwidth/resonance, operator mode (fixed/ratio/skirt) and feedback. This is possible only using SysEx messages.

For preparing controllers or entering data into software sequencers MIDI View function can be very useful. It is activated by double-clicking the enter button on the front panel and it shows MIDI message that needs to be send to setup parameters being currently edited.

Additional Information

When looking for additional information http://smasher.org/fs1r/ provided by Atom 'Smasher' is the best place to start. You may also take a look at https://en.wikipedia.org/wiki/Yamaha_FS1R.

Programming Methods

Due to its complex and powerful synthesis engine FS1R can be used in many different ways. Below you will find various approaches one can take to programming it beyond obvious FM, which is the basis of the synth engine.

Sampling & Synthesis

Don't worry! FS1R is not a rompler! We will begin with S&S as this is probably the least difficult method for programming the synth.

While not a rompler the FS1R can be used that way and the method is really valid. It is enough to treat the voices as waveforms on a S&S synthesizer. There are 1408 preset voices available. These can be used to build performances. Each performance can consist of 1 to 4 parts. For each part you can select the voice that will be used for playing.

Most parameters for individual parts work on an offset basis. This means they adjust voice parameters to create actual values used for playing without modifying the voices. Thus the same voice can be used in many performances and sound differently in each of them. Not all parameters are covered that way but the selection seems very reasonable. The following aspects of voices can be changed:

- Filter
- LFO
- Envelopes
- Pitch
- Portamento
- Mono/Poly mode
- Active note range
- Velocity response/range
- Volume
- Panning
- FS parameters (V/N balance etc.)

The performance contains also the following:

- Control matrix which defines response to MIDI control messages
- Formant Sequence configuration (which can be assigned to a single part only)
- Individual output setup
- Complete FX configuration

- MIDI channel assignments for parts

This offers a lot of possibilities. You can transpose, change tone, adjust envelopes, split and layer voices (both within note and velocity ranges) and process them using FX. As on a typical S&S synth. This can be really creative and with quality voices on board the results can be very good.

Frequency Modulation

I will try to give a brief explanation of what FM is. Keep in mind however this is a deep synthesis method so I will cover just the basics.

First there is so called **simple FM**. In this case we have two oscillators - carrier and modulator. The carrier gives the basic pitch and modulator affects the harmonic content of the sound. If modulator's frequency is in the LFO range the result will be vibrato. Modulating at audio rate creates additional stripes in spectrum within the audible range called **side bands**. The effect of modulation depends mostly on the relation between frequencies and levels of carrier and modulator. Both **harmonic** and **inharmonic** side bands of the carrier frequencies can be created (the former are multiples of the carrier frequencies, the latter not).

If you add amplitude and frequency envelopes you can change the sound over time. Note that the envelopes used for carrier affect the overall volume and pitch of the sound. Modulator's envelopes in general change the harmonic structure, but this causes change in volume as well.

The multi-stage envelopes let you shape the initial phase of the sound (transient) very precisely. This is important as transients play an important role in how we perceive sounds.

The oscillator with envelopes is called an **operator**.

In general the more complex modulator the more complex sound. However on first digital FM synths only sine wave oscillators where available. To overcome this limitation more than two operators could be used. This is called **complex FM**. This is were most people give up, as the number of combinations increases exponentially, and the results become more and more unpredictable. Note that with polymorphous waveforms like on the FS1R you will get a very broad range of sounds even without complex FM.

In complex FM there are many ways the operators can be connected, including feedback loops. The latter can be used to get sawtooth-like waveforms or noise. The particular layout (routing) of operators is called an **algorithm**.

The frequencies of operators can be expressed as ratios with respect to fundamental frequency. This lets you define frequency relations independently of the note you play. Sometimes you may want a fixed frequency rather, for example if you need to use operator as LFO. This is achieved by selecting fixed frequency mode.

In complex FM it is more apparent that the phase of oscillators matters as well. For example two operators out of phase can cancel each other and will have no effect. The phase of oscillator depends on the way it is triggered. If key sync is on the oscillator always starts from the same point of the waveform. While switching key sync will not remove the phase cancellation problems it will ensure consistent results for each key press.

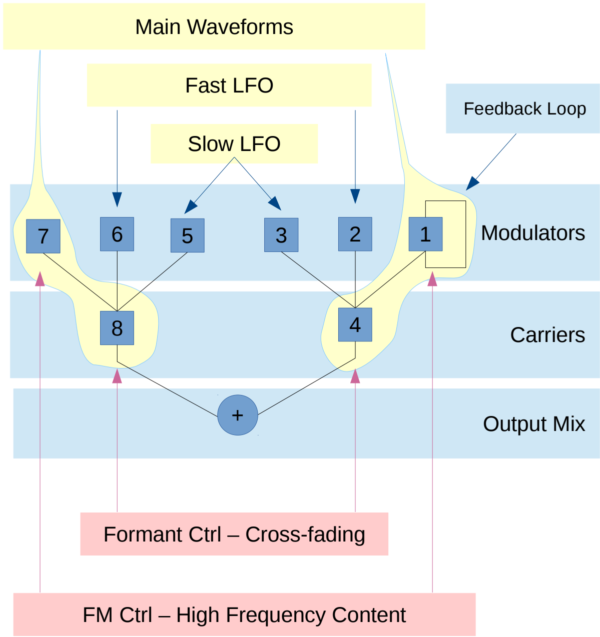
> With 16 operators (8+8), 9 different waveforms (8+1) and 88 algorithms the FS1R seems like the king of FM. If you want the emperor however look at the SY-99. It has only 6 operators, but with the ability to use PCM waveforms, user defined algorithms, three feedback loops and looping envelopes it is hard to beat. I am writing this not to start a competition for the ultimate FM synth, but to support my opinion that the FM engine on FS1R has been tailored specifically for formant synthesis. This is more evident if you look at the 8+8 concept with limited unvoiced operators functionality - these can not be modulated and if you link them to voiced operators they function rather as dependent sub-oscillators, forming together bands for formant sequences.

Strings

I will describe roughly string patch that demonstrates FM capabilities.

> Let us take a look at a typical analog string machines. These synthesizers are dedicated to creating string sounds. It seems there are two crucial aspects that decide of their superb sound.
>
> Firstly, there is the oscillator divide down architecture. It means the oscillator works at the frequency of the top octave and by frequency dividers delivers frequencies for lower octaves. Then with only twelve oscillators for the top octave we can get all the required pitches for the whole keyboard. Alternatively a single master oscillator working at the ultrasonic frequency can be used. In both situations this gives very uniform sound with coherent phase and full polyphony.
>
> Secondly, there is the ensemble effect. This effect uses two or three delay lines delivered by analog BBD chips. The delay lines are modulated by two LFOs and phase shifted to remove vibrato effect. With appropriate LFO frequencies the resulting chorus is very pleasant and brings movement to the sound.
>
> It's a pity to say at this point that none of the above can be directly emulated on FS1R and requires a dedicated device, like the Waldorf Streichfett or élkorus from Synthoma. There is no global phase synchronization (although oscillator key sync gives more consistent results), there are only 32 voices of polyphony and neither of chorus effects comes close to the aforementioned ensemble effect. This gives us just generic hints and not solutions.

Going back to FS1R... We will use algorithm 77 in the following way:



Operators 8+7 and 4+1 will provide main waveforms and should work in ratio mode. Selecting different setups for these pairs helps with phasing problems and gives timbre variety. I've chosen add1 oscillators for 8+7 and sine/res1 for 4+1 respectively but this is really a matter of taste. The resulting waveforms should have a lot of harmonics - this is achieved by high skirt values and algorithm feedback.

Operators 6, 5, 3, 2 work in fixed mode as LFOs with frequencies that mimic string machines ensemble units. It means around 0.3-6.4 Hz and 1.1-23 Hz for the low and high speed respectively (the fast LFO is usually 5-10 times faster). As the LFO is realized at the voice level and not globally we need a different strategy to avoid phase problems. This is achieved by free running operators (oscillator key sync is off) and frequency scaling which varies the frequency of LFOs depending on the note giving less repeatable modulation. The level of these operators must be adjusted to achieve required depth of modulation.

FM/Formant control can be used to change harmonic content or cross-fade the sound. If you set one of the carriers to octave below (ratio=0.5) cross-fading will balance between two octaves. This resembles selecting violin/cello sections of string machines.

To improve chorusing and spaciousness we can use two voices with slightly different parameters (LFO rates etc.) moved symmetrically off the pan center. A third voice transposed octave down working in mono mode with bottom note priority can provide bass register. These tricks let us provide a bit of string machine modeling regarding their functional behavior.

Aliasing

FS1R is a digital machine and produces the sound by mathematical calculation up to the converter stage (DAC). These calculations are performed cyclically in a discrete way at a certain rate. This rate not only limits available frequency range, but introduces usually unwanted phenomenon called **aliasing**. Basically it causes frequencies beyond half of the rate to be reflected at a different position in the spectrum. For example if we calculate at the rate of 40kHz the spectrum will contain 19 kHz instead of 21 kHz, which is beyond the half rate equal 20 kHz; similarly 18 kHz will be instead of 22 kHz and so on. This means that if we raise the pitch the reflected frequencies will fall which is a bit counterintuitive but can be easily observed.

The main problem with aliasing is that the reflected frequencies are practically not harmonically related to the pitch (as they bounce against a fixed frequency). Aliasing depends on ratios and levels of modulator as well as the frequency of the carrier. Higher values usually make aliasing more pronounced. Note that aliasing may become noticeable only beyond certain frequency, so its good to check your patch in different note ranges.

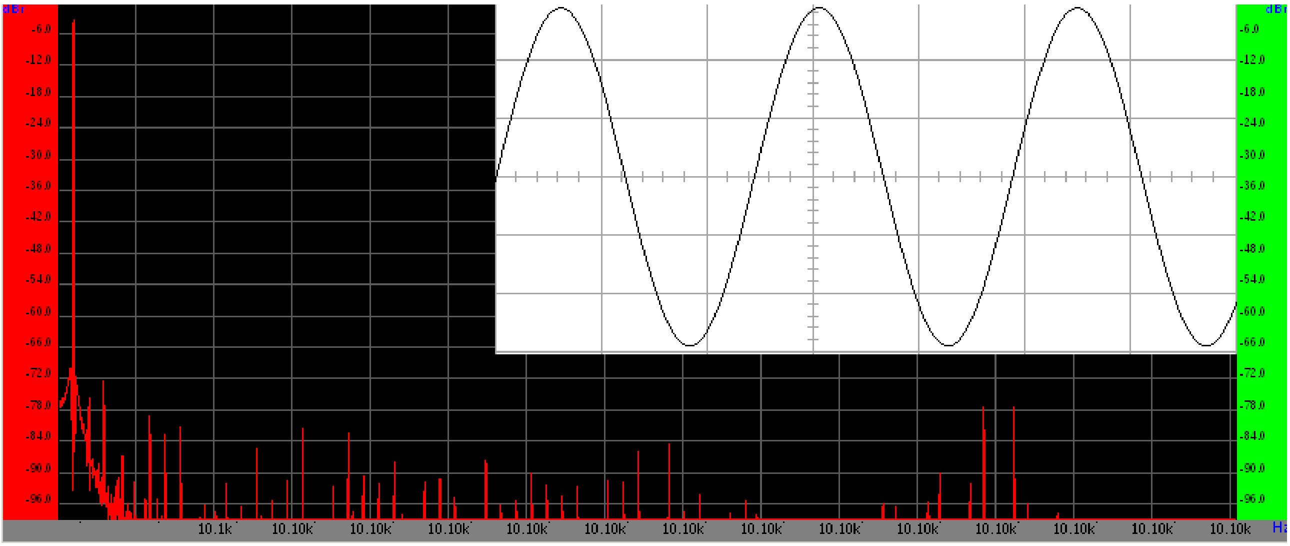
While aliasing is typically associated with sampling it works the same for waveform calculations in a digital synthesizer. To understand why it is so you can imagine that instead of an analog signal we are sampling some real mathematical function. Do not confuse however the calculation rate with the frequency at which the converter works – these are two different things although the values may be equal.

There is another aspect of FM that is somehow related to aliasing. The side bands created by modulation can extend above and below the carrier frequency. If we decrease the carrier to the frequency where the side bands would cross zero they will reflect with an inverted phase. Both harmonic and inharmonic frequencies can be created that way. The reflected frequencies can overlap with *legal* side bands; as the phase is inverted the resulting amplitudes will drop in these points.

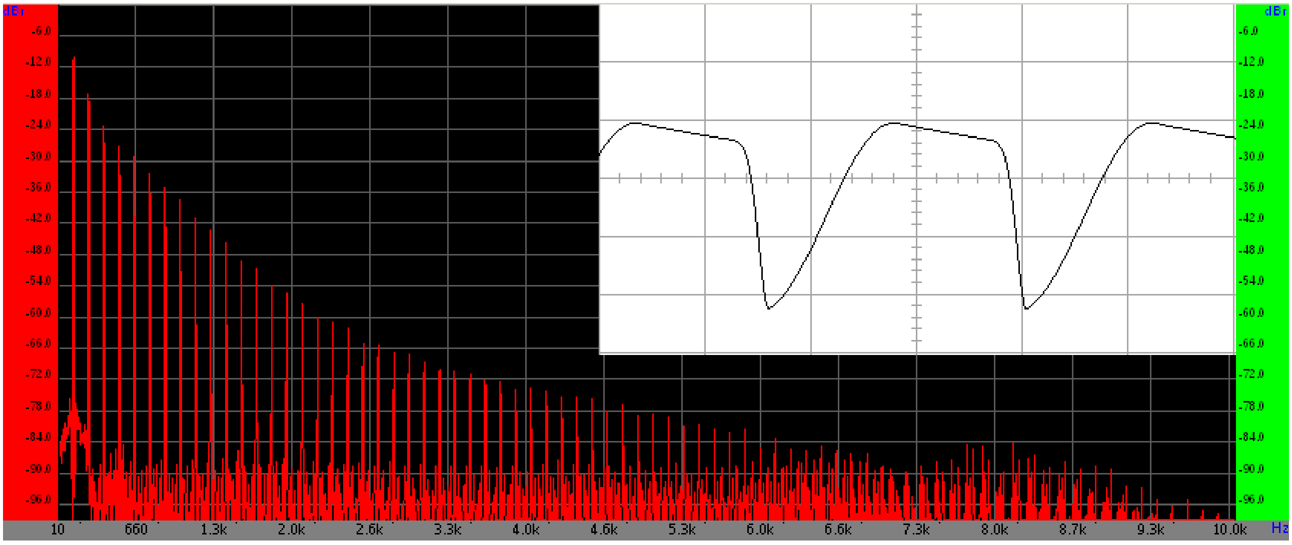
Spectral Forms

The result of modulation depends obviously on the shape of oscillators. FS1R provides 8 different types called **spectral forms**. This term may appear strange at first but frankly speaking it is really adequate as concrete spectrum shapes are modeled rather than waveforms. For simplicity waveform and spectral form will be used interchangeably.

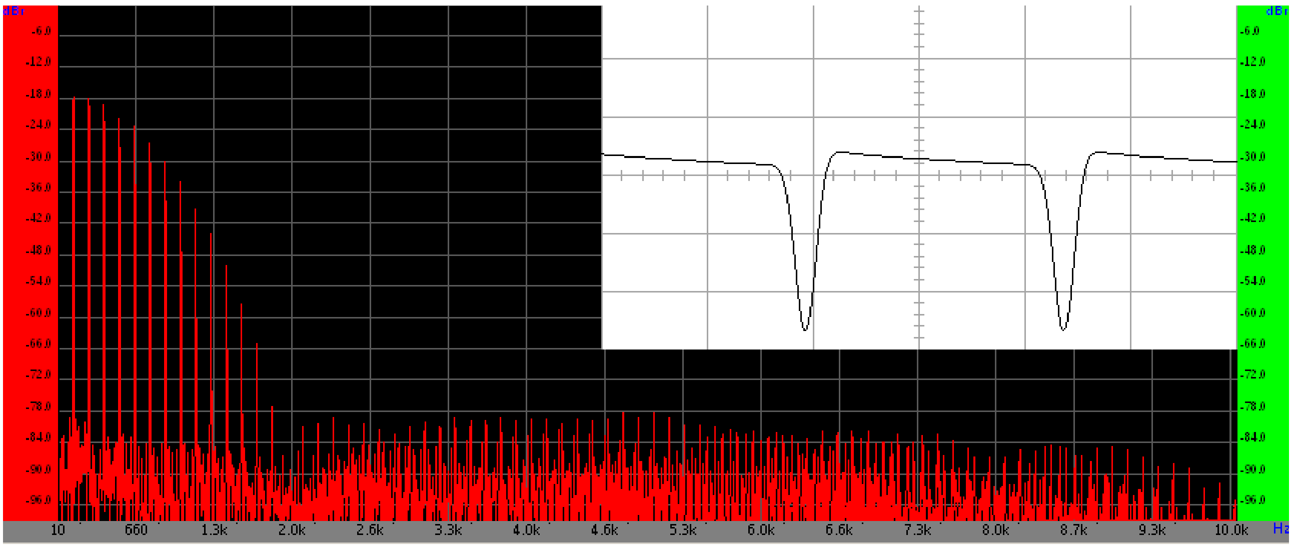
Below you will find snapshots of both spectrum and waveforms for comparison. All were prepared in the same conditions (the same note, skirt=5, band width/resonance=20, formant frequency=1761 where applicable).



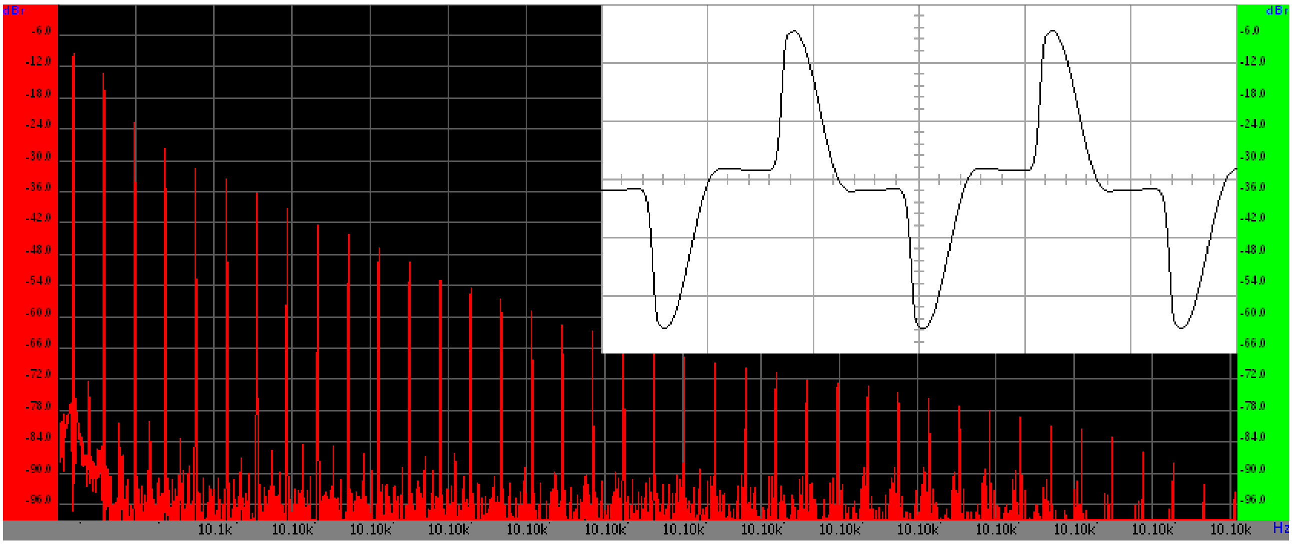
Sine



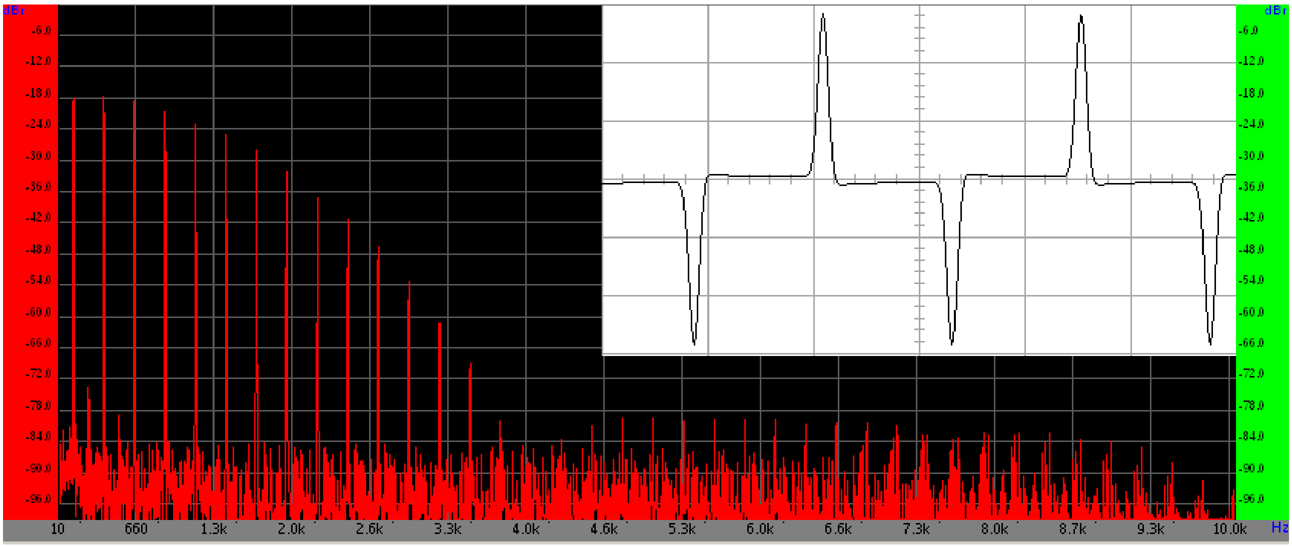
All 1



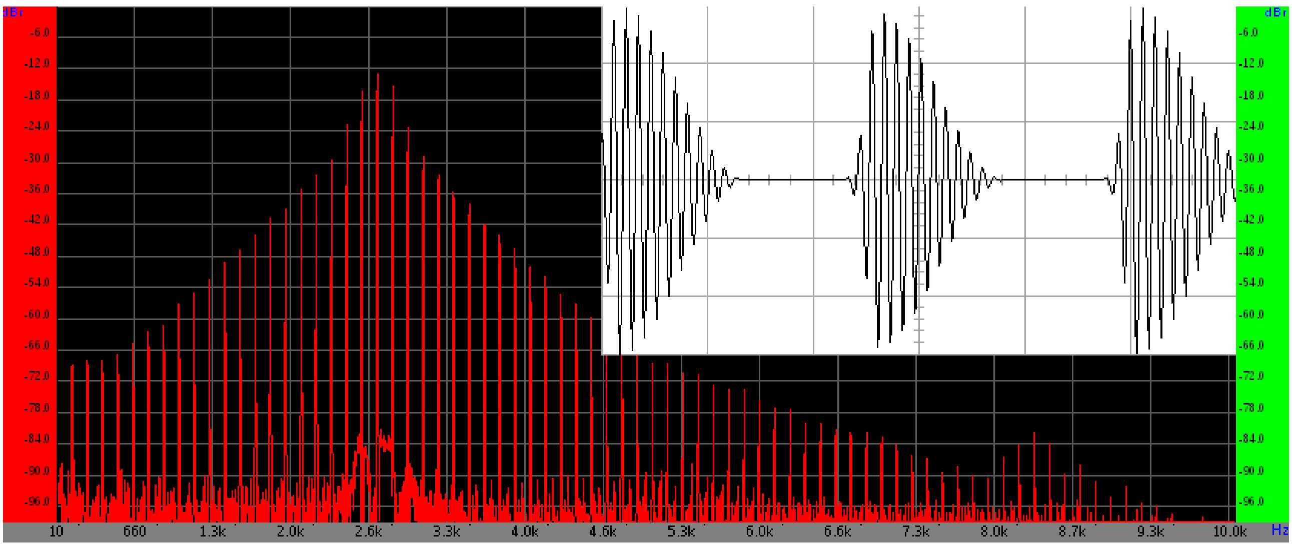
All 2



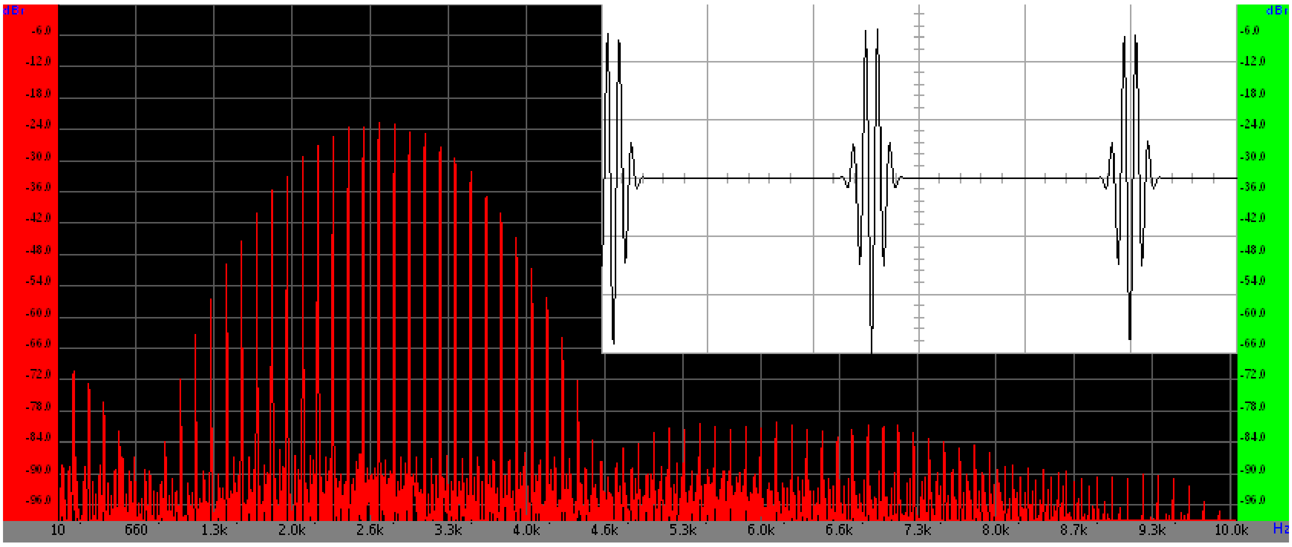
Odd 1



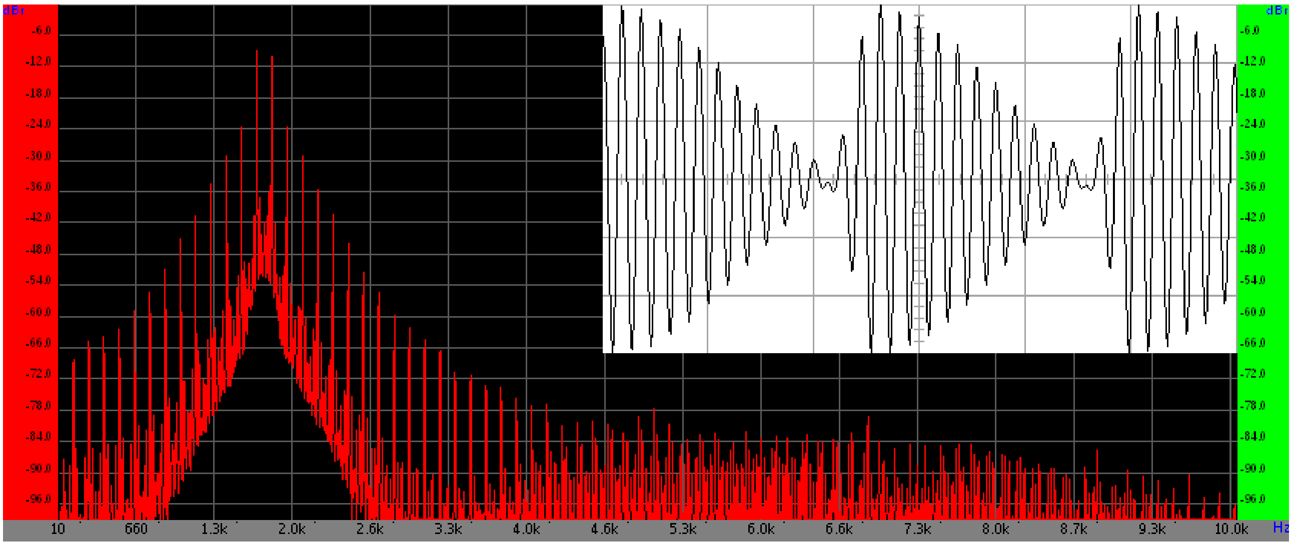
Odd 2



Res 1



Res 2



Formant

Additive Synthesis

Each periodic sound can be represented as the sum of sine waves. Thus by using many sine wave oscillators we can recreate basic timbre (harmonic contour) of the sound. By applying amplitude envelopes we can change the harmonic content over time as it happens for real acoustic sounds. This is the basis of **additive synthesis**. The problem is that to do it convincingly you need a lot of those oscillators, which makes programming very time consuming and tedious. This is why there are just few hardware synths, like the Kawai K5000 series, that use it extensively.

To be more precise periodic waves can be expressed as a sum of sine waves which frequencies are multiples of the **fundamental frequency** (the lowest frequency of the wave). These are called **harmonics** and we can easily get them on Yamaha FM synths which provide basic additive algorithms.

It may seem additive synthesis is a special case of FM. But in fact there is no frequency modulation taking place here. This means additive synthesis is a different method that happens to be possible on Yamaha FM synths.

Harmonics are not something restricted to additive synthesis. For example analog saw wave contains all harmonics with decreasing amplitudes. You can hear individual harmonics by sweeping the filter with high resonance.

Harmonics are equivalent to **overtones**. The latter does not include the fundamental in the series however. Therefore the first harmonic is the fundamental, the second harmonic is the first overtone etc. (Usually we don't hear harmonics as separate pitches. Overtone singers can isolate specific harmonics what makes an illusion there are two independent voices, but actually we hear separately the fundamental and one of its overtones.)

You may also encounter the term **partial**, which means any frequency component of the signal. So each harmonic is a partial, but not every partial is harmonic, as in general partials don't have to be multiples of the fundamental frequency (then they are **inharmonic** partials which appear in aperiodic signals). For completeness let us mention **pitch**, which will be used interchangeably with fundamental frequency, although it is the perceived frequency (height) of the sound rather which can be in general different.

The distance between subsequent harmonics of the same fundamental seems to decrease. This is because we perceive frequency logarithmically. To the contrary semitones, which are different from harmonics, seem to keep the same distance, while in fact they keep the same frequency ratio not the difference.

How many harmonics is enough for additive synthesis? This depends on the pitch you want to use and the timbre of the sound. For the theoretical audio range 20 Hz-20 kHz and the pitch 20 Hz we need 20000/20=1000 harmonics. But this is just the upper limit, in practice much less harmonics will suffice as normal (single) instruments do not fill the whole spectrum. For example the violin covers the spectrum up to around 10 kHz with the lowest pitch at about 196 Hz. This means we need no more than 51 harmonics in that case.

Setting up Harmonics

If you take a look at algorithm #1 you will see no modulation present except for operator #1, which provides feedback. Just keep the feedback at 0 and forget about it. If you set the frequencies to 1.0-8.0 respectively for subsequent operators you will get 8 first harmonics. The operators should be set to ratio mode and use sine waveforms. By using ratio mode we keep the harmonic structure relative to the fundamental frequency which depends on the note you play. This way we get similar sounds for different notes.

There is no need to select consecutive harmonics. By setting proper ratio we can choose up to the 57th harmonic (this is the highest available integer ratio). Note that getting exact harmonics above

31st is a bit tricky as they are not available directly through course tuning (you need to try different course and fine tunings until you find proper integer frequency ratio).

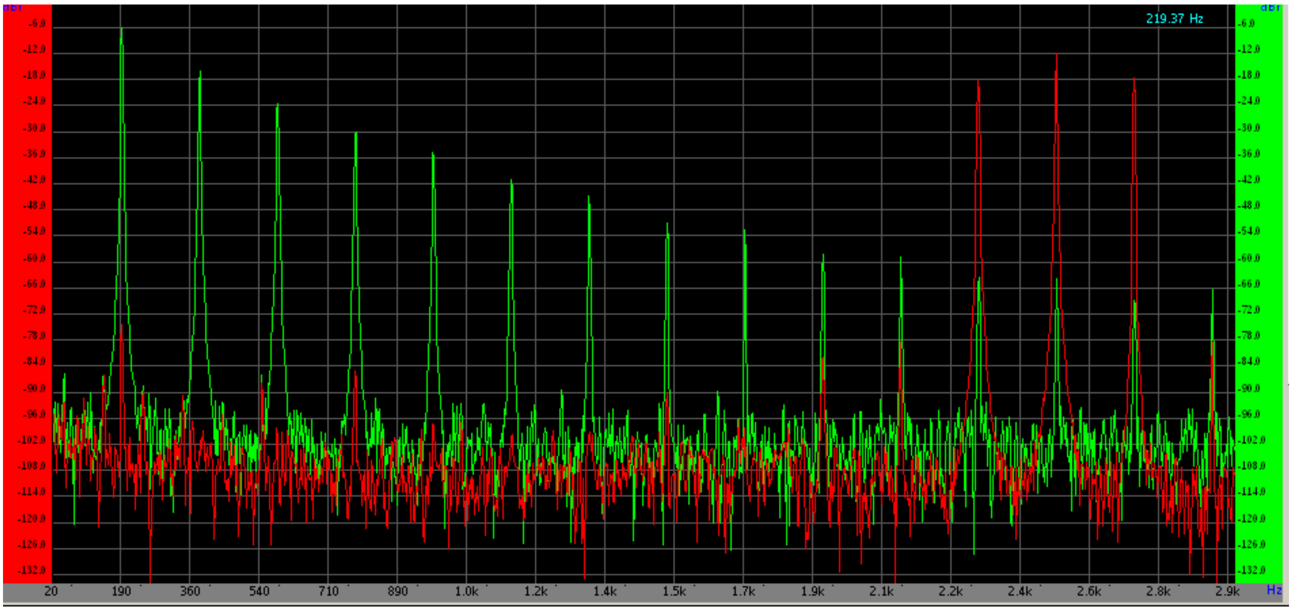
There are four parts on FS1R, so it is possible to create patches with 32 harmonics. This is quite a lot for a synth not dedicated to additive synthesis. Beside setting the frequencies and levels of all operators you need to program the amplitude envelopes. There are 10 parameters per envelope so this gives roughly 384 numbers to set after initializing the voices, which should make you realize how laborious additive synthesis can be.

Alternatively to sine waves you can use res1 or res2 oscillators. These generate selected consecutive harmonics. You can get higher ordered harmonics that way.

Resonance	Skirt	Res1	Count	Res2	Count
0	0	1,2	2	1,2	2
1	0	1,2,3	3	1,2,3	3
2	0	2,3,4	3	2,3,4	3
...	0	...	3	...	3
99	0	99,100,101	3	99,100,101	3
50	1	48-54 ~ 45-57	7~13	49-53	5
50	2	*	19~21	...	9
50	3	*	27	...	13
50	4	*	~41	...	19
50	5	*	~51	...	27
50	6	*	~75	...	37
50	7	*	~87	...	47

The table above contains results of a rough examination of the spectrum. The 'count' columns present the number of harmonics in the main band around the center frequency. The results are not fully reproducible especially for res1 with higher skirts (marked with *), where the spectrum becomes less symmetrical.

With the use of res1/2 oscillators single voice can generate much more harmonics depending on the skirt value. Obviously you loose the ability to control individual harmonics as a single operator generates a group of them. This may be an advantage though as there are less data to enter. To avoid phasing problems the groups of harmonics should rather not overlap. You can use the table above to get a little orientation which harmonics are generated depending on resonance and skirt settings.



Red - res1. Green - all1.

If this is not enough you can use unvoiced operators as additional harmonics. To do this you need to set the bandwidth to 1 and resonance to 7 (maximum). Leave skirt at 0 and do not set the bandwidth to 0, as this gives unpredictable results (unfortunately bandwidth =1 gives a bit 'noisy' sine). To follow the notes you need to set the frequency mode to link with fundamental frequency (linkFO) and transpose the operator accordingly. Note the transposition is in semitones. An alternative approach may be to set the operator's mode to normal and frequency scaling to 99. Then the frequency will follow the note played (but not the pitch bend messages).

Modeling Drawbar Organs

Drawbar organs can be considered as a simple additive synth. It may be a good exercise to implement it, at least partially.

The sound on a tone wheel organs can be changed using drawbars. These correspond to specific harmonics. Note that the fundamental frequency of the Hammond organ's is the so-called sub-fundamental, denoted by 16'. The concept of sub-harmonics somehow diverge with additive synthesis, but in FM this is quite ok. The sub-fundamental and sub-3rd will have 0.5 and 1.5 ratios respectively.

At this point it may seem all we need is 9 sine waves per note set up harmonically. The reality is much more complex. There is a mismatch between true harmonic ratios and equal temperament ratios. This is to keep semitone ratios constant, which in turn simplifies transposition. This should be taken into account to minimize tuning problems. According to http://www.electricdruid.net/index.php?page=info.hammond the Hammond tuning differs from both the true harmonic and equal temperament. But there are more aspects of the Hammond sound like the key click, harmonic leakage or the chorus/vibrato effect. I was thinking of what aspects of the sound and behavior of the organs matter most for me, then what is possible on FS1R.

Let's start with harmonics. You need 9 of them. You can get them from two voices using voiced operators or from a single voice using 8 voiced and 1 unvoiced operator. The latter seems a better choice as it saves polyphony. Much more difficult is to control the levels of them in real-time. Unfortunately it seems impossible to do it using CCs. There is simply not enough control sources and destinations. This leaves only SysEx messages. Formally you set the level using Operator Output Level. There is a minor drawback to this approach, as the value is taken into account at the very beginning when receiving Note On messages. There is another parameter called Operator Attenuator which affects notes currently played, but it causes unpleasant stepping and the attenuation is not deep enough (only 22.5 dB).

All wheels on the organs are in sync. We can not exactly replicate that but it seems that key sync behaves more consistently, so I suggest to switch it on for all operators . As a benefit you can have the fastest possible attack. The key sync does not affect the release moment, so the release time must be adjusted to avoid clicking.

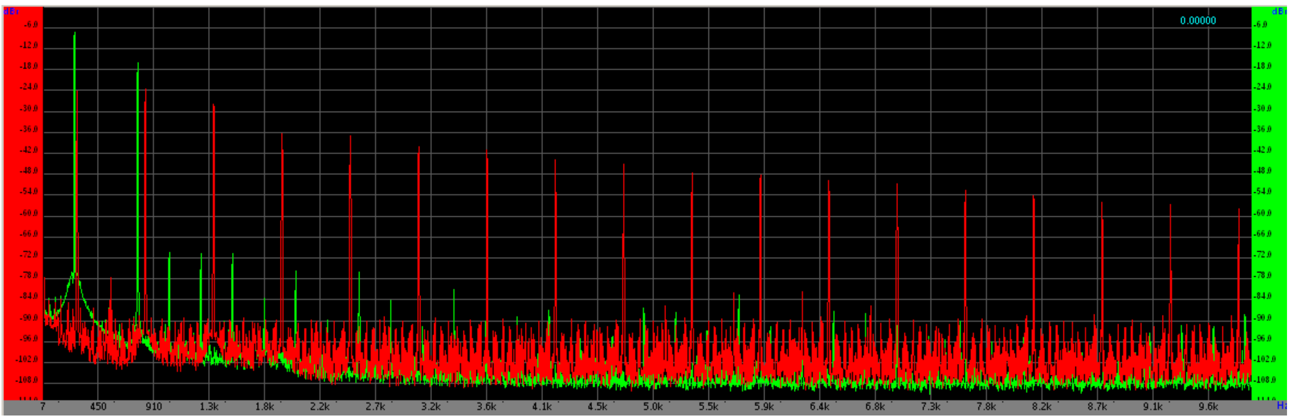
Finally you can add effects. Although FS1R provides a lot of routing options it may be difficult to recreate accurately the signal path of Hammond organs. For example 2WayRotary, which sounds slightly better than Rotary SP, is an Insertion effect. You can feed it to Reverb, but this is not how the real thing works, since the Leslie speaker is obviously at the end of the chain.

Virtual Analog Subtractive Synthesis

The core of subtractive synthesis is the filter for sure, but it needs something to work on, so let's start with oscillators.

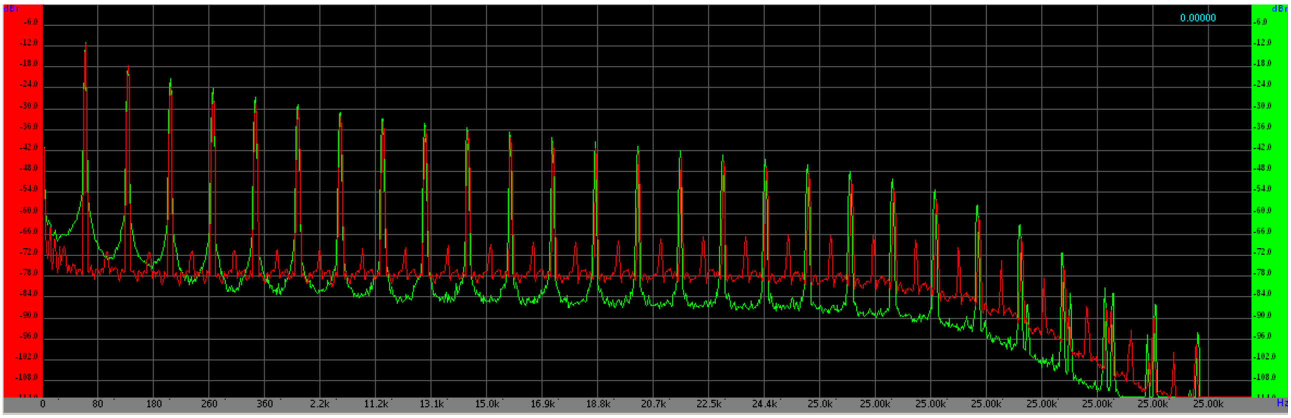
Basic Waveforms

For subtractive synthesis we need harmonically rich waveforms. These can be created in a variety of ways. First we will look at all/odd spectral forms which contain harmonics of analog saw/square respectively. To use them as saw/square replacements we need to raise the level of harmonics using skirt. As you can see below with skirt=0 only first and third harmonics have enough amplitude.



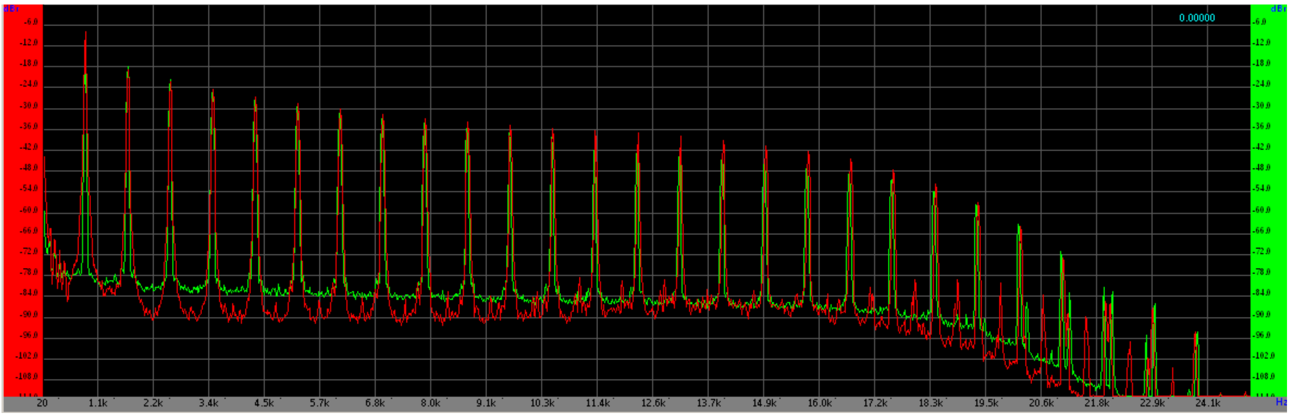
Red - odd1, skirt=7. Green - odd1, skirt=0.

The waveforms produced by all/odd do not resemble exactly saw or square but they sound quite similar. This is mostly because of phase differences between individual harmonics. Therefore on the oscilloscope they look very different, but if you look at the spectrogram they can be quite similar.



Red - all1. Green - true analog saw.

The sound created with all/odd is acceptable but a little gritty. You can create a cleaner saw by using feedback with formant carrier and sine modulator (see Earth Lead preset voice for details).



Red - FS saw. Green - true analog saw.

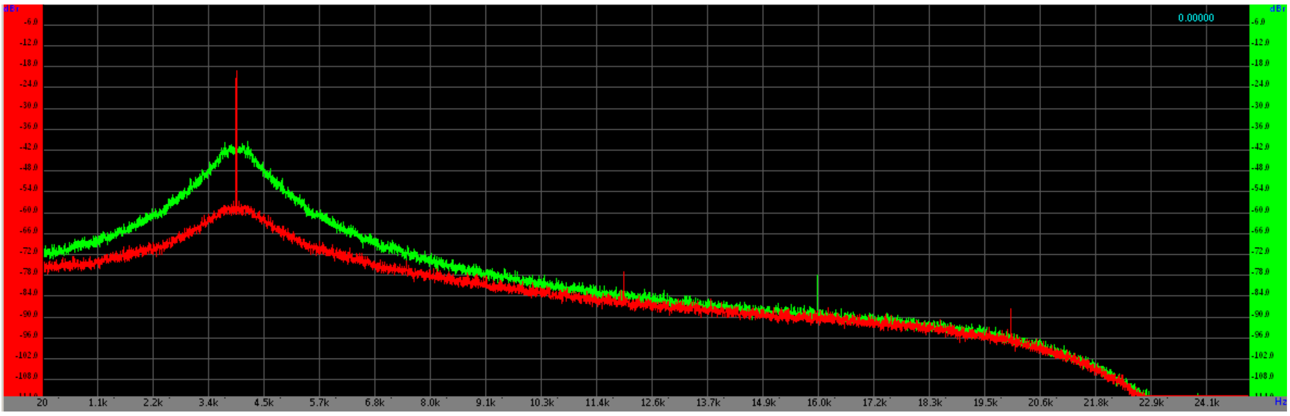
Finally note that classic analog saw, square or triangle waveforms or their spectral equivalents are not required. In fact *any* harmonically rich waveform can be successfully used for subtractive synthesis.

Noise Shaping

Noise can be a critical component for building percussive sounds or sound effects. It is an important part of analog synths as well as human vocal tract.

You can create noise on a FM synth just by using sine waves, but FS1R offers 8 unvoiced operators dedicated specifically for this, that let you control the output much more conveniently. You can treat unvoiced operators as bandpass filters that process internal white noise generator.

The main parameters of unvoiced operators are frequency, bandwidth, skirt and resonance. With narrow bandwidth and high resonance the oscillators start to self-resonate and can be used as additional pitched voices. Then there are envelopes for level and frequency. The frequency can be set to a fixed value (norm), linked to the fundamental pitch of the voice (linkFO) or to the frequency of the voiced operator of the same number (linkFF). The latter is only available if formant waveform is selected for the corresponding voiced operator.



Red - resonance=7. Green – resonance=0.

The unvoiced operators can be controlled in real time using FM/Formant CCs. As with voiced operators you can change level, frequency and bandwidth. This can be used to morph between pink and white noise for example (without additional filtering).

If that is not enough you can use formant sequences to animate the unvoiced operators. (This is possible for a single part only due to formant sequence limitations.)

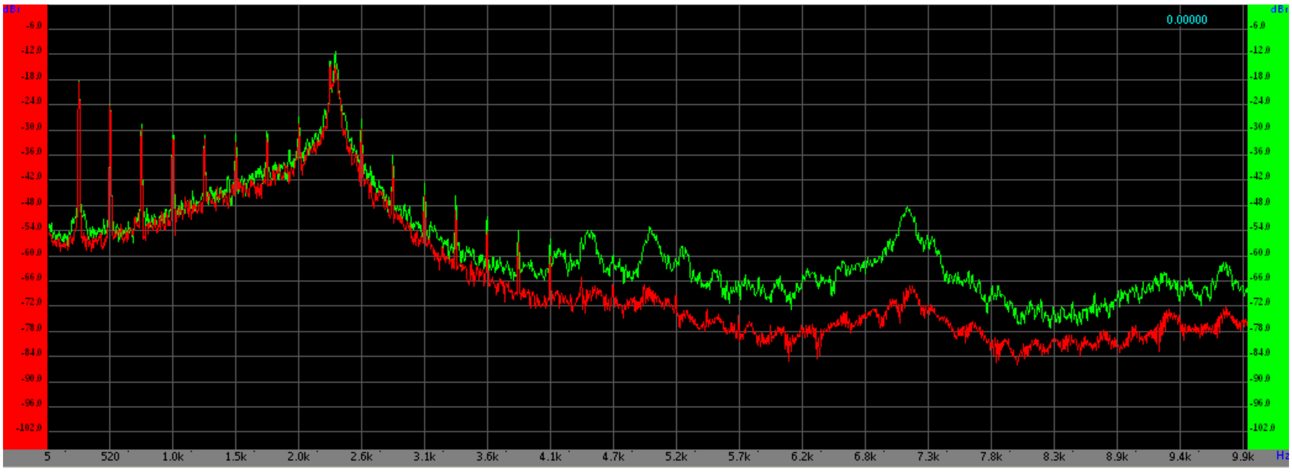
I've seen no other synth that provides so full-featured control of the noise generation alone. You can create any flavor of noise and use it as a source for subtractive synthesis.

Analog Modeled Filter

Yamaha equipped FS1R with an analog physical modeled filter. It is a multi-mode filter offering high-pass, band-pass, notch (band-elimination) and 3 low-pass modes with 12, 18 and 24 dB/octave slopes. The resonance works in all modes.

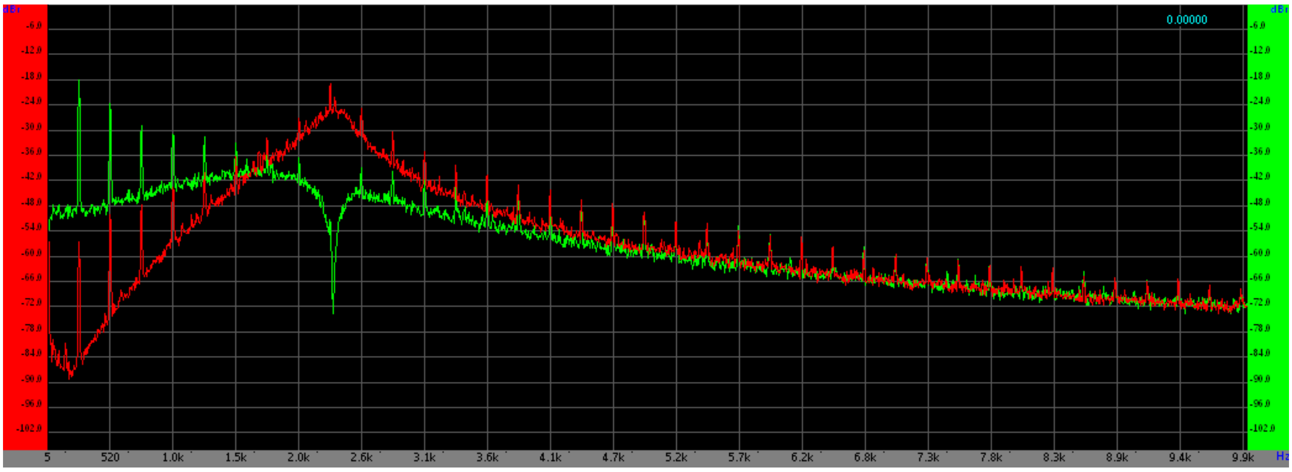
According to the producer the filter uses the same technology as AN1x. Note however, that these filters are not the same. AN1x VCF has different parameters, there is also a second dedicated high pass filter in a serial configuration and a feedback loop.

Below you will find frequency response graphs for various modes. Note that switching filter type on the fly kills sounding voices which is really annoying both from editing and playing perspective.



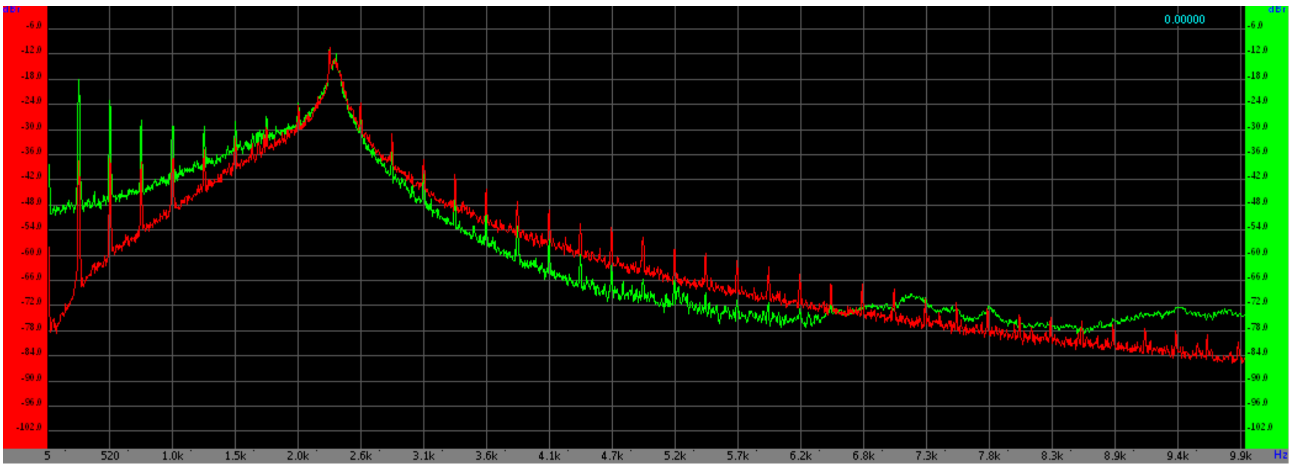
Red - 24dB low pass. Green - 12dB low pass. Resonance=100

There is no dramatic change between the three low pass modes.



Red - high pass. Green - notch. Resonance=50

In the notch mode high resonance actually reduces the band rejection effect. With maximum resonance it can be practically used as an all-pass filter.



Res - band pass. Green - 24 dB low pass. Resonance=80

The filter is wrapped in an FM-style vocabulary like EGTime1-4 and EGLevel1-4 for the envelope. These provide more options than traditional ADSR envelope, but can be confusing.

There is a dedicated LFO just for modulating the filter. Both LFOs can be used together to create more complex modulation of the filter.

Each voice can have its own filter. Using filters requires additional processing power and reduces polyphony. Depending on how many parts have filters and their mono/poly configuration there can be from 1 (all parts mono, any filter configuration) up to 32 (single part active, no filter) notes of polyphony actually. In a single-part-with-filter configuration you still have 16 notes polyphony. This is quite impressive compared to dedicated virtual analog synths.

> You may take a look at more in-depth description concerning polyphony provided by Robert Wolf (cf. 'Polyphony Bug' on http://smasher.org/fs1r/bugs).

Each part can have a different filter or no filter at all. This in general seems good, but can be a problem if you want all voices to respond the same filter-wise (as it is on a traditional analog synth where all oscillators are mixed together and fed to the same filter). Setting all parameters to the same values will not help if Sample&Hold had been used to modulate the filter.

The filter responds to typical MIDI CCs. Unfortunately there is only 7-bit resolution and this is not enough for step-less cutoff point changes (this regards external control via MIDI and does not contradict the overall opinion stated below as the internal filter resolution is higher).

The filter is smooth and fast. The resonance goes really high and the filter can sound aggressive if you want. In specific conditions the filter clips producing sounds slightly to harsh for my taste, but one can easily cope with this by reducing gain on the input stage. Good true analog filters are better, but this simulated one is really decent. You can use it just to darken the sound as FM/FS can produce substantial amount of high frequency content or even as additional gain stage (keeping the resonance at moderate levels).

MIDI Control

We are probably in a golden age of analog mono synths and for a good reason. One of the most important aspects is the hands-on experience when it comes to programming and playing analogs, so let us see what can we get out of FS1R in this regard. Quite obviously it is practically impossible to make a controller that would allow for programming FS1R in a one-knob-per-function fashion, but one can prepare a patch with a lot of external control for playing live.

FS1R has a very comprehensive MIDI implementation. The list below presents briefly what is possible. Refer to the original Yamaha Data List manual for details.

- Parameters directly accessible using dedicated Control Change messages:
  - Portamento Time & Switch
  - Volume/Expression/Pan

  - Note Sustain Switching
  - Envelope Attack/Release Time
  - Filter Cutoff/Resonance
  - Reverb/Variation Send
  - Formant/FM Controls
- Parameters available using RPNs:
  - Pitch Bend Range
  - Note Shift/Detune
- Parameters available using NRPNs:
  - Filter Cutoff/Resonance
  - Envelope Attack/Decay/Release Time
  - LFO1 Speed/Delay and Pitch Modulation Depth
  - LFO2 Speed and Filter Modulation Depth
- Parameters available indirectly using control matrix:
  - Insert FX Parameters and Send Levels
  - Volume/Pan
  - Reverb/Variation Send
  - Filter Cutoff/Resonance/Envelope Generator Depth
  - Envelope Attack/Decay/Release Time
  - Pitch Envelope Levels/Times
  - Formant/FM Controls
  - Pitch/Frequency/Amplitude Bias
  - Voiced/Unvoiced Band Width and Balance
  - LFO1 Speed and Pitch/Frequency/Amplitude Modulation Depth
  - LFO2 Speed and Filter Modulation Depth
  - Formant Sequence Speed/Scratch

Moreover almost all synthesizer parameters can be individually changed using short SysEx messages. In many cases this affects already sounding notes. This can be very useful both for sequencing and playing live (if your keyboard supports sending user programmable SysEx messages).

Note that it is completely up to you to decide what to use depending on your needs and available hardware controller.

Mono or Poly

There are many options to put together oscillators and filters mentioned above. If you want all oscillators to use the same filter everything must be contained within a single voice. For creating more complex oscillators you may want to use modulation which will eat more than one operator. Even then creating 2-4 oscillators within a single voice should not be a problem. Of course there is also the unvoiced part for generating noise. This scenario gives the most consistent oscillator and filter response, but is slightly less versatile when it comes to MIDI control.

A different approach requires more parts, each dedicated to a single oscillator. This has additional advantages like separate portamento and pan settings for example. You can even mix these 'voice' and 'parts' approaches. In any case you can easily fatten the sound by using multiple detuned or transposed oscillators.

Independently of the number of oscillators or parts you can decide how many voices of polyphony you want. Mono and poly modes behave quite differently when it comes to certain details like portamento or envelopes and each has its advantages. To complicate even further you can set one part to mono and another to poly. The former can be used for example for a 'noise' part, so it doesn't accumulate when multiple notes are held on.

With the filter turned on for a single part there is still 16-voice polyphony. This is a lot (as for a virtual analog) and makes poly mode really useful.

Envelope Issues

It appears there are two important problems regarding envelope implementation on FS1R.

There is unfortunately no sustain level available for control. You can obviously set it during editing (this is Level 3), but it is not available as a control destination. Note that the sustain level should affect simultaneously all active unvoiced and voiced operators working as carriers, so doing this is really problematic even using SysEx. Moreover changing envelope parameters do not affect currently playing voices.

> The best thing I came up with is setting Level 3 to zero and using Decay to control the time of sustain (the Decay is available as a NRPN and as a control matrix destination). With bigger Decay values the Level 1 or Level 2 can be sustained for a long time and can imitate envelope with high sustain level. Smaller Decay values can produce percussive envelopes. Alternatively one can prepare two sets of oscillators with different sustain levels and switch or even cross-fade between them. Each approach has its deficiencies but they are better than nothing.

The other problem is that in mono mode the envelope is not re-triggered when playing legato. This gives a smooth transition to the next note which is good for vocal patches for example. In cases of percussive sounds this is totally undesired (to make it clear the envelope re-triggering should be optional). It is practically impossible to avoid the problem if you are playing even moderately fast on the keyboard – the next Note On message will be send earlier than previous key Note Off (in contrast step sequencers send Note Off usually before Note On of the next step so they should not be affected by this phenomenon). This can possibly be corrected by some kind of software or hardware MIDI processor. Eventually you can use a software sequencer to correct the triggering at a later time.

Duophony

While the polyphonic mode seems to be a very good choice there is another interesting option. You can use two monophonic parts with bottom and top note priority respectively. This configuration behaves similarly to ARP Odyssey (although it is not paraphonic). You can detune the parts slightly for a bigger sound for single notes (four parts can be configured that way if two is not enough). With slight portamento it works very nice and natural.

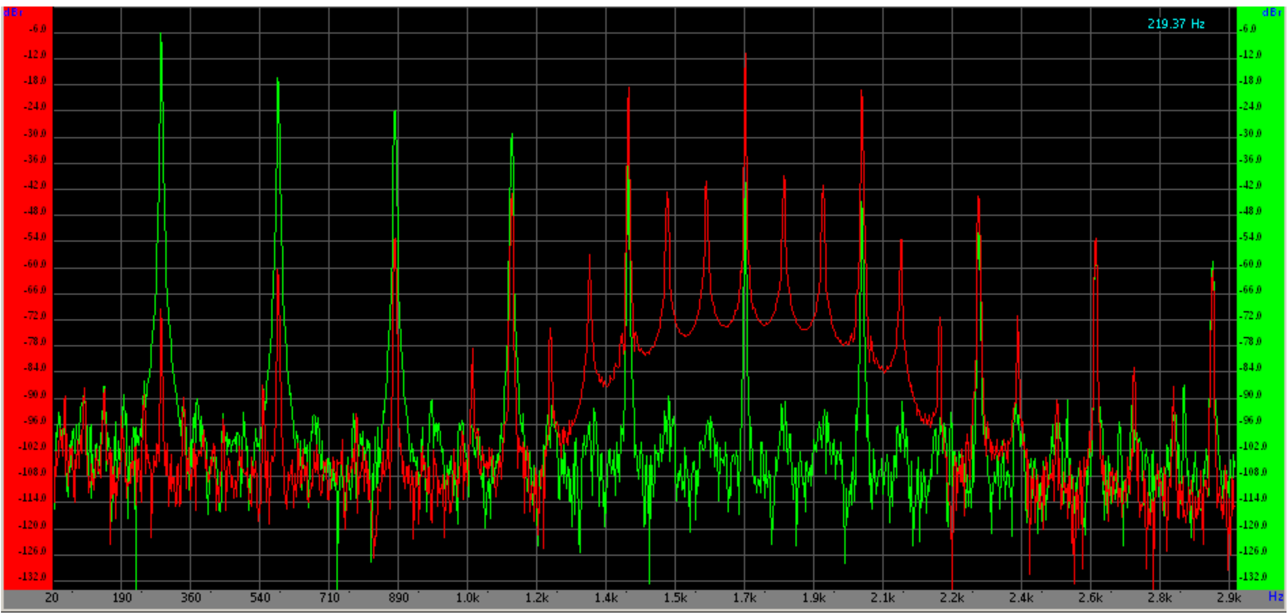
Formant Synthesis

Formants

From an acoustic point of view **formants** are peaks in spectrum of a sound (mathematically they are local or global maximums). In phonetics formants can also mean acoustic resonances produces by human vocal tract. Different phones have different formants. This let us distinguish between them. While occasionally there can be more than 6 formants the first two are most important and usually enough to disambiguate the sounds. But let's translate this wiki knowledge to our hardware...

FS1R provides up to 8 voiced and 8 unvoiced formant operators per voice. These model respectively pitched and noise components of sounds. This is certainly enough to create speech or singing. (In many places in this document the term formant is used as an abbreviation for voiced formant, which hopefully is not misleading, however unvoiced operators are formants as well.)

The spectrum of formant resembles bell. The maximum amplitude is in the middle. This is the **center** or **formant** frequency. The bell extends to the left and right covering certain frequency range. Most of the energy is located in a narrow band near the center frequency. The formant may extend however beyond this band to a various degree. On FS1R these aspects of formants are controlled by operator frequency, bandwidth and skirt respectively.



Red - formant. Green - all1.

To simplify things we can say that formants amplify harmonics passing through the bell. As the shape of the bell is independent of the pitch we have a separation between pitch and timbre. In other words the fundamental frequency and formant frequency are something different and can be independently controlled. This is probably the key to understand how formants work in FS1R.

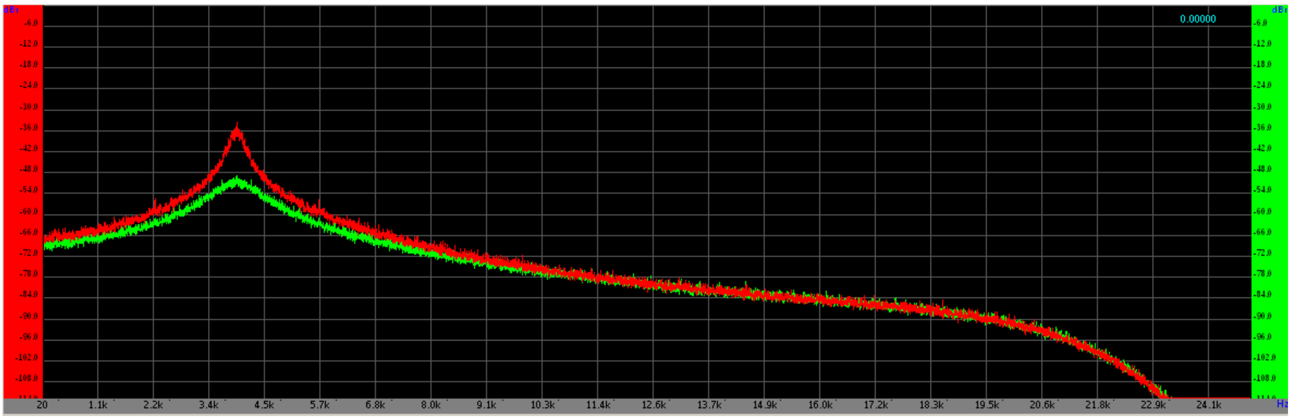
This separation allows to program human voices that cover relatively wide range of notes without the Chipmunks effect characteristic for samplers. But this is not as easy as advertised. For a basic vocal patch with fixed formants this range seems to be around 1-4 octaves, depending on how picky you are. The main problem here is that actual formants of the same phones differ depending on the voice type (like soprano or tenor). Additionally there can be movable formants that change together with pitch. You can use level or frequency scaling to adjust formants accordingly to create more realistic patches that cover wider range. You may also need to setup frequency modulation sensitivity respectively to make the formant vibrate along with fundamental.

The harmonics mentioned earlier are normally related to the fundamental frequency of the note. (We can transpose this frequency at will. The transposition changes the *fundamental* frequency not the *formant* frequency.) Due to the shape of formant and fact that usually the center frequency lies high above fundamental using only formant operators is not sufficient. The sound may lack body. To overcome this “traditional” operators may be used to provide necessary base for the sound.

If you set the formant frequency below the fundamental the resulting spectrum is quite different. There is just slight emphasis near the formant frequency but the harmonics are clearly audible. You can even move the formant frequency down to zero leaving just the harmonics. This can be used to create good quality saw through modulation and feedback (see Earth Lead preset voice for details).

To use formant operators we need to enter the center frequency, bandwidth, skirt and level. This is quite problematic as we don't hear formants as separate frequency components. So how do we get them? The easiest way is to look at some formant tables available on line (see for example Csound manual). If you want to analyze existing sounds using Praat seems to be the best way. Praat offers listings and plots of formants, so you can check individual values and observe how formants change over time. This can be used to setup basic patches. Visual Analyser on the other hand lets you make a snapshot of the spectrum. This can be used to tune the formants on FS1R until they match with the snapshot. You can also try to match the spectrum between left and right channels (feeding looped audio sample and FS1R output to different channels).

The additional unvoiced operators can provide noise component if needed. You can adjust the spectrum to your needs by setting band width, resonance and skirt. For formant synthesis it is convenient to link the frequency with voiced operator of the same number by selecting linkFF mode. Then the unvoiced operator will follow the (formant) frequency of the voiced operator (even if the voiced operator is not sounding because for example its level is zero).


Red – bandWidth=10, skirt=5. Green – bandWidth=50, skirt=0.

Note that noise is present not only in consonants – if you're unsure try to whisper for a moment. LinkFF mode ensures that the peak voiced and unvoiced frequencies appear in the same areas. FS1R provides dedicated control to balance between voiced and unvoiced operators so you can add as much noise as needed. This is the V/N balance which can be changed on the fly.

The bell shaped formants resemble band pass filters. And they should, this is how human vocal tract works – by filtering sound waves generated in larynx. The FS1R approach is however different. Filters work subtractively and FS1R formants additively. In the end the result may be similar so if that helps you to understand or use formants treat them as filters.

Formant Sequences

Formant sequences are control data containing fundamental pitch as well as frequencies and levels for all available operators (8+8). Each sequence can consist of up to 512 frames which can be played in a number of ways. There are 90 preset sequences. There is also place for 6 user sequences if you set the memory mode to Int64. The sequence can be applied only to a single part of the performance.

The preset sequences contain various phrases. They sound very specific what may discourage you from using them. Note however that you don't need to replay them in a 1-1 form. The parameters responsible for setting up the playback mode are grouped in the Common/Fseq menu of the performance. You can select a small part of the sequence and play it at a different pace. Reverse playback is also possible. The internal playback can also be bypassed and the sequence can be driven externally using MIDI CC. This is the *scratch* mode which additionally allows to freeze any selected frame. All this functionality offers many possibilities.

> You can extract a diphthong (adjacent vowels) and use the sequence as fixed or even automated formant filter producing wah-wah type effects. An important advantage of formant sequences is the ability to sync them to MIDI Clock, so the wah-wah can keep the tempo of the song.
>
> At the end of bank B there are 14 voices prepared specifically for formant sequences

> (FseqBaseXY). These can be a good start to explore the preset sequences. Most of these voices can be additionally tweaked using Formant/FM control.

One important thing to note is that the sequences are not limited to formant operators. Depending on the settings they can affect level and/or frequency of any type of operator. You can also switch the sequence for individual operators, so they will affect just part of the voice. Selecting other spectral forms for operators driven by formant sequence can change the timbre quite dramatically.

As suggested by Hannes Jaeckl the sequences can be used as unique LFOs. Indeed depending on the setting both amplitude and/or frequency modulation is possible. From a different perspective this can be seen as adding an envelope (as if there were not enough) with looping capabilities.

There is no doubt that many if not all of the preset sequences are snapshots of actual recordings. This means that they are in fact specifically encoded samples! If you are unsure take a look at how mp3 works. These compression concepts are based on band pass vocoders. FS1R can be seen as an 8 band vocoder. An advantage over traditional vocoder is that the center frequency of each band can be different for each frame. Nevertheless eight bands is not so much so the resulting sounds are intelligible, but rather artificial. (For comparison mpeg audio mp1/mp2/mp3 use 32 critical bands, however in mp3 these are in turn divided into 6 or 18 sub bands).

Note that the sequence frames do not contain band width. This even more supports the vocoder perspective as the bands in traditional vocoders are fixed. On the other hand with bandwidth and skirt formant operators can be much better tuned to match required spectrum. To conclude Fseqs give another approach to programming better suited to rapidly and significantly changing sounds, while formants alone can more precisely define static sounds.

> If Fseqs are samples is FS1R a rompler? Fortunately no. Traditional samplers/romplers basically use PCM so they vary the speed when the pitch changes. This in turn moves formants up or down, which results in the Chipmunks effect. Unlike PCM which takes a snapshot of the waveform formant sequences are essentially snapshots of the whole spectrum. This makes it possible to separate the pitch from harmonic content of the sound. In effect FS1R can change the pitch, slow down or speed up the replay without changing the timbre of the sound. It means the FS1R provides time-stretching on the fly. Advanced modern samplers like Roland V-Synth or Elektron Octatrack can do this too, even with better intelligibility, but FS1R is still unique in the way the sequences can be used. Finally, formant sequences are just one of many possibilities and you don't have to use them to get sounds out of FS1R.

The user sequences open additional possibilities. With FseqEdit you can use your own samples, which is probably the most wanted feature. It seems the user sequences could also be used for sequencing purposes (based just on the pitch values stored in frames), simulating wavetable synthesis or even for turning FS1R into a simple speech synthesizer. The main problem is that Fseqs are rather heavy weight objects. You need dedicated external software for preparing and/or animating them which can be quite burdensome.

Granular Synthesis

In granular synthesis the sound is composed of many separate short waves called grains. We are talking about hundreds or thousands of individual waves which are typically less than 100 ms long. For this reason grains are not perceived individually but rather as a complex sound that has its own unique timbre. The resulting sound depends on the amount of grains, their waveform, frequency and position within the sound field.

> If you take a walk in a meadow during summer you may hear thousands of insects buzzing around, what creates very special, omnipresent sound with distinctive spatial quality. This is granular synthesis in nature. This is just to keep distance to the concept of synthesizer as the source of sounds unheard before.

With 512 oscillators FS1R seems to be a good candidate for granular synthesis. However it is not the amount of oscillators that matters most, but whether they can be controlled in a prescribed way.

Frequency

In FM style synth there are plenty of methods for setting and controlling frequency. On top of that FS adds some more. The relevant parameters are listed below.

- ❑ Oscillator frequency – In ratio mode the frequency is relative to the pitch, which depends on the note played. In fixed mode the frequency is absolute but can be shifted according to the note played using scaling.
- ❑ Operator frequency envelope – simple with two stages
- ❑ Voice pitch envelope – this is the main four stage frequency envelope (with time scaling if you want). This envelope can be adjusted using CCs.
- ❑ FM modulation – this obviously can change oscillator's frequency
- ❑ LFO1 modulation – affects frequency according to LFO Pitch/FreqModDepth and operator Pitch/FreqMod sensitivity parameters
- ❑ Pitch Bend – changes fundamental frequency and affects ratio, formant and linkFO operators. Maximal depth can be set using PB Range (Lo) part parameters (these are independent for bending up and down).
- ❑ Pitch Bias – this is one of the control matrix destinations. Works similarly to Pitch Bend, but with a single up/down depth.
- ❑ Frequency Bias – this control is independent of the Pitch Bias/Bend and works for operators not affected by these controls. The depth of control is set with Voice/Sensitivity/FreqBias.
- ❑ Voiced/Unvoiced Band Width control
- ❑ FM/Formant controllers – each of them can change frequency of up to 5 operators
- ❑ Note Shift and Detune – this sets the offset related to received Note On message. The transposition is available at many different levels. There is
  - ♪ Master instrument note shift (affects all patches)
  - ♪ Performance note shift (affects all parts)

    - Performance part note shift (for every part)
    - Voice note shift (affects all operators)
    - Operator transposition (for certain operator types).
- Portamento – this will be problematic if you trigger notes very fast

Panning

Parameters responsible for positioning the voice within the stereo field are bound to the performance:

- Performance Pan – this is the master setting for the whole performance and affects all parts. It works basically as an offset to pan settings for particular parts.
- Part Pan – selects a fixed position or a randomly generated one for each Note On message
- Part Pan Scaling – applies correction based on the note played
- Part Pan Mod – adds another modulation destination for LFO1 (this is not very smooth especially with higher depth settings)

The initial position gets calculated according to Performance/Part/Pan and Pan Scaling. Then it can be changed at the rate of LFO1 or by using standard MIDI CC #10.

The independent master performance pan helps keep relative positions of parts within the mix. This is also the parameter changed by CC #10 on the performance receive MIDI channel, so you can change the position of all parts at the same time. Alternatively you can change pan positions of individual parts if they are set to different MIDI channels. Note that the master pan setting is always active and affects the range of available pan positions.

The initial positioning is not performed if mono mode is selected and there is no Note Off message sent (in other words the voice keeps pan settings when playing legato).

There is only a single general purpose LFO. Use it with care when controlling both panning and frequency to avoid funny correlation effects.

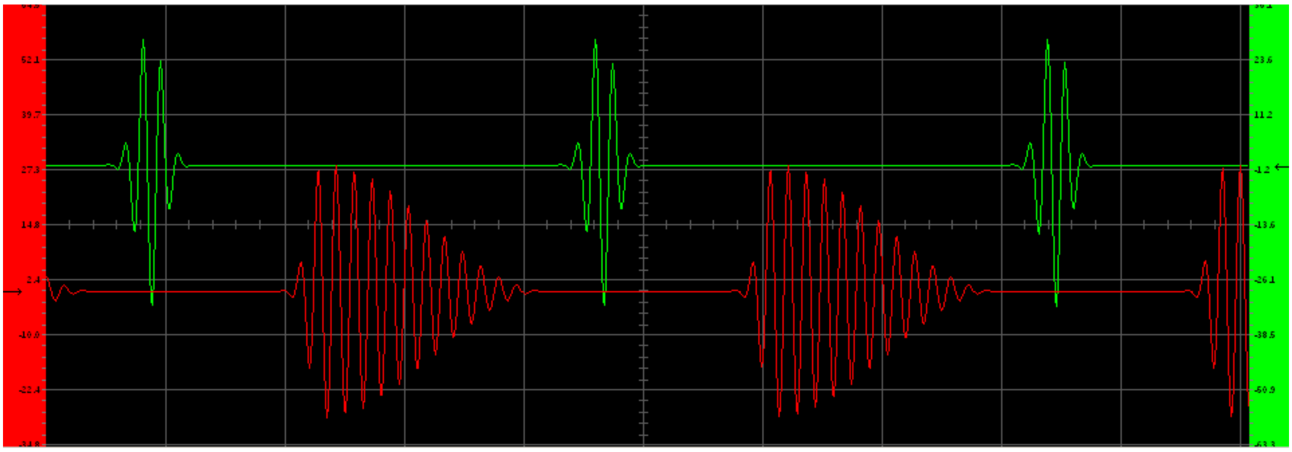
Both frequency and panning can be further changed using FX. There are effects that seem particularly related here like Pitch Change or Panning, but in fact almost all effects change these aspects of sound.

Triggering Grains

In the first approach it is assumed that the synth is controlled using an external sequencer or special program.

Sending a single MIDI Note On message takes below 1ms. This means it should be possible to send more than 1000 messages per second. Whether the receiver will be able to process them on time is a different story. According to my tests it takes around 2500 ms to process 1000 Note On messages. (This is in polyphonic mode with a single part patch without sending Note Off messages, which can be problematic as the dynamic voice allocation seems a little unpredictable.) It means FS1R can handle around 400 Note On messages per second. At this rate however certain side-effects appear related to how the internal processing is organized. For example the voice LFO may have no effect at all, as it triggers so late that another voice kicks in. The safe rate seems to be around 100 grains per second (but only 32 at a particular moment, as there are exactly 32 voices). This is still enough to demonstrate granular effects.

Another approach requires res1/2 oscillators. These waveforms with higher skirt values behave like impulses. We can treat these impulses as individual grains and control rate at which they are produced by setting oscillator frequency. To extend the wavelength we can set the frequency to LFO range and use high resonance settings to bring the oscillator into audio spectrum. If we set the frequency of individual operators to different values the grains will appear in different moments, which is needed to give relative independence to them. With 32 voices this gives 256 grains available simultaneously. Since we can trigger notes in a standard way (at very moderate rates) there is plenty of space for additional control.



Red - res1. Green – res2. Resonance=24, skirt=6.

Manual Addendum

I've read the manual more times I would like to mention. Both English and Polish (as a rule the translated manual is never better and includes additional errors and misconceptions). Yamaha delivered kind of reference manual, aimed rather at experienced FM users, that lacks both general description and detailed information of many functions.

This addendum tries to supplement the original manual by providing additional technicalities and explanations. For convenience the layout is more or less the same.

General Editing Procedure

There is no separate edit mode. The performance gets loaded into the edit buffer together with voices assigned to particular parts and instantly becomes ready for playing and editing. To keep the changes you need to store performance and/or voices depending on what has been modified. Saving voice affects all performances that use it. Both performances and voices can be saved to a different memory slot than they had been loaded from to keep the original location intact.

If something has been edited the display will contain the edit marker (E). This applies to performance and voice parameters. The marker can be misleading if you've saved just the performance or just the voices. Note also that certain MIDI messages, including many CCs, affect the edit buffer. These changes may easily be overlooked and saved inadvertently.

Performance

Part Assign

• Receiving Channels

Each part can have a different MIDI channel on which it responds. This can be used to create splits if your master keyboard supports it or for sequencing.

For convenience you can set the channel to performance. Then the part will receive messages on currently selected performance channel which is a global setting. Thus if you change the performance channel all programs will adopt to that new settings.

Parts 1&2 have also maximum receive channel setting. This allows to respond to messages on several channels. If for example receiving channel is set to 1 and maximum r.c. to 6 the part will respond to messages on channels 1-6. This setting is used for the G50 guitar converter which can send data for each string on a separate channel. This could also be useful in a complex setup with more MIDI controllers which may send data on different channels. Note that if the maximum receiving channel is lower than the (main) r.c. the part will respond on all channels.

• Bank/Program Number

This selects the voice assigned to part.

Note that in addition to setting the Bank to something other than off you need to set the receiving MIDI channel accordingly to activate the part.

• Reverb/Variation Send, Insertion Effect Switch

These settings basically define routing through effects.

Reverb/Variation Send correspond to standard MIDI CC #91 and #93 respectively. If the part is routed through the Insertion effect these send parameters do not apply. Instead Insertion-to-Reverb/Variation Send Levels should be used.

Common

Control Matrix

The control matrix allows to change various synthesis parameters in real time using MIDI. This is vary useful for playing live as well as sequencing.

Without setting up the matrix you will have almost no control over MIDI. In other words the synth will not respond to aftertouch or modulation wheel! However some MIDI messages, like Pitch Bend or Volume CC, are processed without the matrix.

• Control Destination

Up to 8 destination parameters can be selected. For each destination you can define depth of control and parts it will affect (for relevant parameters). The depth can be negative to reverse the effect of control or to create cross-fading effects. These settings together define so called voice control sets denoted by VC1-8.

The depth equal 31 basically provides 1-1 mapping between source and destination values. This means the minimal/maximal destination values will be reached only in extreme source controller position. If depth is greater than 31 the maximal destination values will be reached sooner.

If depth is less than 31 the source controller will cover only part of possible destination values. This allows for more precise control of synthesis parameters. This is one of the reasons why parameters like Pan, Attack Time, FM etc. for which there is direct control with MIDI CC appear also as matrix control destinations.

Pitch Bias is probably the only destination that behaves slightly differently. The depth determines here the available pitch bend range, which is limited to 24 semitones. Thus the 1-1 mapping is achieved earlier for depth equal 24.

• Control Source

There are 14 MIDI source controls including Continuous Control, After Touch and Pitch Bend. For each source control you can select destinations it will trigger.

> The actual MIDI CC's assignments for sources like MC1-MC4 can be defined globally in the Utility/Control menu, to match the hardware controller you are using.)
>
> The front panel knobs send KN1-KN4 controller values if bottom led near them is lit. It might be worth checking them as they are configured in many preset performances.

There are no restrictions on the assignments between source controllers and destination parameters. It is up to you whether to use some kind of convention here (so for example KN1 always controls LFO1 Speed), which can be convenient from performance point of view.

A single source controller can trigger multiple destinations. This is useful for changing more parameters at once. Conversely the same destination can be triggered by multiple sources. Know what you are doing here however as this can be really confusing.

> Let's say parts 1&2 use the same voice which has FM setup to control frequency of certain operators. Direct MIDI control assigned to FM (CC #81) will change both parts simultaneously. But what if we need independent control for the two parts? To achieve this we can setup two sets VC1 and VC2 with FM as destination for parts 1 and 2 respectively and assign Knob 1 and Knob 2 to these voices.

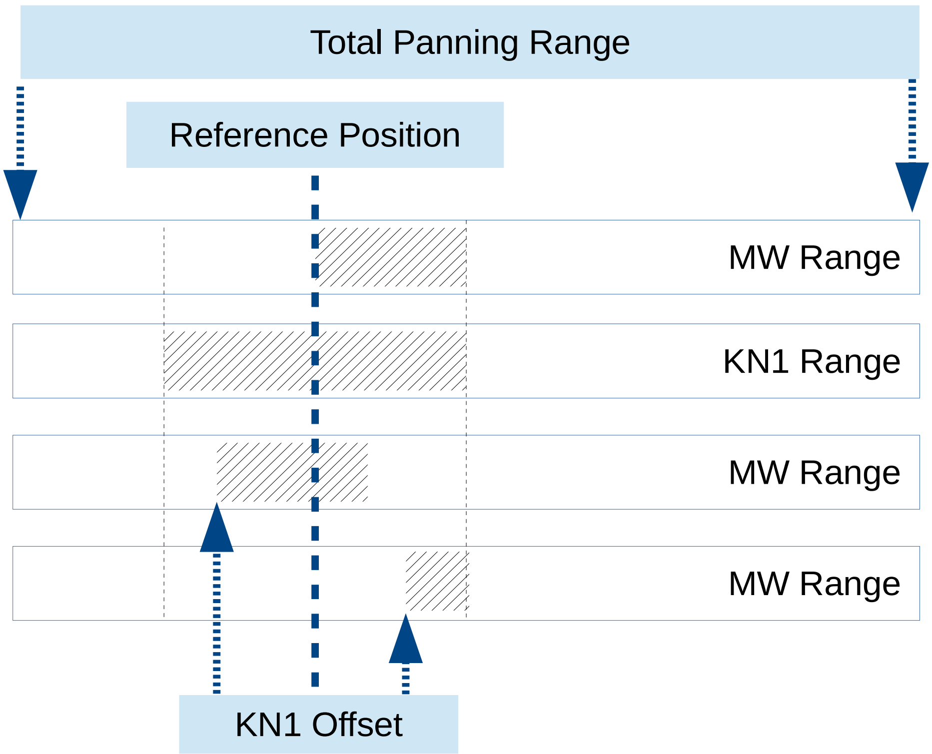
It seems the control matrix follows the rules below (this is complicated):

- ❑ For each destination there is a reference point. Matrix updates the actual synthesis parameter relatively to this point. The reference point is setup during editing but can be changed on the fly using dedicated CCs.
- ❑ There are two groups of source controllers:
  - ♪ Unipolar: FC, BC, MW, CAT, PAT
  - ♪ Bipolar: KN1-KN4, MC1-MC4, PB

  Unipolar controllers can change the actual parameter only on one side of the reference point. Bipolar controllers affect both sides.
- ❑ If more then one source is used for the same destination the actual value is the mean calculated from all sources. This prevents distortion caused by rapid changes, but is very confusing as the result depends on all source controllers all the time.
- ❑ The available destination range depends on the reference point and depth. The actual value calculated by the matrix is restricted to this range irregardless of the source controllers used. This can occasionally be confusing as well, since you move your physical controller and nothing happens.

Let me explain this with an example.
We assume that Panpot has been selected as the destination with a positive depth. The pan position you set in performance/part menu is the initial reference value. You can change it using CC #10 (Pan) – then this will become the new reference value. The response to modulation wheel (MW)

and KN1 controller in this scenario is pictured below (this is hopefully simpler):



The response to MIDI CC is impressive. The synth can handle CCs delivered at the interface rate both for messages routed directly and through the matrix. This means more than 1000 control messages per second. There is no noticeable latency (this is just my impression, no measures taken). All of this makes the FS1R a good companion for any kind of controller including wind controllers which generate lots of messages and require as little latency as possible. Multidimensional controllers, like X-Y pads, can take advantage of many available sources and destinations.

Formant Sequence

• Part

The sequence can be applied to a single part of the performance only. To be more precise only a single sequence is running, independent of the number of notes pressed, which drives all sounding voices of selected part. This has its limitations but ensures that all voices will always be in sync.

• Mode

In the default fseq mode the sequence runs automatically according to the speed setting. If the speed is set to midi then the sequence will be synchronized to MIDI clock. To be more detailed there are 5 midi settings which allow to run the sequence 2 or 4 times faster or slower.

In the scratch mode the sequence is not animated internally, but stays on a selected frame. This selection can be changed on the fly via MIDI, producing scratch-like effects. What is really interesting we can freeze the sound for as long as we want (note events do not alter the selected position, this is unlike in fseq mode).

To engage MIDI control we need to setup Fseq Scratch as one of the control matrix destinations. In scratch mode the Start Offset becomes the reference point for control. This will be also the initial frame so it should be selected carefully.

> Select voice FseqBase01 and sequence Wao (#23). Switch on scratch mode and setup the control matrix accordingly. This should create very basic patch with controllable wah-wah effect.

The scratch mode can also be used statically to set the sequence (and hence the spectrum) to a fixed predefined frame (selected with Start Offset).

Individual Outputs

The individual outputs provide auxiliary mix of parts directed through the insertion effect. This means you need to switch on the individual outputs *and* insertion effect for relevant parts. The mix can be pre or post insertion. Individual outputs are independent of the insertion send levels, so the parts can be fed back to main outputs if needed (this is the default setting after performance initialization).

> The individual outputs can be useful in a number of cases, for example if you want to bypass internal FX and mix everything externally. To direct each part to a different mono output do the following:

- Hard pan parts 1&2 to left&right with insertion effect switched off (these will be directed to main outputs)
- Hard pan parts 3&4 to left&right with insertion effect switched on (these will be directed to individual outputs)
- Make sure all relevant FX and respective send levels are off. Note that the default performance setting for Insertion Dry Level is 127. This is actually the return level to EQ block. Set it to 0 as otherwise the parts directed to individual outputs will appear in the main outputs too. The EQ which can not be bypassed should be set flat.

The individual outputs have fixed volume. The front panel volume sets the levels of headphone output *and* the main outputs simultaneously. To keep the same level of main and individual outputs you need to turn the volume knob to maximum. Do not use headphones without attenuator in this case as the output is high and can damage your hearing!

Part

Most of the parameters included here are offsets to voice parameters. This increases usability of individual voices. The same voice can be used and adjusted in many patches without actually modifying its definition.

Tone

• V/N Balance

This balance is pre-filter. If your voice uses exclusively voiced or unvoiced operators it can be used as additional attenuator which can be controlled via MIDI CC (instead of the filter gain which can not be controlled that way).

• Filter Switch

Activates filter for the part.

The detailed filter definition belongs to voice, but the actual usage of filter is set within performance. This allows to use the same voice with or without filter in different performances (or even in different parts of the same performance).

For convenience this performance parameter is available at the voice menu as Common/Filter/Part Switch, but is not saved within the voice. To avoid confusion remember to activate filters for relevant parts for each new performance.

Envelope Generator

• Attack/Decay/Release Time

These parameters modify multiple envelopes simultaneously - amplitude envelopes of all operators *and* filter envelope (obviously if the filter is switched on).

These are offsets of Time1, Time2&3 and Time4 respectively (note the Decay controls two stages). Positive values extend the time, negative shorten it. As the source envelope parameters behave exponentially, so do the corrections, which can be confusing if the times of particular operators differ substantially.

Attack/Release times can be modified using standard MIDI CCs (#73, #72) as well as NRPNs. The Decay time only using NRPNs.

Unfortunately there is no direct control for the sustain level.

Pitch

• Portamento

Portamento introduces continuous sliding effect between subsequent notes. You can switch it independently for each part of performance. With short times it can make solo instruments sound more natural while longer times can be used for special effects.

Portamento works also polyphonically. In general the last note played becomes the reference pitch from which the sliding will occur. In mono mode, if the sweep is in progress, portamento will continue from the current position rather unless you press the same key twice – then it jumps straight to that note.

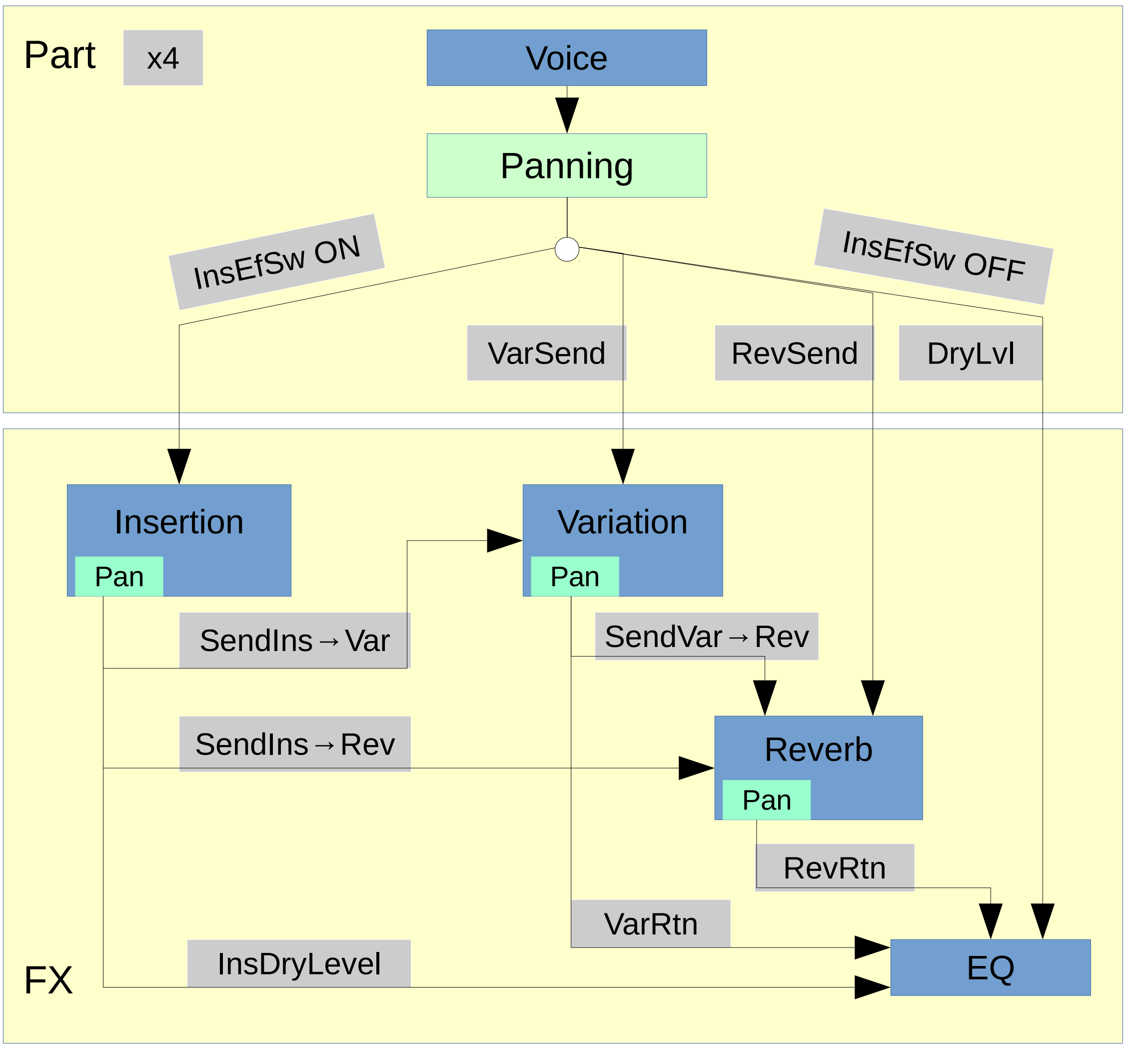
If Fingered mode is selected portamento works only when playing legato.

The actual times seem to be in exponential relation to entered abstract values. Therefore you have more resolution for shorter times which is good. Note also that for the same setting the actual time will depend on the interval between notes (the bigger the distance the longer the time – for example for the same maximal setting of 127 one octave sweep takes around 5sec. while three octaves more than 10).

Quite peculiar the rise time is slightly longer than the fall time (for the same interval). Moreover the sweep in pitch seems to be linear when going up and rather logarithmic when going down.

Effect

FX Block Diagram



Stereo Imaging

Each synthesized voice is a monophonic signal which is panned just before entering the FX section. With independent panning for each part and pan scaling one can create nice stereo image. The FX section can alter this image however substantially. All FS1R effect blocks have stereo inputs and outputs, but internally the processing may be monophonic. Reverb, Variation and Insertion blocks have Pan parameters which effectively balance the output.

Processing variants used in FS1R are listed below:

- Mono – the stereo input is mixed down to mono, the monophonic output is panned to stereo. This is typically used in cases where stereo is not very useful, like various distortions.
- Mono to stereo – the stereo input is mixed down to mono then processed creating stereo output (the stereo image is inherent part of the effect).
  Typical examples include most delays.
- Dual mono – left and right channels are processed independently. Despite the name this is a true stereo mode that usually best preserves the original image.
  This is used in many Variation/Insertion effects including various choruses, tremolo, auto/touch wah, 3-band EQ, compressor and echo.
  Certain effects, for example Chorus, have a mode setting that switches stereo operation on or off. Note that Phaser 2 is dual mono while Phaser 1 is mono or mono-to-stereo depending on the diffusion setting.
- Full stereo – the signal is processed as stereo from input to output.
  This is mostly used for reverbs. (To be more precise reverbs have an early reflection and reverberation stages, which work differently regarding stereo. To increase confusion let me add that the reverb processing is different in Reverb and Variation/Insertion blocks.)

The mix-down means that the input is summed to mono. This is used in all cases with the exception of Cross Delay, where you can additionally select left or right channel as the source for processing.

MIDI Control

The synth responds to standard MIDI Reverb/Variation Send CCs. These provide direct per part control. Additionally there is comprehensive control of the Insertion effect through the control matrix. Not all parameters can be controlled that way however and the delay time is unfortunately one of them. This is not surprising as the delays, while not bad, are implemented in a typical digital fashion and produce unpleasant garbage when changing the time. For the same reason the FX are not MIDI syncable.

Voice

Common

LFO 1

This is the main LFO which can modulate operator frequency, amplitude and filter cutoff point. You can set the depth of modulation independently for these destinations.

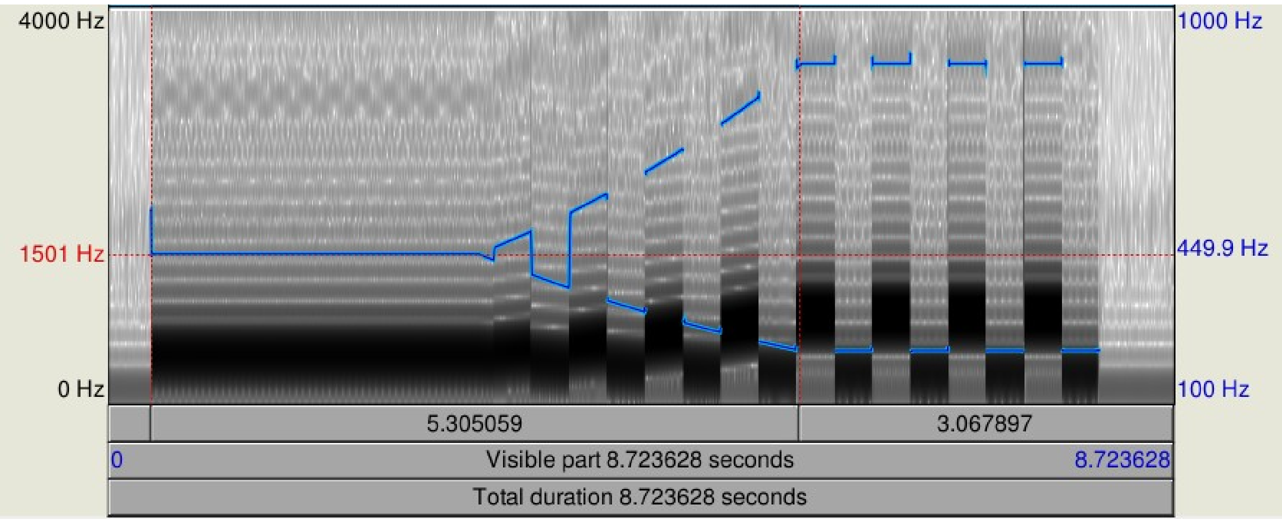
Be aware that the LFO operation can be affected by intensive MIDI input and even using the front panel can produce strange artifacts. This is most noticeable with high speed/depth LFO settings.

• Speed

For the slowest setting (0) the frequency is around 0.065 Hz (the cycle takes slightly more than 15 seconds). The fastest setting (99) is around 50 Hz.

• Delay

This delays triggering of LFO. Note that LFO is not started rapidly after the prescribed time but rather smoothly as can be seen below:



Blue - pitch modulation resulting from square wave LFO

Maximum delay time is slightly above 5 seconds.

• Pitch/Frequency Modulation Depth

Pitch/Frequency modulation works essentially the same but affects different operators (depending on their frequency mode).

Pitch affects:

- ❑ Voiced operators working in ratio mode
- ❑ Formant operators (changing fundamental frequency)

Frequency affects:

- ❑ Voiced operators working in fixed mode
- ❑ Formant operators (changing formant frequency)
- ❑ Unvoiced operators working in normal mode

The modulation will have an effect only if Pitch/Freq Mod in the Voice/Sensitivity section is set for relevant operators. You will find rough estimates of effective modulation depths below:

Depth	Sensitivity	Effective Range
99	7	Around -2 ··· +2 octaves
99	6	Around -15 ··· +15 semitones
99	5	Around -9 ··· +9 semitones
99	4	Around -5 ··· +5 semitones
99	3	Around -3 ··· +3 semitones
99	2	Around -2 ··· +2 semitones
99	1	Around -1 ··· +1 semitones
50	7	Around -1 ··· +1 octaves

Usually you will set the same sensitivity for all operators which you want affect. Then the sensitivity works as a range selector and the depth can be used to fine tune the effective modulation depth. Different sensitivities can be used to achieve detuning or special effects.

LFO 2

This LFO is dedicated to the filter cutoff point. It has slightly different parameters and behaves differently than LFO1. You can use both LFOs to modulate the filter at the same time to create more complex patterns.

• Speed

This is quite different from LFO1.

First the setting of 0 switches off modulation (so it seems, but who knows, I've tested it only for a couple of minutes). However the initial step, according to the waveform, phase and depth settings, is even then performed. This can be useful with sample&hold waveform, which will produce different cutoff points for subsequent notes in a single-shot random manner.

For the slowest non-zero setting (1) the frequency is around 0.046 Hz (the cycle takes around 22 seconds). The fastest setting (127) is around 43,5 Hz.

Due to different concrete frequencies the LFOs will drift inevitably. For shorter periods you can use the following correspondence if you want to match the speeds:

LFO 1 Speed	LFO 2 Speed
10	35
21	70

Note also that LFO1 Delay has no effect on LFO2, which starts immediately.

• Filter Modulation Depth

With maximal setting (99) practically the whole audio spectrum is covered.

Filter

The voice contains full filter definition except activation switch which is part of the performance.

• Part Switch

This switch is placed here for convenience. This means that if you change its state and save just the voice it will have no effect – you need to save the performance.

• Frequency Scaling

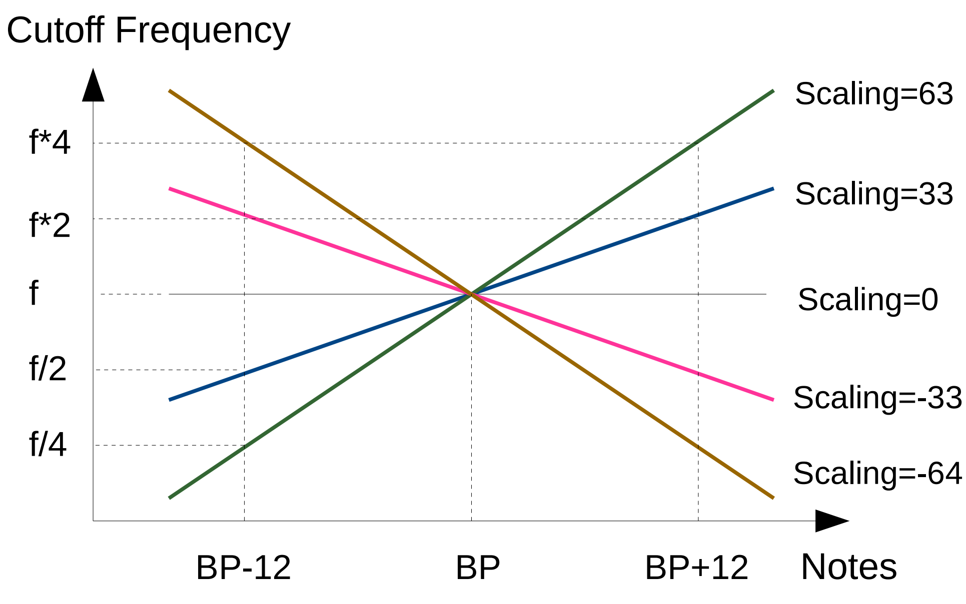
This parameter controls the way the cutoff frequency changes depending on the note you play, commonly known as keyboard tracking.

Frequency Scaling	Keyboard Tracking
0	None
33	Around 100%
63	Around 200%
-33	Around 100% reversed
-64	Around 200% reversed

Note that the actual cutoff frequency according to scaling is calculated generally on receiving Note On messages and unfortunately is not corrected in mono mode when playing legato.

• Frequency Scaling Breakpoint

This defines a reference note for scaling. For this note no change in frequency takes place independently of the scaling value, as illustrated below (frequency is in logarithmic scale).



To get the same response parts must have exactly the same frequency scaling and breakpoint. Otherwise the cutoff frequencies will drift depending on the key.

• Envelope Generator Times

The actual times correspond exponentially to the entered values and are independent of levels and depth settings. The table below contains rough estimates of maximal periods corresponding to upper settings (timeX=99).

EG Stage	Maximum Period
1	10 s
2	10 s
3	55 s
4	55 s

• Envelope Generator Levels

These values define offsets (with respect to nominal cutoff frequency) used to calculate boundary frequencies for subsequent envelope stages. These offsets are scaled according to EG Depth setting. The envelope generator sweeps between these boundary frequencies. The sweep is logarithmic (the initial change is faster then it slows down when approaching the end of particular stage).

With high EG Depth and Level settings practically whole audio spectrum is covered.

• Envelope Generator Depth

This is used to scale the whole envelope. Negative values invert the envelope.

Note that even with all levels set to 0 there is slight change in frequency, which can cause distortion with high resonance. If you want flat envelope you need to set all Levels *and* Depth to 0.

EG Depth is one of the destinations available for the control matrix. This can be used not only to increase or decrease the influence of filter envelope, but also to invert it or simply make it flat. This is very useful as there is no separate control for amplitude and filter envelopes.

Once started the filter runs until it is reused by subsequent Note On or until it is shut down because of editing. After initial excitation the filter can be heard even if all oscillators become totally silent. This is most noticeable with high resonance settings, but the level is low so this is not a big issue. To make sure the filter closes after Note Off you may need to setup the envelope accordingly which can be a bit tricky if you manually control EG Depth. An alternative approach is to use Noise Gate from the Insertion block to get rid of this unwanted dirt (if you've ever wondered why there is a noise gate on a digital synth without external input this is the answer).

Pitch Envelope Generator

This is the voice main 4 stage pitch envelope. It affects the fundamental frequency of:

- ❑ Voiced operators working in ratio mode
- ❑ Voiced formant operators
- ❑ Unvoiced operators working in linkFO mode

Note that there are independent frequency envelope generators per operator, which are much simpler (with 2 stages). The final frequency is the result of both envelopes.

Note also that in case of formant operators pitch and frequency envelope generators control fundamental and formant frequency respectively, which are independent parameters. This may be one of those cases where using both envelopes for the same operator makes a lot of sense.

• Times and Levels

All envelope stages behave similarly. The actual frequencies at boundary points are calculated based on Levels (in an exponential way) and scaled according to Range setting. The actual sweep time depends on Time (in an exponential way) and difference of boundary frequencies. This keeps more or less constant rate at which the frequency changes. The sweep in frequency changes slightly – is faster in the beginning and slows down in the end.

In the table below you will find example times for range=8oct. The times are the same for all stages.

Time	Level Difference	Sweep Time
80	50	7.8 s
80	100	15.6 s
98	50	43 s
99	25	1.7 s
99	50	1 m 25 s

The pitch and frequency envelopes are different when it comes to actual times and frequency offsets (for example PEG Time1=50 has the same duration as FrqEG AttackTime=47), so don't rely on the abstract numbers if you need to match the envelopes for some reason.

• Range

The maximum deviation from the nominal pitch is achieved when levelX=±50.

Range Setting	Actual Range
8 octaves	±4 octaves
2 octaves	±1 octave
1 octave	±6 semitones
1/2 octave	±3 semitones

Others

• Algorithm

This selects the way voiced operators are combined together. Unvoiced operators are simply added to the mix without any FM.

Depending on the selected algorithm all operators except #8 can appear as carriers or modulators. This is why in the initial voice operator #8 is chosen as the only sounding operator.

If you change the algorithm operators that were carriers can become modulators which may result in getting the golden sound of all that is silence. The feedback loop also changes. Therefore it is good to select the algorithm carefully as changing it can cost a lot of work if you need to exchange configurations of many operators.

• Feedback

There is only a single feedback path in FS1R. You can try to use skirt instead, which behaves similarly.

• Formant/FM Control

These are intended for live voice control using MIDI CC. You can also use the front panel knobs labeled Formant/FM which basically do the same as CC.

Both Formant/FM work the same and you can decide whether to use some sort of naming convention here. Both can control up to 5 voice parameters. For each of them you select the operator that is being affected, destination parameter and control depth.

The same operator can be used repeatedly, so for example you can control both frequency and level simultaneously.

The initial value of Formant/FM setting is saved within performance and can be different for each part even if they use the same voice.

The Formant/FM is essentially a bipolar control and works like this:

- ❑ For each destination a reference value is calculated
- ❑ This reference point corresponds to the middle position of the source MIDI controller value (64). So for CC value=64 there will be no change independently of the depth settings.
- ❑ If depth is positive then CC values >64 increase the actual destination parameter, while values <64 decrease it (with respect to the reference point)
- ❑ Negative depth inverts this behavior, but the ranges do not match exactly (for example if operator level is set to 49 then the depth=-25 equals +24 when it comes to available correction range – this is something to take into account if you need 100% accuracy)
- ❑ The actual destination parameter is limited to a specified range (for example the resulting output level can not be negative)
- ❑ Higher (absolute) depth settings increase the range of control

The level correction affects the amplitude. As there is no negative amplitude the effect of correction depends substantially on the initial operator level. If set to 0 the correction will not take place in one half of the CC range values (0-63 or 65-127). This can be used to include additional sound components after certain control level is reached (12 o'clock).

> Set levels of operators 1&2 to zero and assign the FM to control their levels with depths +59 and -60 respectively. With these settings in the middle FM position no sound will be heard. Above the middle position we will hear only operator 1 and below only operator 2 (boundary values 0&127 correspond to full operator level =99). This works like a combination of toggle switch with volume control and can be really useful.

Two additional observations should be mentioned regarding level correction. Firstly note that you can increase the output beyond the nominal operator level, even if it is set to the maximal value (99). This can be used to give extra boost for certain operators (the excess is around 5 dB). For certain types of operators this introduces interesting clipping. Secondly it seems the correction curves do not allow for getting uniform operator cross-fading (so the level is constant). This is quite unexpected and unfortunate.

The frequency correction changes

- ❑ Pitch of voiced operators except formant
- ❑ Center frequency of formant operators
- ❑ Pitch of unvoiced operators working in normal mode.

Unvoiced operators in linkFF mode follow the correction applied to the corresponding voiced operator.

The maximal correction range for depth equal +63 or -64 is nearly ±4 octaves. For depth equal +15 or -16 it is close to ±1 octave.

The band width correction applies to voiced and unvoiced formant operators. As in case of level the effective width is limited to available range (as there is no negative band width).

Operator

Oscillator

• Fseq Switch (V & N)

This has an effect only when a formant sequence is applied to the relevant performance part.

For voiced operators:

- ❑ If switched on the oscillator frequency and level will be taken from the formant sequence. These effectively override parameters set with coarse/fine tuning and level. The output can be corrected by setting attenuation.
- ❑ Fseq frames contain also pitch data. This will be used for ratio mode operators (fixed are not affected) even if the fseq switch is set to off. The use of pitch data can be turned off by selecting fseq pitch mode to fixed.
- ❑ The formant sequence can be applied to all types of oscillators. For types other than formant be sure to set the frequency mode to fixed as in ratio mode the actual frequency lands generally below audio range.
- ❑ The effective frequency can be corrected using transposition (for formant operators) and detune (for all types).

For unvoiced operators:

- ❑ If switched on the oscillator frequency and level will be taken from the formant sequence if frequency mode is set to normal or linkFF.
  In linkF0 mode only the level is taken from the sequence.
- ❑ Fseq pitch data will affect only linkF0 mode.
- ❑ The effective frequency can be corrected using transposition.

• Fseq Track (V)

Formant sequences are organized into 8 tracks. Each track contains frequency and level data for a pair of operators (voiced and unvoiced). Thus selecting the track for the voiced operator changes it also for the unvoiced operator of the same number.

Changing the default setting can be useful in a number of cases:

- ❑ For auditioning particular tracks
- ❑ To adjust the assignment to suit particular algorithm
  (Usually you will use formant tracks for carriers, so if operator #1 works as modulator you might want to assign track #1 to a different operator.)
- ❑ To create intervals or chords by transposing operators using the same track

• Spectral Form (V)

Each voiced operator, irregardless of whether it is a carrier or modulator, can use one of the 8 available waveforms called “spectral forms”:

- ❑ sine – single harmonic (fundamental)
- ❑ all 1/2 – all harmonics with decaying amplitudes
- ❑ odd 1/2 – odd harmonics with decaying amplitudes
- ❑ res 1/2 – selected consecutive harmonics with symmetric amplitudes
- ❑ formant – all harmonics with bell shaped emphasis

Sine is a standard waveform for digital FM synthesizer.

All/Odd/Res forms have two variants. These differ similarly to wide and narrow analog pulse waves. Effectively '1' sound fuller and '2' thinner.

All/Odd forms with higher skirt values produce spectrum similar to saw and square respectively. These waveforms can be used as basic saw/square equivalents.

If skirt is zero, which is the default initial value, all/odd generate narrow spectrum with one or two harmonics (first and third). Therefore usually you will want to increase the skirt.

With similar configuration all/odd waveforms appear quieter than sine, because the all/odd fundamental has lower level. This seems to balance the overall volume of these waveforms which have additional harmonics.

Res 1/2 are very intriguing. They seem to be in fact impulses. The width of the impulse is controlled by the skirt. This provides several interesting applications:

- ❑ Gating signals

Setting frequency to LFO range and skirt above 0 produces gating signals. As the impulse is not rectangular this is not a typical (analog) pulse wave, but the behavior is quite similar. This can be used for producing sequencing effects for example.

- ❑ Frequency shifters
  You can shift the spectrum of assigned formant sequence track according to the setting of resonance
- ❑ Sync-type sounds
  Sweeping the resonance in real-time can produce interesting timbres resembling slightly sync sounds, although less aggressive. Unfortunately the resonance parameter can be updated only using SysEx.

Formant waveform is the key component of formant synthesis. This waveform produces spectrum which emphasizes harmonics within certain range. The center frequency of that range and its width are independent of the fundamental frequency (and can be independently controlled). This waveform models human vocal tract.

• Frequency Mode (V & N)

For voiced operators there are two modes:

- ❑ ratio – the operator frequency changes relatively to the pitch (to say it more formally the value equals fundamental frequency multiplied by nominal operator frequency expressed as an abstract ratio)
  It means the frequency will follow the notes played in a usual expected way.
- ❑ fixed – the operator frequency stays fixed at the entered value expressed in Hz
  This is useful if you want to use the operator as LFO (let's say below 20Hz) or create drone sounds (than the frequency should be in audio range).

For unvoiced operators there are three modes:

- ❑ normal – it behaves essentially like voiced fixed mode
- ❑ linkFO – the frequency follows fundamental pitch (of the whole voice) according to transposition which can be set within 2 octaves up or down
  The name probably comes from f0 which is used to denote fundamental frequency.
  This mode is mostly useful if you want to use the unvoiced operators as additional pitched operators.
- ❑ linkFF – the frequency will follow exactly the formant frequency of the voiced operator of the same number (therefore this mode is available only if formant is selected as the spectral form of the corresponding voiced operator)
  This mode is designed specifically for formant synthesis to bind voiced and unvoiced operators.

Note that the actual frequency depends on more parameters, like envelopes or LFO.

In fixed/normal modes the frequency can be eventually changed according to the notes played by using scaling. Fixed/normal oscillators do not follow Pitch Bend messages however.

• Frequency Coarse/Fine (V & N)

These parameters set nominal frequency of the oscillator.

Total range of available frequencies depends on the oscillator frequency mode. This total range is divided into chunks. The coarse parameter selects the chunk. Note that chunks are not separated but overlap, although the specific frequency values differ. Therefore you can find a closer frequency to the one you are looking for in a different chunk.

> To give you an example let us assume we want the 32nd harmonic, so we are looking for ratio 32.00. The last coarse frequency in ratio mode is 31.0. However within this chunk there is no exact 32.00 value (the closest are 31.93 and 32.24). You will find exactly 32.00 in the chunk selected with coarse frequency equal to 25.00.

For formant oscillators these parameters set the formant frequency (the center frequency of the formant), which is independent of the fundamental frequency.

• Frequency Scaling (V & N)

Frequency scaling is available for fixed/normal oscillator modes. If set to a non-zero value the frequency, even though it is nominally fixed in these modes, will actually change depending on the note played. If set to 99 it will follow the notes exactly (100%) which means that if you play an octave higher the frequency will rise an octave. If set to 49 the frequency will follow the notes in a 50% proportion, so if you play an octave above (+12 semitones) the frequency will rise only 6 semitones.

For formant operators scaling changes the formant frequency. Movable formants depend on the fundamental frequency, so if you set scaling for them, they will follow the pitch. This can be used to achieve more natural sounds or extend usable range of notes. High formant frequency and scaling can easily produce aliasing so it's good to check the sound in upper keyboard registers.

Frequency scaling can be useful in a number of cases:

- ❑ When using unvoiced operators as additional voiced operators
- ❑ When using operators as LFOs to vary their frequency slightly for different notes
- ❑ When creating deliberately dissonant sounds that do not follow traditional pitch
- ❑ To adjust formant frequency

If the frequency is in the audio range you may want to tune it to standard pitch. You can find the exact value by setting another operator with appropriate ratio and execute tuning until there is no noticeable beating. Note that fixed oscillators even tuned exactly as in ratio mode do not behave 100% the same, for example they do not follow voice pitch envelope. These differences can be usefully exploited (but also can cause trouble if you are unaware of these subtleties).

Without transposition and for the global tuning A4=440 Hz operator with ratio=1.0 equals fixed operator with frequency=261.7 and scaling=99.

• Transpose (V & N)

For voiced operators this parameters is available only for formant oscillators. Contrary to the manual it changes the fundamental frequency and not the center/formant frequency.

For unvoiced operators it changes the center frequency of the noise generator. The center frequency depends on the operator's frequency mode.

• Skirt (V & N)

For voiced operators skirt controls the number of side bands generated and their level. This works similarly to feedback.

For unvoiced operators this controls the level of low and high frequency content (below and above the center frequency).

• Output Level (V & N)

This is the nominal operator level representing maximum amplitude. It is used as a reference value for calculating actual operator level for each played note.

The following rules apply:

- ❑ The actual level may be different if level scaling is used.
- ❑ The actual level changes according to amplitude envelope settings.
- ❑ Changing output level will have no effect on currently played notes, but will affect notes triggered afterwards.
- ❑ Formant sequence (if switched on for the operator) overrides actual level.
- ❑ In any case the resulting output level of voiced operators can be attenuated.
- ❑ In any case the output level can be corrected using FM/Formant control

One important thing to remember is that the operator may be effective even if its level is set to zero. Possible causes are included in the list above – level scaling, FM/Formant control and FSeq. So switching off an operator requires not only turning down its level but also reseting some other parameters. Generally it is a good practice to leave all “unused” parameters at their neutral settings.

• Level Scaling (V & N)

Levels of both voiced and unvoiced operators can be corrected depending on the note played. This can be useful in a number of cases:

- ❑ To make splits or cross-fading sounds
- ❑ To provide consistent perceived level and timbre across the keyboard
- ❑ To augment certain frequency range, for example so beloved bottom end
- ❑ To tame aliasing which may appear only in higher registers

To correct voiced operator you set the breakpoint note which corresponds to the nominal level and setup correction curves for notes below (left curve/depth) and above (right curve/depth).

Unvoiced operators have just a single correction parameter (scaling) and a fixed breakpoint note (C3).

Note that operators corrected using this scaling can produce output even if their nominal level is set to zero.

• Operator attenuation (V)

The effective level of voiced operators can be attenuated by up to 22.5 dB.

The correction can be changed on the fly and affects playing notes. It's a real pity that the correction is limited both in depth and resolution, it could have been much more useful.

From the front panel this parameter is available only for carriers. However via MIDI it works for all voiced operators (although the setting cannot be effectively saved for modulators).

Attenuation can be used

- ❑ To correct output of operators controlled by formant sequence, which overrides the nominal level; this maybe quite important to avoid clipping
- ❑ To simplify editing as it works in real-time
- ❑ To directly enter formant amplitude data which are usually given in dB

Amplitude EG

Each voiced and unvoiced operator has its own amplitude envelope.

This is a multistage envelope with hold, attack, decay1, decay2 and release stages. The actual time seem to relate exponentially to entered abstract values. This gives better accuracy for shorter periods.

Rough estimations of maximal periods corresponding to 99 are given below:

Stage	FM Name	Max. Time (min.)
Hold	Hold	1:27
Attack	Time1	0:57
Decay1	Time2	0:57
Decay2	Time3	0:57
Release	Time4	0:57

The Hold stage affects also the frequency envelope. This means the attack of both envelopes begins simultaneously. This is expected behavior.

The initial voice envelope has both attack and release time equal 0. This is too fast and causes unpleasant clicking, most noticeable for the sine wave. This is probably a mistake as decay1/2 times are equal 20 – just enough to avoid clicking in the attack/release stages.

Level4 should normally be set at zero. Other values will make the voice sound (almost) for ever. This can be used to create drones that keep sounding even after the release stage (know what you are doing here as it can be quite risky during a concert).

Frequency EG

Each voiced and unvoiced operator has its own frequency envelope.

The relation between actual frequencies and rise times and level/time settings is non-linear. This gives better control for shorter periods of time and closer frequencies.

The time setting really correspond to the rate at which the frequency changes. This means that for the same setting of Attack Time the actual rise time will depend on the levels. You will find rough estimates of actual frequencies and times in the tables below.

Level	Frequency
+8	+3 semitones
+19	+7 semitones
+28	+12 semitones
+38	+19 semitones
+50	Nearly 4 octaves above
-8	-3 semitones
...	...

Time	Level	Actual time
30	-50	0.3 s
50	-50	3.2 s
50	-28	0.8 s
60	50	11 s
60	28	3 s
99	50	11 m 40 s
99	28	2 m 55 s

You can use the envelope to model frequency changes characteristic for acoustic instruments (related for example to changes in the tension of the string being plucked). These will be rather short term. Longer periods can be used for slowly evolving sounds or drones.

For formant operators this envelope changes formant frequency (not fundamental frequency). When the formant frequency moves within the spectrum various harmonics gets accentuated, while the fundamental frequency stays fixed. This resembles analog synced oscillators and sweeping the formant frequency with the envelope can sound a bit similar.

Sensitivity

These settings define response to amplitude and frequency related messages sent via MIDI and influence of LFO1. The default settings after voice initialization are equal zero. The main consequences of this are that the voice will have fixed velocity and will not react to LFO1 changes made in Voice/Common menu.

Each operator has its own sensitivity setting. Available options depend on the type and frequency mode of the operator.

Negative values for velocity and bias revert the response. This can be used for cross-fading effects for example.

• Amplitude/Frequency Velocity

This defines the response to how hard you hit the keyboard.

• Amplitude-EG/Frequency/BandWidth Bias

This defines the response to changing the relevant destination parameters via control matrix (there are no direct CCs for controlling these parameters). It means you need to setup the control matrix accordingly for this to take effect.

The Amplitude EG Bias can be used as an alternative to channel volume. Unlike volume which affects the whole voice AmpEG Bias can vary the response depending on the operator. This lets you leave for example more bottom end for low level sounds. Both AmpEG and Volume are typically used for wind controller patches to vary the level according to breath controller input.

Frequency Bias works for fixed voiced, formant and normal unvoiced operators. Fixed/normal operators do not react to pitch bend messages. Formant operators follow pitch bend, but it affects fundamental frequency, while frequency bias changes formant frequency.

> You can apply pitch bend to the above frequencies indirectly by using the control matrix. To do it you need to set up Pitch Bend as the source controller and Frequency Bias as the destination parameter. (Of course Frequency Bias can be used independently of Pitch Bend and be assigned to any available source controller.) Pick sensitivity and control depth to match the frequency bias response with pitch bend. It can be hard to match it exactly however; refer to the table below for a few examples. Note that the matching must be done whenever you change the pitch bend range and will work

correctly only if bend-down and bend-up ranges are equal.

Frequency Bias Sensitivity	Control Destination Depth	Pitch Bend Range
1	10	±2 semitones
2	31	±1 octave
4	15	±1 octave

For the BandWidth there are actually two destination parameters – for voiced and unvoiced operators. If you want to control them together use two voice control sets with relevant destinations.

• Pitch/Amplitude/Frequency Modulation

These are related to LFO1. You need to set them to something other than zero as otherwise the LFO will have no effect.

Utility

System

Master

• Tune

This is the global tuning of the instrument.

- ❑ Ratio mode operators follow the tuning exactly
- ❑ Fixed mode operators follow the tuning proportionally to frequency scaling.
  This means if scaling is 0 the oscillator will keep the frequency irregardless of the tuning.
  If scaling equals 99 the oscillator will follow the tuning as in ratio mode.

MIDI

• Device Number

This is mostly needed if you have more than one FS1R (in the same MIDI chain). By setting device number to different values you can direct SysEx message to exact unit.

If set to off the synth will almost completely ignore SysEx messages. This means you will not be able to program it from outside.

• Receive Bulk Dump

If set to off the synth will ignore changes to stored performances and voices, but the edit buffer will still accept changes. This means you can change current performance and voices, but you can't save them from outside. This protection can be useful for concerts.

• Receive System Exclusive

This is more restrictive than the bulk dump switch above. If set to off even the edit buffer will be protected.

• Receive Note

If you have two FS1R units (who is so lucky?) you can set them to react to different notes. This can be used to increase available polyphony and should work for most piano or organ sounds (this is where it's needed most). It maybe problematic for specific patches for example if spatial/panning effects are used.

• Receive Bank Select

You can switch it to off to ignore bank select messages if you are afraid of doing this inadvertently from your keyboard for example.

> The restrictions mentioned above concern messages received via MIDI and do not affect the front panel, so can not be considered as secure memory protection.

Dumpout

Save yourself a headache and backup your patches regularly. There are several options to do it. I am using arecordmidi utility on Linux to save the dump-outs from FS1R directly to a standard MIDI file, but you can use any sequencer to do it. Full backup should contain all performances *and* voices (so you need to call the Util/Dumpout function twice).

The dump-out will not work properly if the device number is set to off (although no error is reported). To restore the data the Receive Bulk Dump/System Exclusive switches must be set to on and the device number must match the one stored in the backup.

Remember that you don't need the latest backup but a *working* one, so test in advance whether you can restore something to ensure your procedure is reliable.

Initial

- ❑ Both Performance and Voice initialization function operate on the editing buffers. This means no changes will be permanently made until you save.
- ❑ Voices are shared by all performances and edited within them. If you change the voice it will affect all the performances using it. Keeping consistent naming for voices and performances may help a lot.
- ❑ Performance initialization does not change voices assigned to particular parts. Therefore if

you initialize performance and all 4 voices, but save only the performance, you will keep the old voice intact and after reloading the patch will sound different.

Block Diagrams

The diagrams included at the end of this document illustrate in a simplified form the synth from the outside HUI/MIDI perspective. There is no doubt that the internal architecture is different and much more complex.

MIDI Routing

Channel messages affect parts according to their receiving channel setup. Note that a single part can respond on multiple channels.

Messages routed through the control matrix are further restricted to the parts activated in the destination menu. The control matrix is independent of the direct control however (Pitch Bend for example can work directly and through the matrix at the same time doing different things).

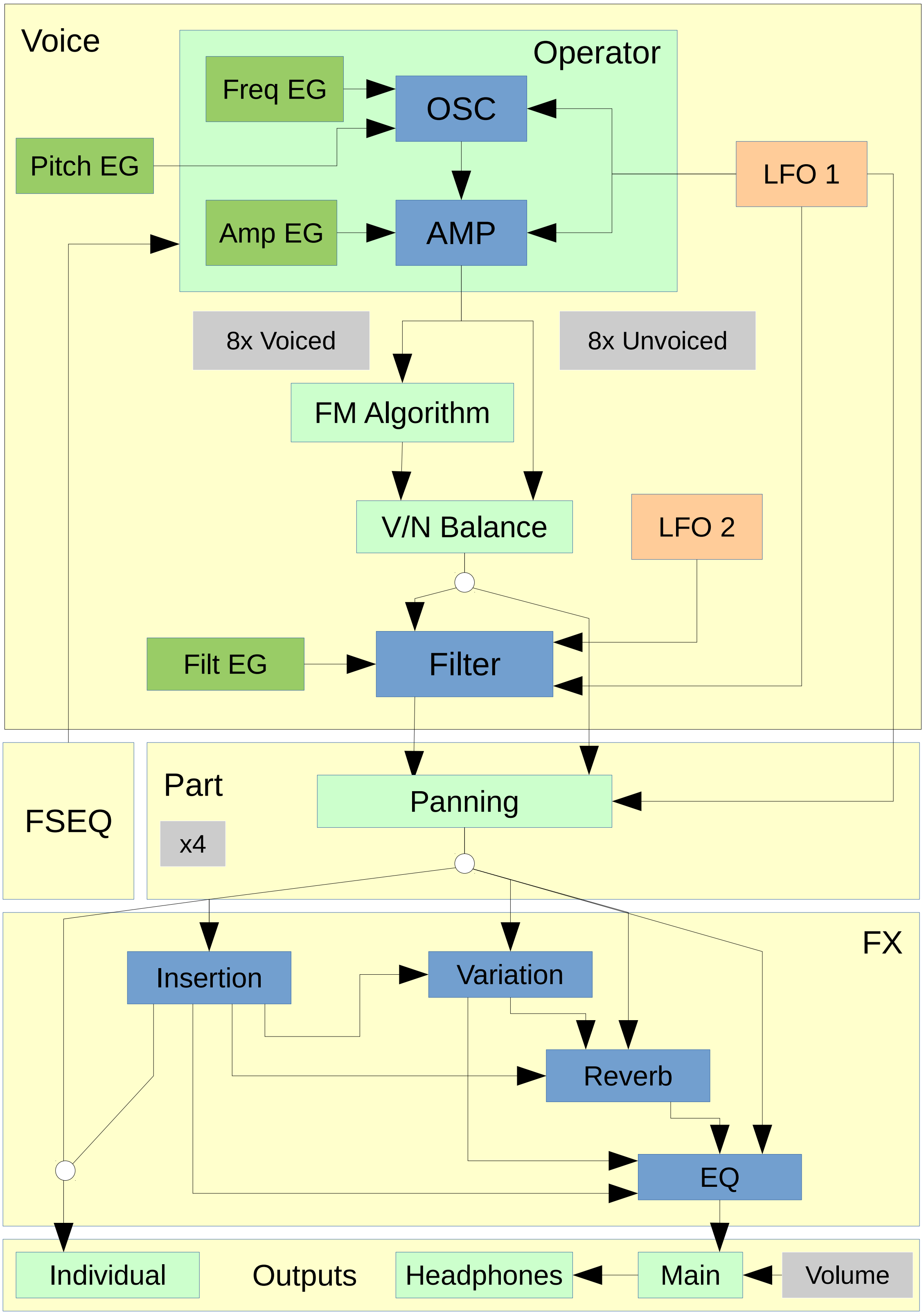
SysEx messages are routed according to the parameter address specified in the MIDI Data Tables included at the end of Yamaha Data List manual.

𝄞

Good luck with your Yamaha FS1R

Michał Wiernowolski, September 2015

Sound Generation



Sound Control