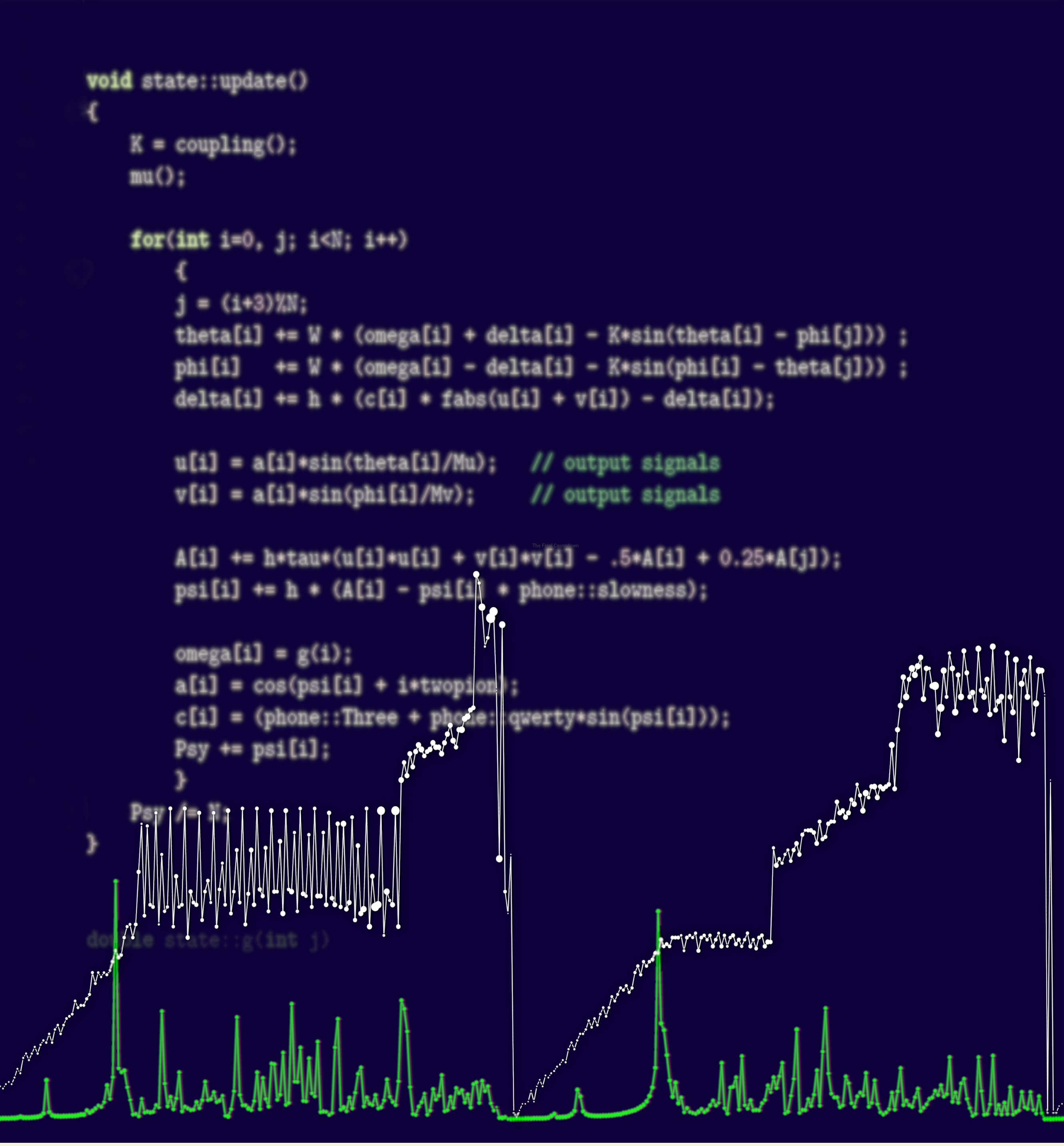


MERE021

KOLMOGOROV VARIATIONS

(Eleven Hard Pieces)

Risto Holopainen 2021

MERE

----- KV (11HP) -----

1.	Hello World!	1'42
2.	Statement	4'14
3.	Bâtir et Démolir (ZAD)	4'01
4.	Auto-detune	4'46
5.	Megaphone	5'20
6.	La Follia	2'37
7.	Speed of Thought	7'19
8.	The Wastebasket of Timbre	8'08
9.	Origami	3'09
10.	The Quasi-periodic Table	4'39
11.	The Final Countdown	6'15

Composition, programming, mixing, and mastering by Risto Holopainen

Støtta av Kulturrådet
Fond for lyd og bilete

KOMPONIST FORENINGEN

Komponistenes vederlagsfond

----- KV (11HP) -----

Similar to improvised music, algorithmic composition seems to have an affinity for process and flux rather than definitive finished versions of the work. Code can be refined endlessly, structures reconfigured, parameters varied. At some point the process has to stop for the sake of documentation or a performance. In *Kolmogorov Variations* the endless exploration of possibilities had to stop at an earlier point, because the eleven tracks were going to be submitted to a secondary phase of analog and acoustic sound processing.

The Eleven Hard Pieces are about algorithms, complexity, and autonomous systems. Every level of the pieces, from sound synthesis to large scale form, is the result of programming.

Autonomous systems in music

An autonomous dynamic system is described by a state variable, typically a vector, and a function that determines how the state evolves in time. The state variable contains all there is to know about the system at each moment. There is no input from the outside.

Typically a composer is responsible for putting notes in the score in a certain order, or shaping and positioning the sounds in an electroacoustic work. Making music with autonomous systems involves the indispensable step of constructing the system, which can be quite remote from the concreteness of music making.

Acoustic instruments can be modeled as a dynamic system with input in the form of an excitation, such as bowing a string, blowing air into a wind instrument, or striking a membrane. In the absence of excitation no sound is produced. Musical instruments also have "control inputs" allowing the musician to shape pitch, loudness, and timbre. An instrument without realtime input might be called an *autonomous instrument*. Such instruments produce sound without any means of external control, which is not exactly what we would normally think of as an instrument. Autonomous music systems can be regarded as an approach to algorithmic composition insofar as the system is realised by programming a computer.

Composing music is about making decisions. The medium, technique, duration, form, and content have to be chosen in full detail, or the musicians will complain, if it's instrumental music. Even a title must be chosen, no matter how unrelated to the piece itself. The more choices to be made, consciously or not, the more of the composer's taste or personality can be assumed to have influenced the music.

Generative and algorithmic approaches to composition seemingly remove several of those choices and leave them to the algorithm. Indeed, escaping the burden of choice may be an important reason for some composers to turn to automated composition. Another reason may be a wish to reach beyond one's ingrained knee-jerk aesthetic instincts. A third reason is the delight of unexpected discoveries when a system generates some surprisingly beautiful output, or simply anything unexpected. A fourth reason, perhaps less common, is the scientific curiosity and mindset of the explorer who wants to understand the mechanisms at work.

In any case, the escape from choice is obviously illusory. Algorithmic composition merely inserts an intermediate layer between the composer and the music. While it is a means to express musical ideas that otherwise would be impossible or hard to formulate, the algorithm is not like a ghost writer suggesting musical expressions one cannot stand for. - No, actually it does. Those failed attempts, of which there are plenty, quickly end up in the computer's virtual trash can.

----- KV (11HP) -----

```
;      wt  :fb:? sh
;      tqxx c# :xpe
;       x--|jf|a-wl
;       x--|jf|a-wl
;      : ||amiad|zo
;      fcjjmyumpjb
;      b#ffiuqilf|w
;      : ||amiad|zo
;      ?:aadpldga!r
;       x--|jf|a-wl
;      spwwzb|z!wod
;      hello world!
```

----- KV (11HP) -----

Program Notes

In these days deep learning, big data, and sophisticated algorithms claiming to be artificially intelligent are popular subjects in technologically advanced music. Hand-crafted algorithms that can be sketched with pen on paper may not cause much of a stir in that context. But it must be emphasised that the impressive results obtained by these state-of-the-art systems come at great cost. Massive amounts of input data in the form of pre-existing music needs to be analysed; the training of neural nets or optimisation of evolutionary algorithms require enormous amounts of computation. Furthermore, works relying upon corpora of existing music are necessarily of a derivative nature.

Kolmogorov Variations begins from scratch, if not from a perfect void. Each one of the eleven pieces is composed by writing a specific program whose sole purpose it is to generate that particular piece. None of the programs accept any input, or rather, the input is hard-coded into the programs.

Kolmogorov complexity, also known as algorithmic complexity, was introduced independently by Kolmogorov, Solomonoff, and Chaitin in the 1960's. It measures the complexity of an object in terms of a computer program that produces a description of the object. Specifically, the Kolmogorov complexity is defined as the length of the shortest computer program that produces the description text string.

Kolmogorov Variations plays with the idea of achieving much with little; complex music preferably realised by short programs. In other words, the algorithmic complexity as measured by code size would be small, whereas the musical complexity should be as high as possible. Certainly the code could have been made shorter, and in some cases the music could have been more complex. But the Eleven Hard Pieces are so much more than a complexity contest, they also enter into curious explorations of arcane synthesis techniques or representations of time. Most of all, they are pieces of music in their own right.

*

All pieces were realised following the same procedure:

1) Writing the program
2) Post-processing

Programming is not always such a rational and goal-directed process. Most pieces grew out of a lot of trial and error, successive additions, and sometimes lengthy processes of debugging to get the code to run at all.

The soundfiles generated by running the programs are of a somewhat crude quality. Much of their sterile purity has been softened in the post-processing phase.

The pieces were processed acoustically and by analog effects. Acoustic processing was applied on all tracks except *Bâtir et démolir*. Mono or stereo tracks were played through a pair of loudspeakers clad with aluminum foil, as well as a pair of exciters attached to a drum and a guitar, respectively. This sound, with all the buzzing, rattling, and strong resonances from the vibrating objects, was recorded and used in the mix.

Additionally, a eurorack modular synth was used for analog processing, including simple lowpass, bandpass, and highpass filtering, spring reverb, distortion, ring modulation, frequency shifting, and flanger effects. In the final mix the processed material is combined with the raw output from the programs. All pieces maintain their original form, the editing has been limited to equalisation and mixing the processed recordings and original soundfiles.

Hello World!

The standard first program every programmer writes as they learn a new language prints this greeting and quits. Its musical equivalent translates each letter to a pitch adjusted to a tuning with a constant 18/17 inverval between semitones, while it walks through a simple permutation.

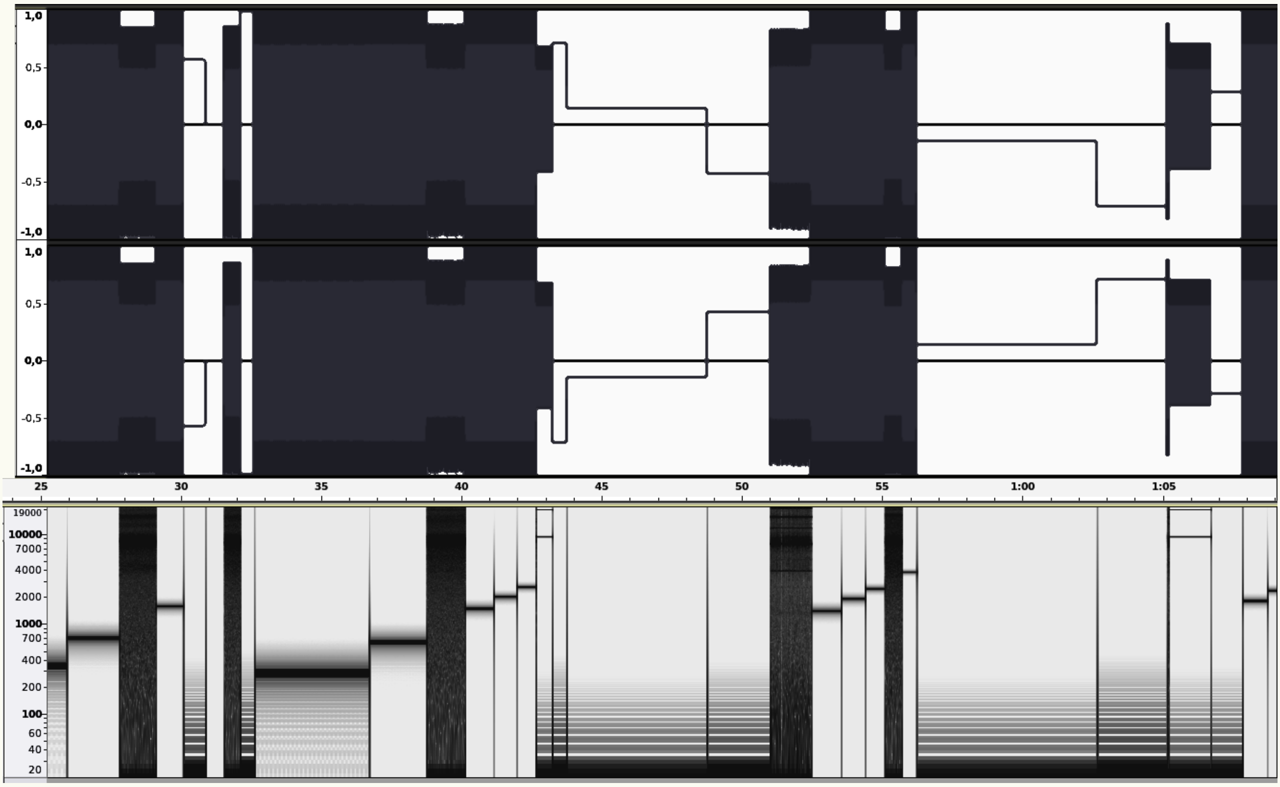
Statement

Waveforms are concatenated from segments of constant amplitude. Similar to other examples of so-called nonstandard synthesis, the waveform undergoes a constant transformation.

Bâtir et Démolir (ZAD)

The building blocks are sine tones, chaotic noise, and DC offsets in segments of varying length. This raw material is written to a soundfile, then read by another program that demolishes it by applying several binary operations to the signal and a delayed copy of itself. The binary operations include linear and saturated mixing, ring modulation, and taking the minimum or maximum of the two signals. ZAD stands for *Zone à defendre*, and the title refers to the sprawling architecture hastily built, soon to be demolished, at occupied ZADs such as that of Notre Dame des Landes in France.

----- KV (11HP) -----



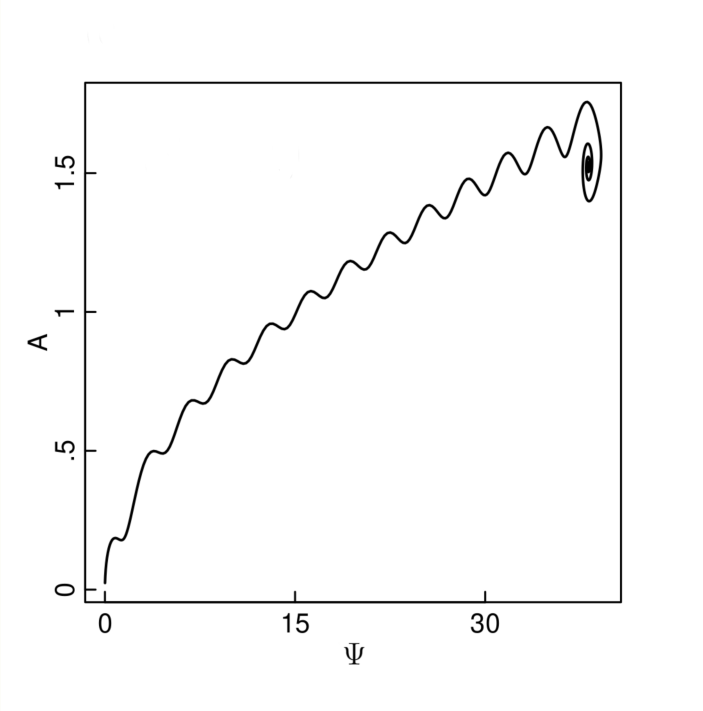
Bâtir & Démolir: Waveform and spectrogram of a part of the piece from the intermediate stage.

<u>Auto-detune</u>

The program that realises this piece is rather complicated, suffice to say that the title refers to a system of ordinary differential equations where two oscillators constantly de-tune themselves.

<u>Megaphone</u>

A large system of ordinary differential equations comprising 65 variables move on different time-scales. The system is partly related to the detuning system in the previous piece.



Megaphone: Two of the variables spiralling in towards a black hole as the piece gradually loses momentum.

La Follia

Not easily summarised, a system of five oscillators and simple feature extractors such as envelope followers and zero crossing rate estimators are used in a feedback network.

Speed of Thought

Randomness, despite being such a useful generative source, has been avoided in all pieces but this one. Random numbers are generated in an unconventional way by calling the function clock(), which measures durations between two function calls. These durations vary in unpredictable ways, depending on processes the computer may run in the background.

----- KV (11HP) -----

Random numbers are used at all levels of this piece: to generate crackling noise and waveforms, to choose pitches and durations. The sound synthesis is similar to that in Statement, consisting of concatenated linear segments. The whole thing would have been highly efficient, if it weren't for the fact that the method of random number generation turned out to be extremely slow.

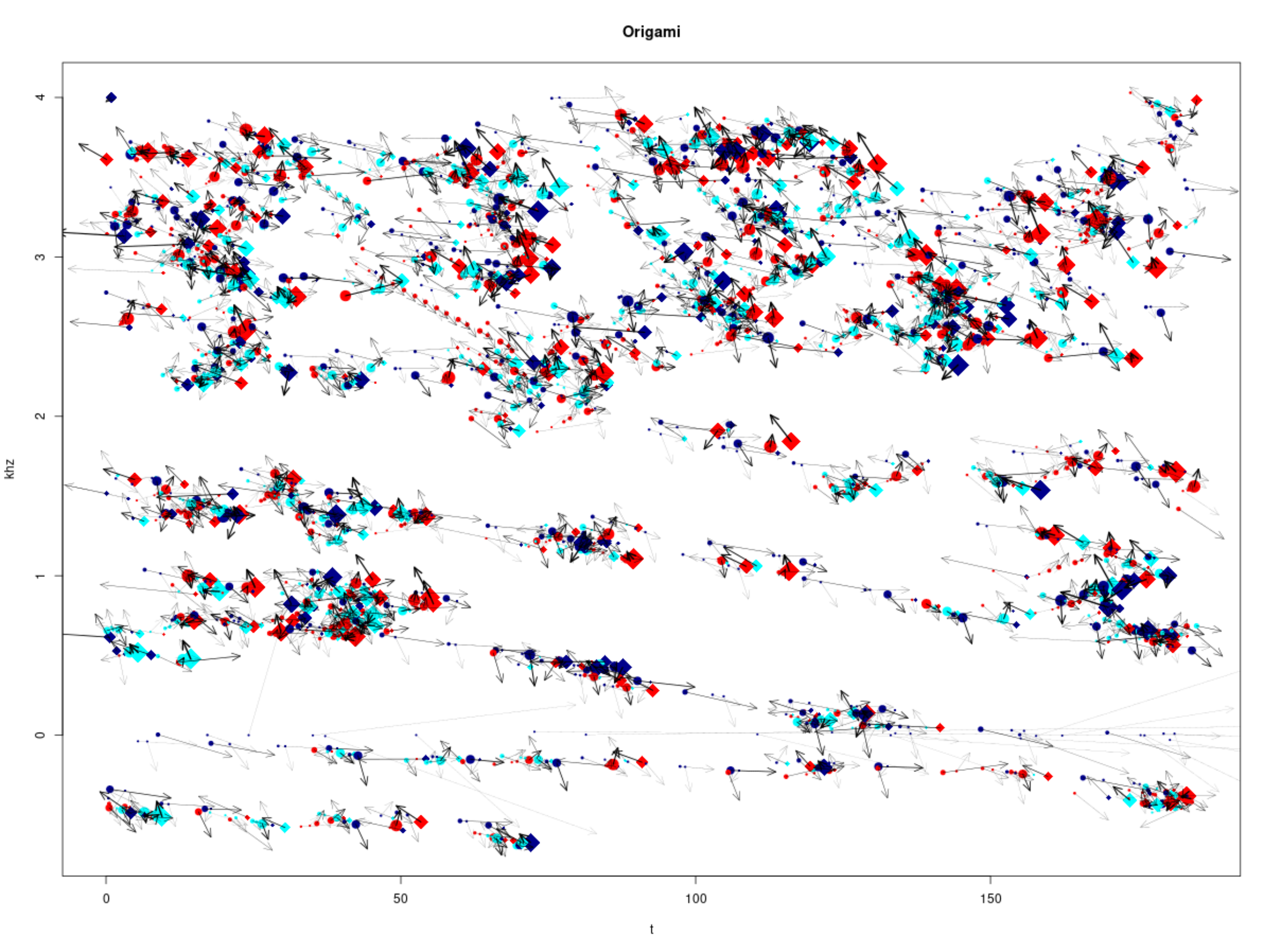
The Wastebasket of Timbre

Instruments produce different timbres depending on register and the manner of articulation. Simultaneous tones may blend to produce a new timbre. This piece uses wavetable synthesis and more than 20 different waveforms, each with their unique timbre articulating something reminiscent of a serial structure. Two tuning systems are used in parallell, a pure intonation and a Pythagorean tuning, both with added tempered tritone intervals.

Origami

Soundfiles are conveniently written from the beginning to the end, but in this piece time is permitted to change direction and zigzag back and forth. The end and the beginning are also joined like a cylinder, such that when the signal is written past the end of the file it continues from the beginning. A tapestry of decaying sinusoids is weaved and folded through the piece.

----- KV (11HP) -----



Symbolic score for Origami. Time follows the direction of the arrows; sometimes forward, sometimes backwards, the beginning and end being joined.

<u>The Quasi-periodic Table</u>

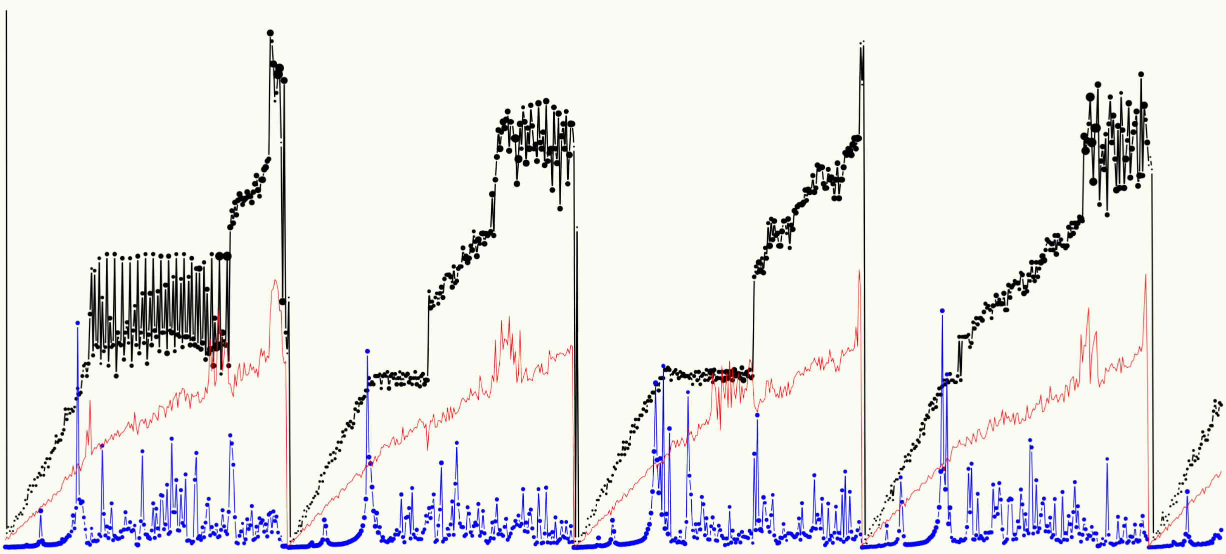
A linear ordinary differential equation with quasi-periodic coefficients is responsible for the sound synthesis. Quasi-periodicity refers to the use of mutually incommensurate frequencies. At certain points new frequencies are chosen, resulting in a slowly evolving melody.

----- KV (11HP) -----

<u>The Final Countdown</u>

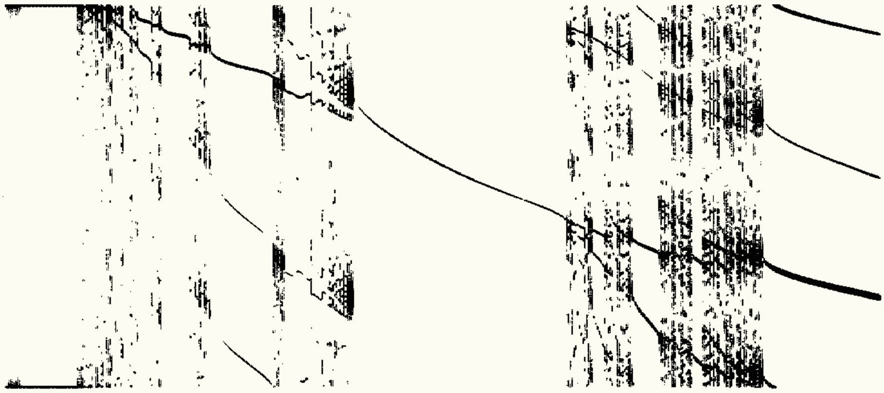
A large, unwieldy, and unstable program is built around three systems of cross-modulating FM oscillators, feature extractors that analyse the signal, and parameters that are intermittently recorded and read in a loop. A definite failure to write a short program!

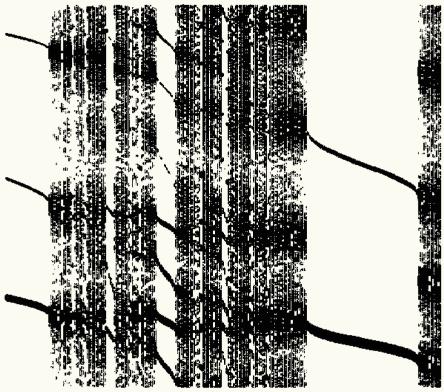
Risto Holopainen
Oslo, December 2021



The Final Countdown: time evolution of three variables.

----- KV (11HP) -----





Source code for all eleven hard pieces, licenced under GNU GPL v.3, is available at https://ristoid.net/prog/kolmogorov.html

Further reading:

Holopainen, R. (2021). Making Complex Music with Simple Algorithms, is it Even Possible? Revista Vórtex, Curitiba, v.9, n.2, p. 1-13, November, 2021.

Holopainen. R. (2021). Algorithmic Composition by Autonomous Systems with Multiple Time-Scales, Divergence Press