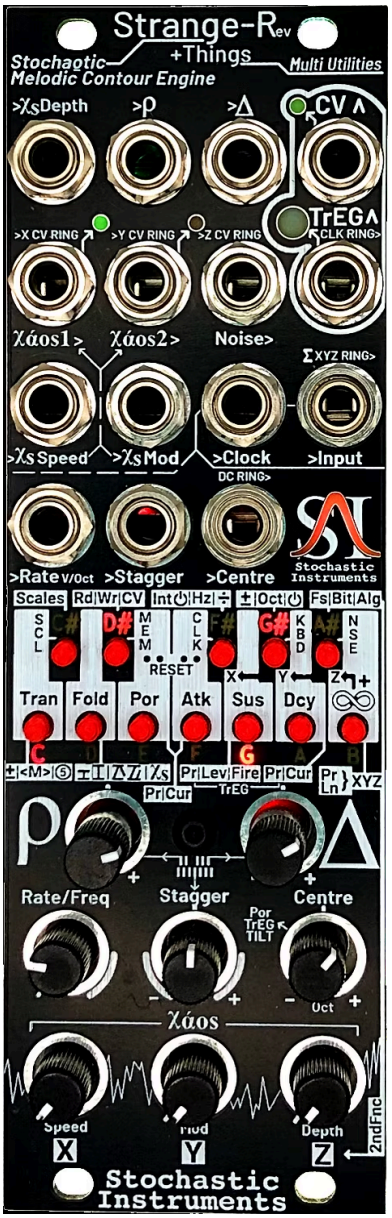
Stochastic Instruments
Rethink Random

Strange-R: A Melodic Contour Engine

Stochastic Instruments Ltd.



Rethink Random | ***Create*** Chaos | ***Perform*** Process

Installation and Safety

Strange-R uses the standard Eurorack power connector with the red edge of the ribbon cable indicating -12V (supplied—however, always remember to check *any* power cable for faults).

Always ensure power is off before installing any module or connecting/disconnecting the power cable, that you do not touch any electrical terminals or rear circuitry during installation and are properly statically discharged, and that your power supply is sufficiently current rated to serve all the modules connected to your bus board (you can check this on modulargrid.net).

Stochastic Instruments (SI) guarantees this product will be free of material defects and construction faults for a period of two years from the date of purchase if and only if bought direct from us. Malfunctions resulting from any misuse of the product, user modification of the circuitry, faceplate or firmware, the application of incorrect power supply voltages or connection of the power cable, design faults of third-party power cables or bus boards, or any other factors determined by SI to fall under the definition of 'user error' are not covered by this warranty.

If you have bought both 'hats', please pay special attention to the section below on Swapping Hats to avoid inadvertent damage.

For the period of the warranty, defective units bought new from SI whose fault is deemed by SI not to be caused by 'user error' will be repaired or replaced at our cost (including shipping to and from us), on presentation of valid proof of purchase from SI by the first owner. Units not bought from SI are not covered by this warranty. While we will undertake repair work of 'used' and/or 'user error' SI units, this will be charged and the customer must cover postage to and from us. Your statutory rights are not affected.

SI implies and accepts no responsibility for harm to the user, or damage to equipment caused through any operation of this unit.

Wait List: https://forms.gle/7WyFvb4PR82zyixM7
https://forms.gle/PQCy6tyJhFcFfojf7

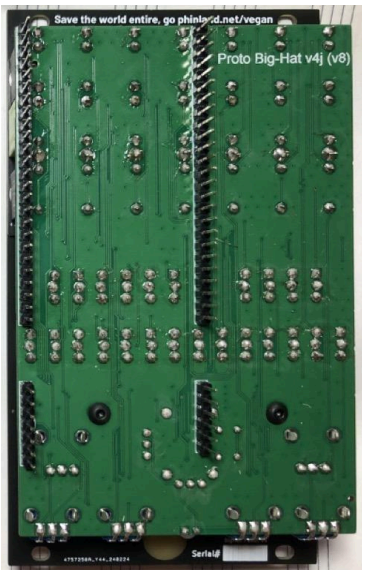
Contents

Two Editions: Live (SmallHat) & Studio (BigHat)

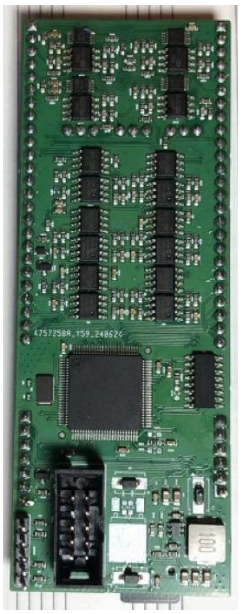
One of our project aims was to explore the tension between footprint and functionality. The more functions a module has, the larger it must be to provide the user interface to control them. At the same time, live rig rack space is precious and requires an ultra-compact design to fit into 104hp budget airline carry-on luggage restrictions. Our solution to balancing these is to provide Strange-R in two interchangeable sizes, the **SmallHat Live Edition** and **BigHat Studio Edition**.

SmallHat uses micro pots for ρ and Δ and puts several CV inputs/outputs on TRS jacks (where the most common 'default' is on the tip and the less common 'second function' is on the ring accessed by a Y splitter).

In contrast, **BigHat** uses a joystick for ρ and Δ in a single control and has a dedicated jack for each CV input/output. In addition, pushing the joystick in fires a TrEG (this control has no equivalent in the Live Edition).



The innovative part is that the same **MCU board** can be connected to *either hat and retain its memories.* With both hats and one MCU board you can prepare a live set on the BigHat, save the scene(s), transfer the MCU board to a SmallHat in your live case and *all your work will transfer across now accessible in the more compact footprint*.



All functionality is duplicated between the Big and SmallHats so throughout the manual we will interchange graphics between them.

For instructions on swapping hats see section at the end of the manual. Please also be advised that the SmallHat can be a snug fit between the rails of some racks. Once in it will slide perfectly easily but you may need to insert the top of the module first and then 'pop' the legs of the bottom row of pots past the lower rail. To assist with this and protect against shorts, the bottom row of SmallHat pot legs come insulated with Kapton Tape.

Overview

Congratulations on purchasing the Stochastic Instruments Strange-R!

With proper care, this unit will give you years of trouble-free recursive complexity.

Strange-R is a totally new way to do sequencing. It's extremely powerful at generating recursive melodies, loops and randomness but how it does so is unlike any other module you'll have used before. The majority of its melodic power comes from just two controls…

ρ (Rho = Correlation = **Melodic *Direction*)**

Δ (Delta = Deviation = **Melodic *Jumpiness*)**

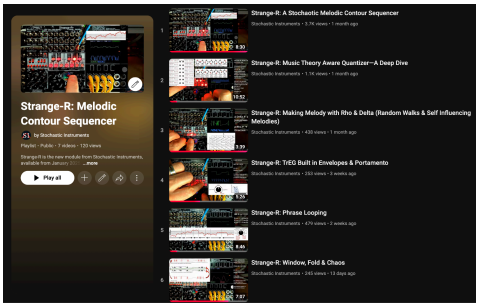
…which both originate the melodic material and then control how it's manipulated by 3 on-board 'phrase loopers'.

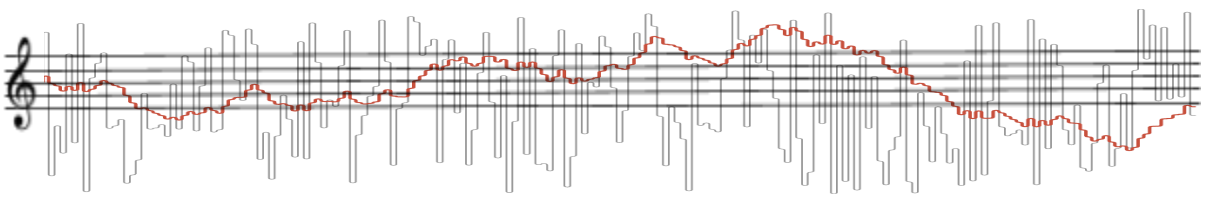
We recommend first just diving in: manually set the quantizer notes to your favourite scale/arpeggio, twiddle around with ρ and Δ, grab a Phrase Loop, and then move *that* around with the same two controls:

Most of your feel for the module will come from that, but eventually you'll want to explore Strange's deeper functions, which is where this manual comes in. Because Strange's approach is new we've explained not just its operation but the *context* underlying its design. To help you find what you want to know we've divided the manual into 3 sections:

I. Quickstart—gets you up and running
II. The Strange Concept—provides the conceptual background
III. Operation Manual—details the controls

Happy exploring and please also see our [YouTube Video Manual playlist]!

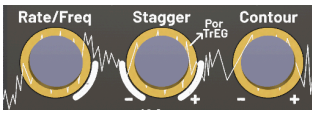




I. Quickstart

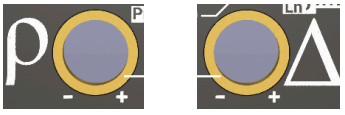
Strange-R makes generative melodies that sound like human melodies. It also invites—and rewards—exploration. Patch CVOut to your favourite VCO and TrEG Out to your best filter's CVIn. Your filter will pulse at the clock rate while the VCO will make random untuned pitches which you can shape with the two main performance controls, ρ and Δ.

Rate—determines the speed at which the TrEGs will fire
Stagger—speeds up/slows the clock for higher/lower notes
Centre—a brute force pitch shift useful for 'centring'

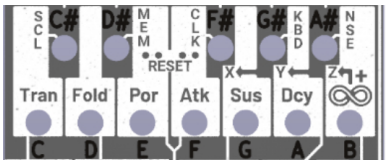


Clocking: Strange defaults to Internal Clock. To switch to External Clock hold note F#, turn Χάος Speed ('X') clockwise and connect to ExtClk in.

Correlation—rho ρ makes the melody tend to rise or fall
Deviation—delta Δ makes the melody smooth or jumpy



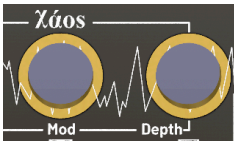
Without the quantizer engaged you'll see note lights dancing around to indicate the closest actual *note* to the untuned microtones being produced. To make something more traditionally tonal, engage some notes on the **Quantizer**. You can do this manually, note by note, or with a preset: hold **C# (SCL)** and turn the **X Pot** (bottom left). The single LED that walks up the keyboard shows which preset you're on: light 'F' and release C# to load a C Major scale.



Listen to the effect of different values of ρ and Δ. When you've got the hang of steering and shaping the overall characteristics of the melodic contours it's producing, try holding one of the engaged notes and changing ρ and Δ to alter its note-specific controls. Do this again with another note and you should notice little melodic cycles and eddies starting to appear.

Now capture a **Loop** by pressing B (∞) and F# (X Loop): the loop will continue to write for as long as you hold the two buttons. As soon as you release, the loop will start to play. Now dial in some ρ and hear the whole loop *contour shape* move over the scale notes.

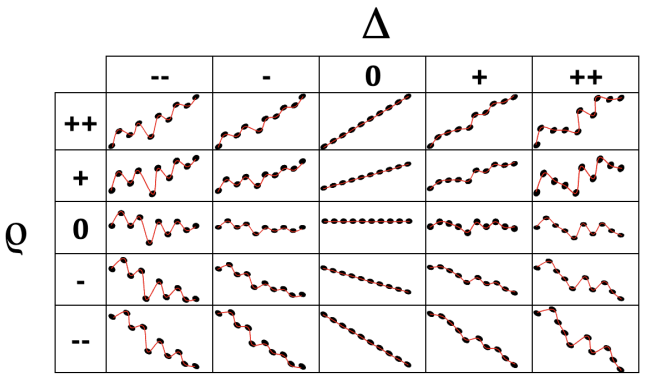
Hold B (∞) and turn the X Pot to about 50% to hear ordinary melody generation interspersed with your Loop. Introduce some χάος (Chaos) Depth to move them both around a little more gesturally.



Now hold **C (Tran)** and twist **Y Pot** to change the musical mode from C Major (Ionian) to something else (the different notes of all the scales *modes* will cycle through the quantizer). When you see one you like, release C to hear it: the melodic contour shapes remain unchanged but they now dance around the new related scale.

Hold **D (Fold)** and change **X Pot**—the overall range of the notes will reduce and will start wrapping back on itself. Finally hold **A (Dcy)** and change **X Pot** to give some notes a decay tail.

There are many more functions but knowing how they fit together is well worth the effort in making the most out of the module, so we strongly recommend reading on from here



Strange-R Cheat Sheet

					Χάος		
ρ	**Δ**	**Rate**	**Stagger**	**Centre**	**Speed**	**Mod**	**Depth**
Rising vs Falling	**Smooth vs Jumpy**	**Tempo**	**Higher Notes Faster vs Slower**	**Course Centring**	**Wiggle speed**	**Wiggle oddness**	**Wiggle amount**

Grab Loop: X=B+F# Y=B+G# Z=B+A#

Loop Length/Reverse: Longhold B+X / B+Y / B+Z

	C Tran Scale Transforms	**C# SCL** Preset Scales	**D Fold** Folder / Wrapper	**D# MEM** User Memories	**E Por** Portamento Functions	**F Atk** TrEG Attack	**F# CLK** Clock Functions	**G Sus** TrEG Sustain	**G# KBD** Keyboard Mode/Arp	**A Dcy** TrEG Decay	**A# NSE** Noise Source	**B ∞** Phrase Looper
X	+/- semitone transpose	Bank X standard scales	Window Width	Read Location CDEFGAB	Portamento Probability	Attack Probability	Internal / External	Sustain Probability	+/- semitone transpose	Decay Probability	Sample Rate	X Loop Probability / Length
Y	Modal Shift	Bank Y unusual scales	Reflect vs Wrap Probability	Write Location CDEFGAB	Portamento Curve /Random	Attack Curve /Random	BPM / Hz	TrEG Level /Random	+/- octave transpose	Decay Curve /Random	Bit Depth	Y Loop Probability / Length
Z	Circle of 5th Shift	Bank Z composer scales	Window pre/post Chaos	CV Assign >X (F#) >Y (G#) >Z (A#)	-	-	Clk Divide	TrEG Probability / Rhythm Mode	OFF / Clocked / Free	-	Analogue/PRNG Algrithm	Z Loop Probability / Length
ρ	C ρ	C# ρ	D ρ	D# ρ	E ρ	F ρ	F# ρ	G ρ	G# ρ	A ρ	A# ρ	B ρ
Δ	C Δ	C# Δ	D Δ	D# Δ	E Δ	F Δ	F# Δ	G Δ	G# Δ	A Δ	A# Δ	B Δ

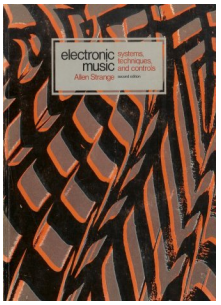
II. The Strange Concept

Background

We're very proud of how our first module, SIG, was received by the Eurorack community. With our 'difficult second album' we've tried hard to balance continuity of approach with something individual enough in sound, function and interface to be a worthy successor—distinct and complementary without replacing or duplicating it.

Like SIG, Phin's original (2018) Strange concept was intended to be manufactured by omsonic but only a tiny prototype run (left) ever materialised from them. We therefore wanted to revisit the idea as SI in digital form to offer incomparably more depth than the original analogue version could achieve—hence R for 'Revision'.

We also wanted to approach a generative system from a different angle to SIG. SIG subverts the 'fixed steps of parameterised pitch' approach of standard sequencers with 'fixed pitches of parameterised probability'. Its power lies in the complex polymetric relationships its 4 tracks can create, and how its pitches and durations can be *tonally organised* by their relative weightings. However, they are not *phrasically organised:* SIG's events are a linear stream of notes, unrelated to each other across time. **With Strange-R we wanted to explore a single track system whose power lay in more recursively complex, self-referential note transitions and higher-order motivic groupings of pitch events: the poetry of related phrases and sentences rather than a linear stream-of-consciousness.**



Strange's name is both a description of the module's *behaviour*—its multidimensionally interdependent unpredictability and its generative spontaneity—and a homage to *Allen* Strange (1943–2008), composer, performer and author of one of the foundational texts of our discipline, Electronic Music: Systems, Techniques and Controls (1972)[1]. It's not however an implementation of any patch within it (e.g. the 'Dream Machine'). Rather, it is an embodiment of the book's ethos and overall approach—*modelling compositional processes by combining discrete parts into an organised whole.* It provides an ultra-compact, highly CV-able generative system whose internal complexities bely an extreme simplicity of basic operation: performative immediacy, gestural potency and melodic expression. By exposing its internal sub-modules to the front panel **Strange-R also doubles as a powerful multi-utility unit, offering a range of core functions ('Things') far beyond its footprint and price point: hence its full name, Strange-R +Things.**

Strange-R has consistently surprised us over its many design cycles with the unexpected sophistication of its behaviour. Most excitingly, the final result exhibits a level of emergent musical complexity *which we didn't explicitly program*. Algorithmic personality that invites exploration is always an encouraging sign. We're really pleased with it. We hope you are too.

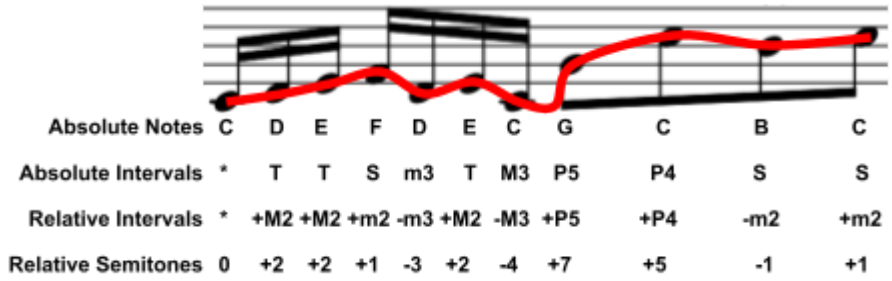
[1] Long and lamentably out of print but now available once more thanks to the dedication of Prof. Jason Nolan and colleagues. Phin found his original edition for 50p in a jumble sale in the early '90s *long* before he had the faintest clue of either its sale value or how profoundly it would affect his eventual career path.

Melodic Contour Sequencing

Traditional sequencers, generative systems, and random sources operate at the level of the *individual note event*, and offer direct compositional control only over *which note* will fire, and *when*. Even our own Stochastic Inspiration Generator, while wholly probabilistic in operation, operates entirely linearly on a note-to-note basis. Strange is different.

If we examine almost any piece of music we see that composers make pieces not from endless streams of individual notes but from *phrases*—larger collections of notes which repeat, transpose and permute. In other words, **poetry is formed not from an endless string of individual *letters* but from *phrases* and *sentences* who *rhyme, repeat* and *recombine*.**

Consider this phrase which opens Bach's Zweistimmige Inventionen in C BWV 772:



Instead of absolute notes, we could *abstract* the music into its series of relative interval jumps…

C:[0 +2 +2 +1 -3 +2 -4 +7 +5 -1 +1]

…to provide the 'shape' of the contour shown with the red line. This is its 'interval series'. We can then see how Bach moves this phrase around, and when he does so how he has to alter some of the intervals to fit into the scale ("diatonic correction"). For example, we can't just brute force the contour up +2 semitones to start on D *because it wouldn't work in the key of C major:*

[C D E F D E C G C B C] + 2 = [D E F# G E F# D A D C# D] ***wrong!***

…F# and C# aren't in the key of C so we must *diatonically correct* them back to the scale…

[C D E F D E C G C B C] >> [D E F G E F D A D C D] ***correct!***

…which then changes the *intervals*…

C:[0 2 2 1 -3 2 -4 7 5 -1 1] >> D:[0 2 1 2 -3 1 -3 7 5 -2 2]

It's the *same contour* ***shape*** *'walked up the scale':* different notes and different intervals make it *sound slightly different yet completely related to the original, and still within the same parent key.*

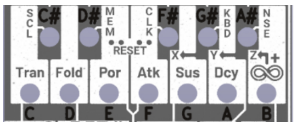
Very simple, *very* powerful: your ear is *astonishingly* good at recognising these contours and will even do so despite their starting note and intervallic distances changing constantly in the piece[2].

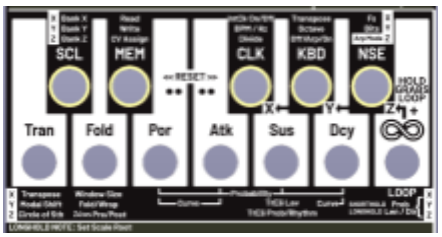
[2] Phin's academic background was Auditory Neuroscience where he specialised in Auditory Object Formation. Many of those ideas about higher level abstraction also informed Strange-R's design concept.

The fundamental design idea of Strange-R then is that it…

1. **Generates raw abstracted unquantized melodic contours <u>first</u>…**
2. **Stores these as phrase loops <u>second</u>…**
3. **Moves the phrases around with the same controls used to create them <u>third</u>…**
4. **Routes them through its quantizer to diatonically correct them <u>last</u>.**

This is a small but incredibly important point: by placing the quantizer last Strange-R always applies the diatonic correction needed to 'walk a phrase' around the scale. The quantizer is thus an intrinsic part of the module's performance controls in its own right.

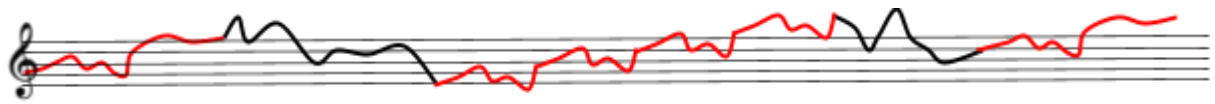




Its micro keyboard can be controlled manually in performance but Strange can also load preset scales, apply shifts, transpositions, modal interchanges (switching instantly between modes of a common root note), inversions (turning scales upside down) and key-centre modulation through the Circle of Fifths—all live in performance with a single button press.

Strange moulds and morphs the same phrasic DNA into new tonal territory, graphed into arpeggios, transposed, reflected and wrapped into new forms, while still retaining the motivic continuity of the original melody.

Contour loops intersperse between free-running stochastic melody, just as composers do:



Both bold and subtle musical gestures are available via live manipulations of the controls which are designed to give the maximum expressive power with the minimum fuss. The 3 phrase loopers allow you to capture Strange's free running output and then walk that through the scale as a phrase, riff or motif. Superimposed on that, the χάος function can shape higher level 'structural' aspects of the melody between periods of quiescence and frenetic activity.

The result is musical output which is both recognisably patterned yet never predictable, highly responsive yet always surprising. Strange-R is therefore…

- **not a traditional sequencer** — you don't 'program' it and it only repeats if you tell it to
- **not a probability sequencer like SIG** — you don't set individual note probabilities
- **not a pure random source** — it outputs recursive patterns which related and 'rhyme'
- **not a pure chaotic source** — identical initial conditions don't produce identical output

Instead Strange-R live composes *with* you, responding in performance to the fundamental parameters of pure melody—scale, step, coherence, contour and phrase. Let's find out how.

Strange-R is a Stochaotic System

A ***stochastic*** process (/stəˈkastɪk/ from the Greek στόχος for '*aim*' or '*guess*') is one governed by controlled amounts of randomness (as in SIG). A ***chaotic*** (χάος) process is completely mathematically determined: it contains no randomness at all, is *entirely* deterministic and runs identically every time...but its output can behave *as if random* or self organise *as if designed*[3].

A simple example is picking a lottery number vs predicting the weather.

Winning the lottery is entirely random: no amount of prior knowledge (like the complete history of all previous winning numbers) gets you closer to knowing tomorrow's numbers—so you can kiss goodbye to that 12 row 300hp system.

Predicting the weather is different. Knowing its state yesterday gives you a pretty good idea of its state tomorrow. Conversely, its exact state in 4 weeks time remains a mystery even though everything about its moment to moment transitions are based on simple, predictable, well understood processes.

Chaotic processes tend to be based on relatively simple rule systems from which highly complex behaviour *emerges* over time. While entirely deterministic, they are incredibly sensitive to initial conditions and external input, and often exhibit mixtures of repetition, stasis and rapid change, being relatively predictable over short timeframes and wildly unpredictable over long ones.

A system which contains elements of *both* fundamental behaviours is termed ***stochaotic***.

Strange-R is one such hybrid system. Using some colour codes to help us throughout the manual, its overall note-to-note transitions are **stochastically** determined but note-specific controls let you introduce **chaotic** dynamics that produce musical phrases and motives. A second distinct **chaotic** system determines larger-scale melodic gestures on top of this. Strange's **Phrase Loopers** then interact with the performance controls to produce **pseudo-chaotic** patterns and the pitch reflection system then folds and wraps the phrases which can then be recursively recaptured into a new **Phrase Loop**. Finally, note articulation via the **TrEG Envelopes** and **Portamento** offer a final **stochastically** determined stage in Strange's behaviour.

There's a lot going on there, so we'll break it down in the next section: think of Strange as a self-contained modular patch made up of smaller unique SI sub-modules.

[3] You, dear reader, are a chaotic system too! 😀

The Strange-R MicroPatch

Signal flow within Strange-R is relatively straightforward but it's important to understand the layout to get the most out of the unit. It's built from eight elements…

Pre-Quantizer

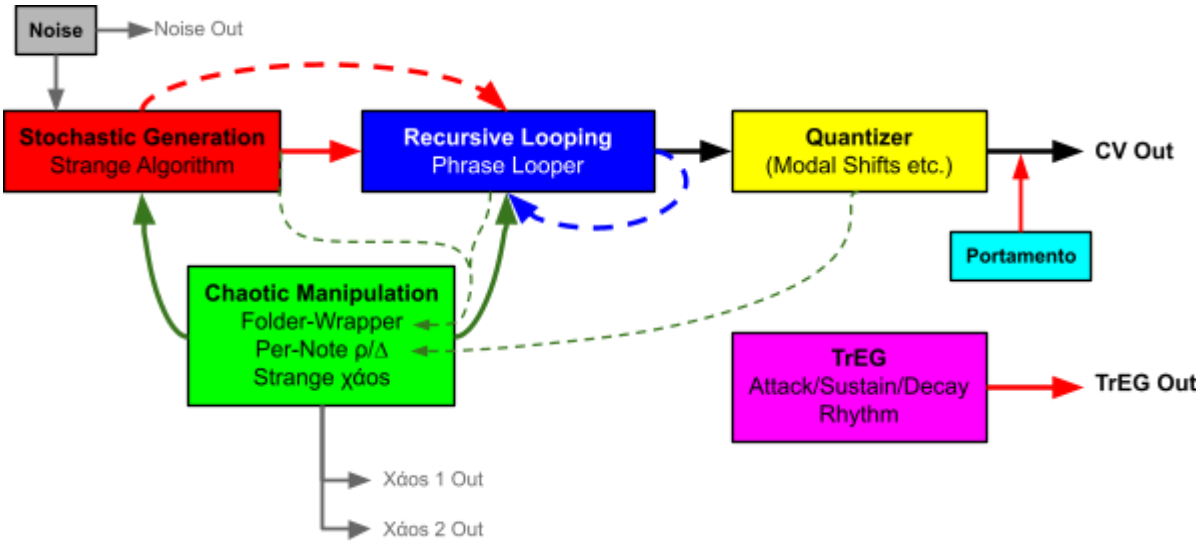
1. **Strange Algorithm**—an advanced Random Walk creating **stochastic** melodic contours
2. **Per-Note ρ & Δ**—provide recursive **chaotic** melodic 'eddies' to these basic contours
3. **X/Y/Z Recursive Phrase Loopers**—3 behaviourally distinct **stochastic** loop buffers
4. **Strange χάος**—**chaotic** macro-level melodic gestures steer both the contours *and* loops
5. **Folder-Wrapper**—**chaotic** transformation of the contour/loops inside a pitch range

6. **Theory-Aware Quantizer**—provides scales, modes and performative manipulation

Post-Quantizer

7. **Portamento**—**stochastically** determined portamento alters note inflection
8. **TrEGs**—**stochastically** determined rhythm/articulation (attack/sustain/decay/level)

The specific organization of these blocks is critical to Strange-R's design**:** a full patch-within-a-module of unique SI subunits[4] whose real power comes from the dotted arrows feeding back into the system:



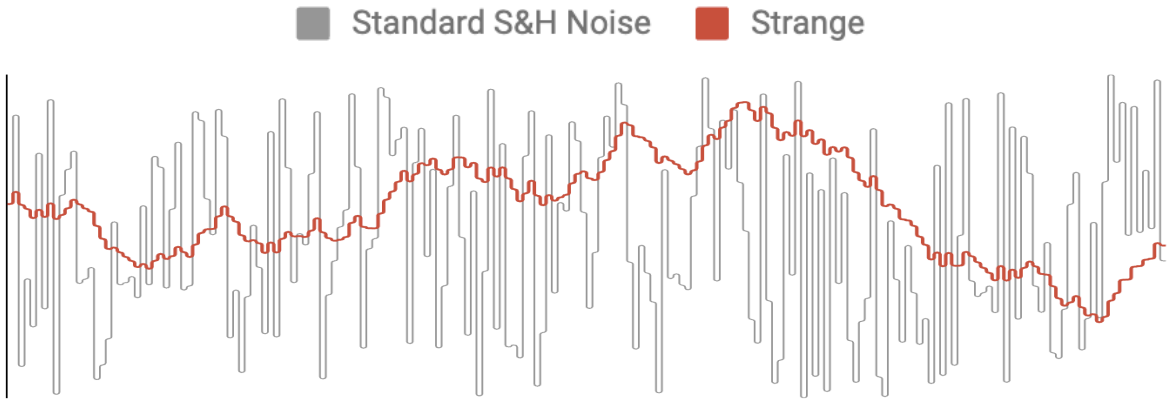
We'll now explore each block in turn…or you could just jump straight to section 3 Operation Manual and dive right in!

For Looping, Portamento (and pitch) and TrEG controls, see the Second Functions section below.

[4] Even assuming the sub-(sub-)modules were all only 2hp wide, and forgetting that every one is a unique SI module in itself, to break Strange-R out into its constituent units would occupy a vastly greater footprint and cost ten times the price, yet would do less as be harder to use!

1) Overall Melody: The Strange Algorithm

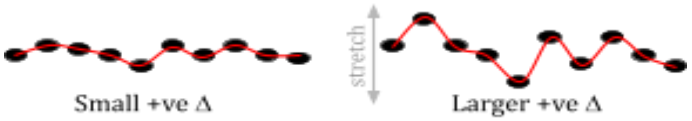
The Strange Algorithm is based on our own unique take on the **Random Walk**. While Clocked Sample and Hold Noise causes each event to sit anywhere in the entire voltage range, the classic Random Walk *deviates each event from the last by a random value*.



Our special implementation of the walk contains some special mathematical secret sauce under the hood to keep things interesting, and also introduces two new controls not found before…

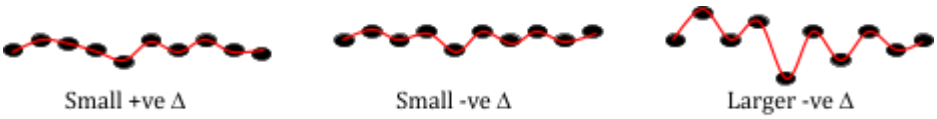
a) Positive Deviation (Delta Δ) stretches event-to-event jumps

i) Small Δ produces relatively smooth lines
ii) Large Δ produce relatively rough jumpy ones

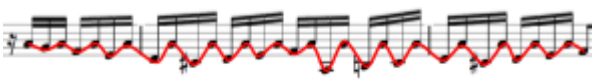


b) Negative Deviation (-Δ) *alternates opposing directions*

i) Small -Δ produces effects like filigree trills 
ii) Large -Δ produces **compound melody** (Ausfaltung or melodic fission).



A famous example of Ausfaltung is the fugue from the Toccata & Fugue in Dm BWV 565 probably by Bach

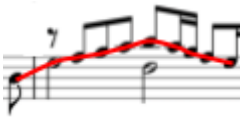


iii) +/-Δ doesn't mean jump up/jump down: +Δ deviates each note from the last in *either* direction à la Random Walk, whereas -Δ always *changes* direction (Random Hop!). The *maximum* amount possible in each case is determined by Δ size: that is, large Δ still permits small jumps but larger ones are more likely.

c) Correlation (Rho, ρ) controls the overall directional trend

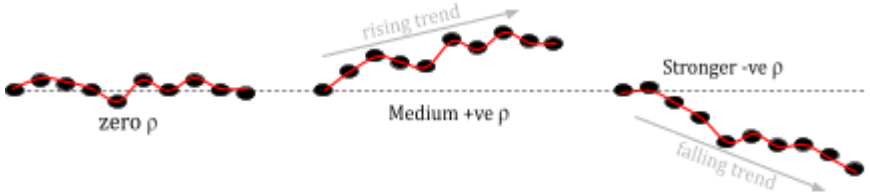
A defining feature of melodic *contour* is higher-order shaping over the individual point-to-point transitions. Each step of a Random Walk is equally likely to go up or down so the likelihood of *n* notes going in the same direction is $1/2^n$ This gets very small very quickly but real melodies do this much more often.

For example, the last of Bach's Goldberg Variations BWV 988 sounds completely natural but its 6-notes-up 5-notes-down motive contour would only happen by chance $1/3^9 = 0.0051\%$ of the time in a standard Random Walk (assuming 3 choices for each step of ascend/descend/repeat)!



The Correlation ρ control allows you to create melodic trends which still permit note-to-note movement up and down, but nevertheless cause the overall contour to rise or fall.

i) +ρ causes the overall melodic *trend to rise*
ii) -ρ causes the overall melodic *trend to fall*

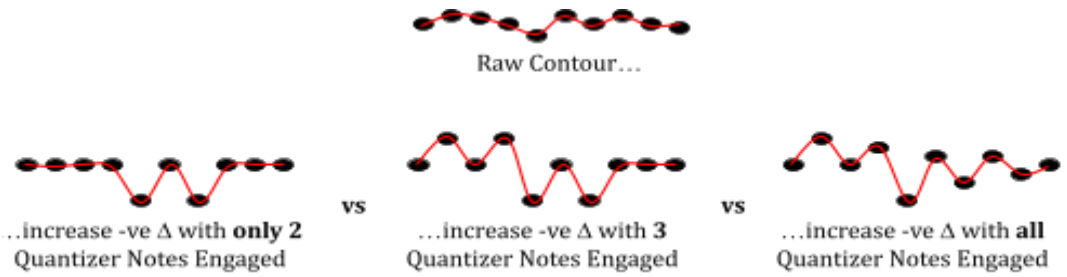


iii) The contour can still move rise and fall between events depending on the Δ value but the *overall trend* iterated over several events will rise or fall depending on the ρ value

iv) Small ρ values can be 'defeated' by large Δ values and large ρ values can 'defeat' small Δ values—again, the best way to get this is to play!

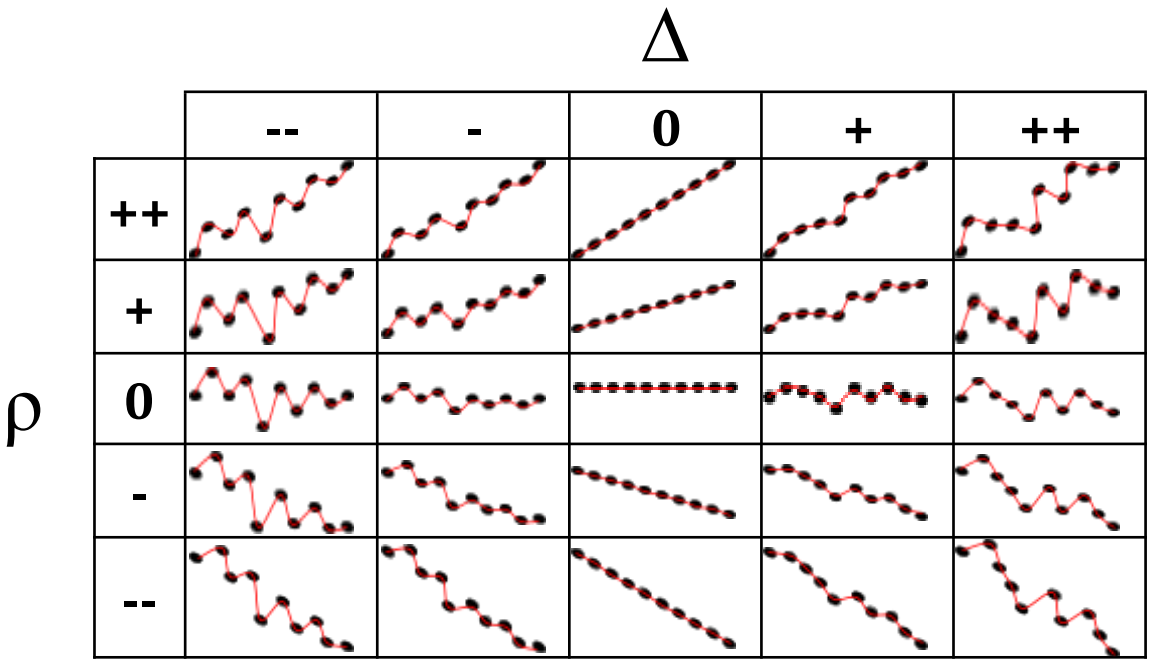
d) (Remember the Quantizer)

It's also important to realise that your quantizer settings will determine whether a note will change. Small Δ values will still cause the *algorithm's* output to change but because that happens *before* the quantizer, if the changes are smaller than the gaps between the selected quantizer notes then the quantizer output will remain on the same note.



Engaging or disengaging notes in the quantizer is therefore an important aspect of playing Strange-R because it reveals or hides smaller ripples in the algorithm's output.

Considered together then, Δ **controls 'melodic smoothness' and ρ 'melodic continuity'**. Maximum Δ yields classic S/H Noise and maximum ρ produces constant rising/falling runs depending on the **Folder-Wrapper** settings (see below), perfect for for arpeggiator sequences.



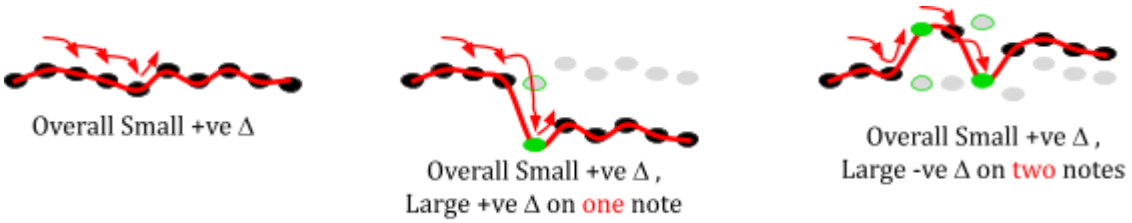
The Folder-Wrapper (see below) automatically takes care of the scale rising or falling into infinity so it always remains within sensible bounds.

Remember, Δ and ρ do not have to be set-and-forget values: indeed, they invite (and should receive!) constant dynamic manipulation through play and CV control to create spontaneous and developing melody.

2) Chaotic Per-Note ρ & Δ

Applied overall, the Strange Algorithm is purely stochastic. However, *very* interesting things happen when we apply *different ρ & Δ values to different notes*, achieved by holding the required note on the quantizer and setting ρ & Δ .

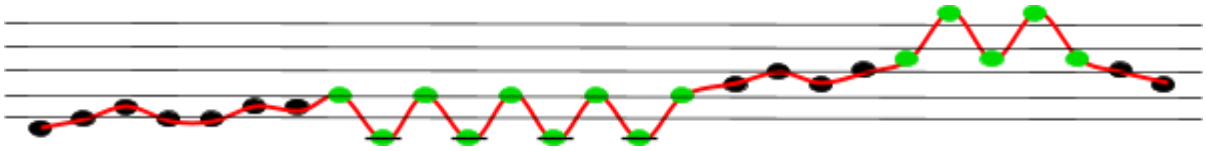
Consider a large Δ on one note given an otherwise small masterΔ :



Notice the family resemblance between the original contour and the transformed versions: they are different but also retain aspects of the original. In short: they are ***theme and variation.***

This behaviour is even more striking when ρ is applied to single notes, particularly where we combine *opposing directions* (+/- ρ) with small amounts of Δ . Consider the following settings:

- +ρ of about a rising Perfect 5th on **C**
- -ρ of about a falling Perfect 5th on **G**
- +Δ of about a whole tone
- C scale on the quantizer



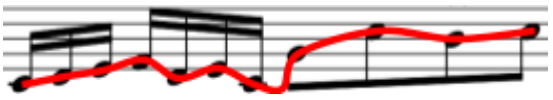
Because of the equal-but-opposite ρ settings between C and G, as soon as the contour meanders onto either of those notes a short-term cycle is set up whose stability is dependent on the Δ setting! A large Δ will 'break the cycle' fairly quickly but a small one will allow it to continue indefinitely. Mixtures of small and large note-specific Δ and several opposite-but-not-quite-equal ρ settings produce even more interesting chaotic effects.

Simple rules, and chaos *goes*,
Where it leads to none can know!

3) X/Y/Z Recursive Phrase Loopers

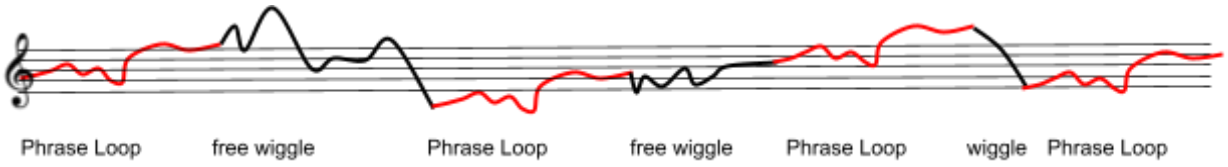
Human melody repeats itself, from cell to phrase to verse to movement, and usually with all levels operating at once. Much like a poem that rhymes, melodies exhibit contours of different lengths that recycle and permute throughout the piece. As with the Bach example above, these repetitions often also occur at different steps in the scale as they are recalled in motives or riffs.

To accommodate this Strange-R has 3 separate Loop[5] locations you can write to: **X**, **Y** and **Z**. Each is subtly distinct in its behaviour (see below), and whether they fire is determined stochastically. Let's first see how they interact with the main controls. Recall our Bach example…



…from which we abstracted the contour…

***Once a Loop is captured Strange-R applies its Random Walk algorithm to the Loop's pitch offset* so the underlying contour is shifted through the available notes on the quantizer. A 50% Loop probability and a medium +Δ might therefore produce something like this:**



a) X/Y/Z Loop Characteristics

Strange-R's Loops are not about making fixed EDM riffs, although they can easily do that. Instead, they're designed to introduce recursion by feeding the contour back into the scale and moving it around. Because there are 3 Loops you can also use them differently: you might set X to be a short melodic 'cell' that fires a lot, Y to be a longer phrase somewhat less frequently deployed, and Z as a much longer 'refrain' set to occur only occasionally.

The 3 Loops are independent in the notes they contain and the probability they will trigger (see below). However, they also have their own unique *characters*. **Loop X captures *everything***: notes, portamento and TrEG settings for each note, so Δ and ρ move the phrase through the quantizer but the articulations always repeat. **Loop Y captures *only the notes*** and ignores portamento and TrEG so while you move the loop through the scale you can also apply new articulations on the fly via the Portamento and TrEG settings. **Loop Z also only captures the notes *but applies* Δ *and* ρ *within the loop, not to the loop, so it acts like a variation system.***

[5] What Doug Hofstadter might call a Strange Loop

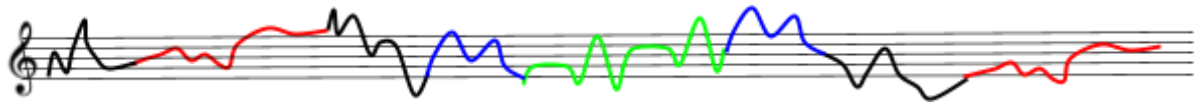
b) Loop Recursion & Recapture

The main point of the loop is to feed a melodic contour back into the quantizer, so you can walk the phrase through different scale degrees, apply different quantizer modes, or transpose it through the Circle of Fifths (see below). However, that doesn't have to be lost in the moment! Strange-R lets you build new 'loops of loops' from the results by capturing the output into one of the other loop locations. For example, you could set Strange-R running, capture that into Loop X, apply some Δ/ρ to walk it through the scale, turn it backwards, and recapture *that* into Loop Y!

c) X/Y/Z Loop Capture & Probability

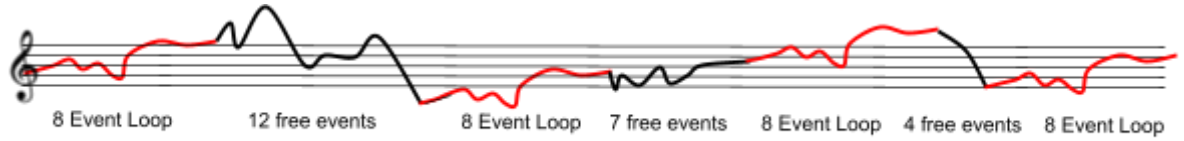
Strange captures a loop 'in performance', for as long as you hold the button combination (B+ either F#, G#, A# for Loop X, Y, Z respectively). Since each loop buffer is extremely large (over 5000 events!) they *can* be very long indeed but you will probably tend to only capture a few notes as shorter phrases are more easily recognised as they move through the quantizer scale.

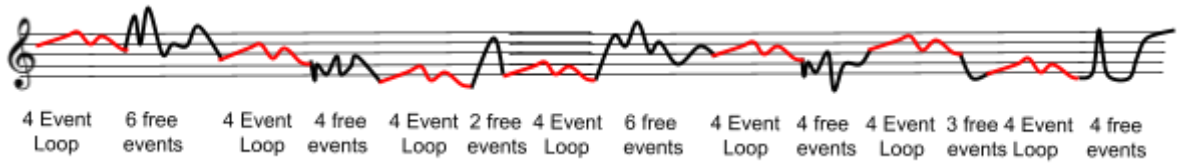
As soon as you end the capture by releasing the button combination the loop you captured will play with 100% probability. You can change the likelihood it plays (B+X/Y/Z respectively) from 0–100% and the probabilities combine between the loops so with all three at 100% there is a ⅓ chance any one of them will play. One set up might be to them them all at about 20% so you get a mixture of free running melody (black) and repeating riffs (red, green and blue):



i) Automatic Probability Balancing

Loop firing is determined on every normal (non-Loop) clock but because most Loops will be >1 event long they would have an 'unfair weighting' e.g. an 8 event loop with a 50% probability of firing would be 50% likely to fire *on every clock. Also, if it fires it will then play its entire 8 event length without stopping.* Thus, at any point in time you will be 8 times more likely to be *hearing the Loop playing* than not (i.e. 50% probability you would hear the Loop interspersed with only 1 or 2 non-loop notes before it fired again). A more intuitive sense of '50% probability' would mean you'd hear on average 8 *non*-loop events before our 8-Loop fires again, 4 non-loop events for a 4-Loop and so on—so this is exactly what Strange-R does.



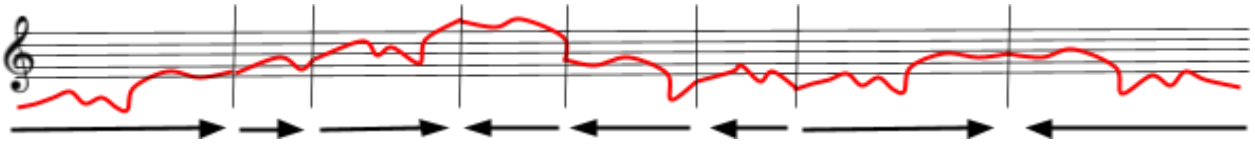


Loop probability is automatically modulated by Loop length: the longer the loop the *less* likely it is to fire but the *fairer* the share of *time* spent between it and free play (100% prob you still always play the Loop).

d) Loop Length & Direction Jockying

Once captured, a Loop can be truncated and/or reversed. Longholding B and turning X/Y/Z increases the loop length from 1 event at noon to the loop's full length (whatever that was) at 4 o'clock. This is a live performance control you can hear in real time, perfect for 'loop jockying'. Anticlockwise past noon the loop length increases in the *reverse* direction to its max *reversed* length at 8 o'clock.

The extreme clockwise/anticlockwise positions (5 and 7 o'clock) cause the length to differ randomly each time the loop plays, forward lengths only (clockwise) and forward *and* backwards (anticlockwise). The 'backwards random' position is ideal for creating an on-the-fly 'automatically random remix' of a repeating phrase:

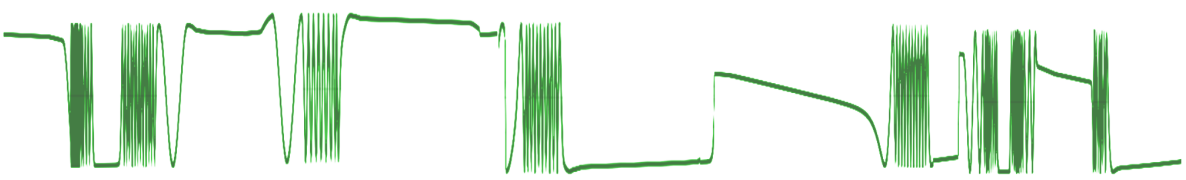


4) Strange χάos

Everything so far has operated on the short timeframe of note-to note transitions. Over longer timeframes music exhibits higher order regularities and transitions: songs have verses and chorus, symphonies alternate fast and slow movements and linear improvisations alternate between periods of relative status and activity. Strange-R archives this with a chaotic LFO subsystem (χάos or 'chaos' spelled using the Classical Greek alphabet[6]).

χάos is applied to the Strange Algorithm (and therefore also the Loop) to move its centre-point up and down. Because the LFO is *chaotic* the shifts will sometimes exhibit regularity, sometimes wild unpredictability, sometimes rapid fluctuations, sometimes stasis and constancy.

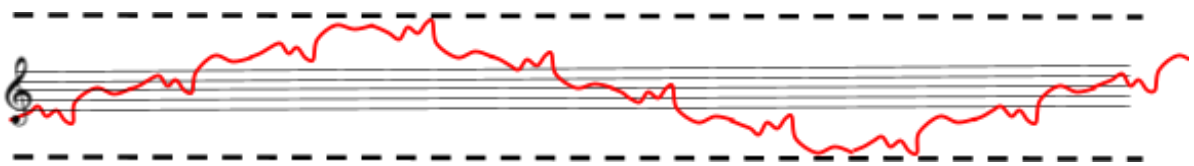
Χάos Speed sets the overall timescale of changes, **χάos Mod** controls how wild the fluctuations can be, and **χάos Depth** defines the extent to which they are applied, from nothing to 'a lot'. The result can be anything from sitting completely still (i.e. doing nothing) to a slow oscillation in centre-pitch to suddenly adding very complex contours, and transition speeds between these states can be very gentle or very rapid, as per the nature of a chaotic system!



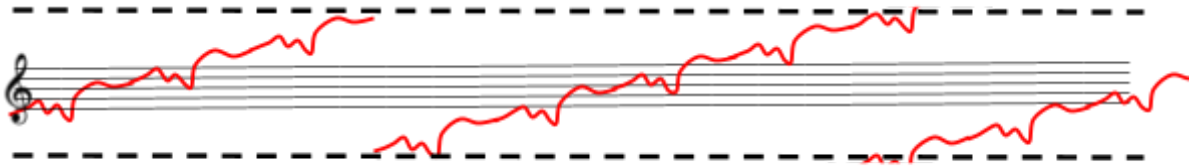
[6] Mostly: the Greek lower sigma 'ς' in χάος was harder to read, so for panel legibility we used an Arabic 's'.

5) Folder-Wrapper

Any max or min ρ setting would quickly send the algorithm straight to the +/- 10v limits so Strange-R incorporates a 'pitch window' system to avoid this. Its width defaults to maximum but can be dialled (D+X) right down to 0 to fix the pitch on one note. When Strange-R's output hits the top of the window, by default it reflects back down until it hits the bottom where it reflects back up again, and so on ('**fold mode**'):

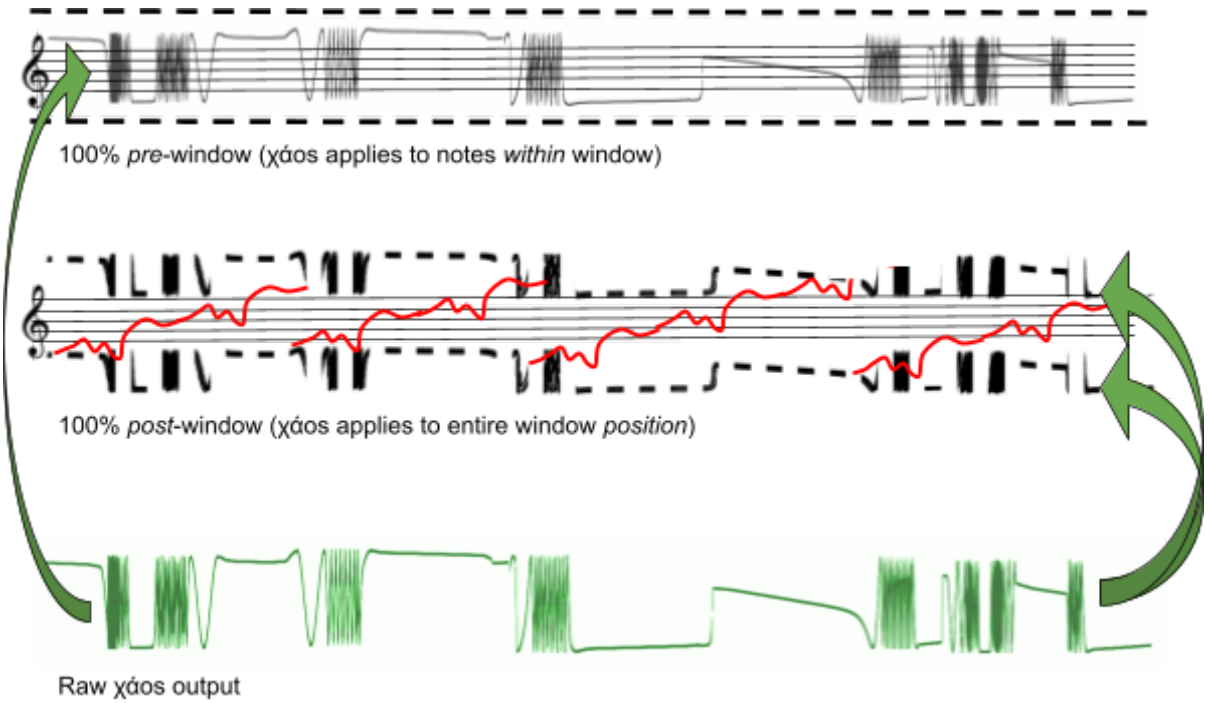


This behaviour can be changed to '**wrap mode**' where the output leaves on extremity of the window and re-enters at the other:



These are both very useful for setting up different arpeggiator behaviours: maximum ρ + fold gives you 'up-down', max ρ + wrap gives you 'up', min ρ + wrap gives you 'down'. The window type (fold vs wrap) is controlled with D+Y but is probabalistic: hard anticlockwise is 100% fold, hard clockwise is 100% wrap, anywhere in between morphs from one to the other being more likely each time the window is reached, offering further on-the-fly melodic recomposition.

The final control, D+Z, sets whether the χάος applies to the notes *within* the window or *to* the entire window position itself, again probabilistically:

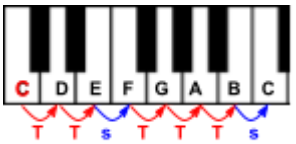


6) Theory-Aware Quantizer & Transformer

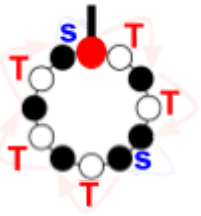
The next stage is the Quantizer. With no notes selected Strange-R's output is microtonal, resolved to single cent precision via its 16-bit output. However, as soon as notes are engaged on the quantizer the voltages are tuned to 12 Tone Equally Tempered values. Obviously you don't need to know the pitch class set theory of how the quantizer works under the hood but it's worth covering some basics to get the most out of it. **Just remember that you can load and transform scales instantly, automatically, live, in performance, and in key: Strange takes care of everything so it always sounds good and you can just enjoy the results!**

Scales vs Modes

The term 'scale' is often used interchangeably with 'mode'. While not exactly wrong, the distinction is worth unpacking. A **Scale** is *any ascending series of intervals which join an octave*. For most Western music the octave is divided into 12 semitones of which usually 7 form the scale. The pattern of tone/semitone steps up from the root defines its sound and the pattern repeats over each octave.

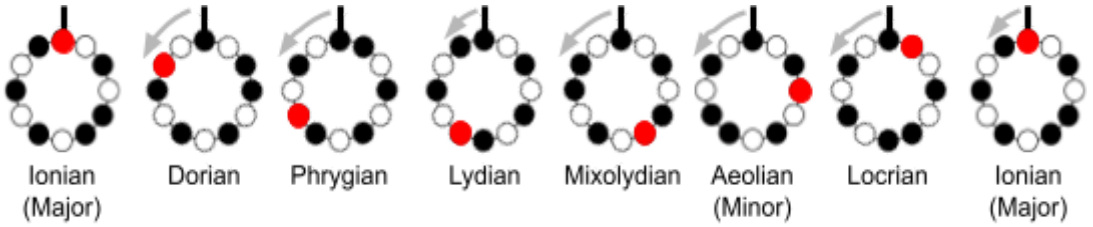


This superb technical resource by Ian Ring depicts this octave repetition circularly as a 'Bracelet Diagram': whatever the starting note, black beads show the notes *present* in the scale and white those that are *omitted* (NB *not* 'white notes' vs 'black notes'). Reading clockwise from noon (the starting note or *root*) gives the interval steps of the scale (usually **T**ones and **S**emitones). To the right you can see the familiar **TTSTTTS** pattern common to every major scale.



Rotations vs Reflections

A *Mode* of a Scale is represented as a *rotation*: it retains the overall *pattern* of spacings but shifts to the next available starting note. We can get the 7 familiar Church Modes, read clockwise from noon, by rotating the Major scale like this (red beads show the *original* root as it gets rotated):



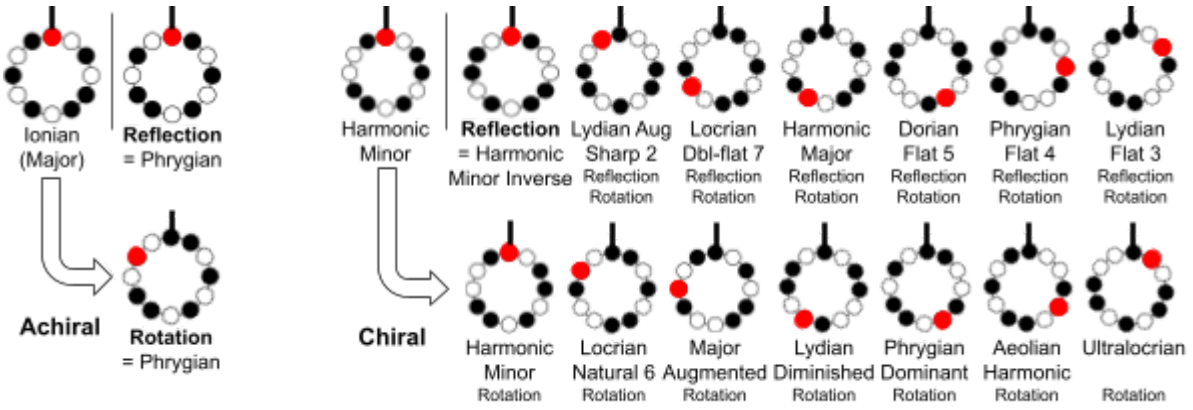
Some 'scales' are Modes of other Scales, and some are not. We obtained the Lydian by rotation from the Major (mathematically, Lydian and Ionian are said to be members of the same *set*). However, the Harmonic Minor scale for example *cannot* be obtained from the Major—there is no rotation of the Major that can produce it because they are drawn from fundamentally different *underlying* patterns or sets. In set theory terms the difference is this:

(Ionian is a member of the set of Lydian)

(Ionian is not a member of the set of Harmonic Minor)

Scales can also be transformed by **reflection:** simply flipping the bracelet on its vertical axis. This yields a further interesting observation. Reflecting some Scales just reproduces their ordinary unreflected Modes whereas reflecting others produces a whole *new* set of Modes. These are completely distinct from their ordinary unreflected counterparts. Scales which do this are termed **Chiral**[7] (meaning 'handed'): those that don't (like the familiar Major), **Achiral**.

The Major (Ionian) Scale is achiral: its reflections are identical to its rotations. The Harmonic Minor scale is chiral: its reflections are completely distinct from its rotations. This means that where the Major Scale pattern only exists in 7 modes, the Harmonic Minor pattern exists in 14: 7 by rotation and another 7 by reflection-then-rotation:



Every Mode is unique in character: think of them as unique colours. Just as we learned the visual colours by names as children we can also learn to recognise Modes by their overall sound. Simply hold the root note as a drone and improvise in the Mode above it: how the unique collection of notes functions in melodic relation to each other and the drone *is* that Mode (as well as the chords we form from the scale notes by stacking alternating scale tones).

Any combination of quantizer notes can be transformed, and you can program and alter the scale at any point live in performance to rotate/reflect *that*. Rotating arpeggios (i.e. 'scales missing some notes') obtains some chord changes from the Roman Number functions (I,ii,iii,IV,V,vi,vii^{o}) as well as from more distant Schoenberg Regions or Neo-Riemannian Transforms.

The preset scales in Strange-R have been carefully selected to offer the widest possible range of modal expression. Rather than spend 7 of our 36 available memory spaces (3 banks of 12 = 36) on the Church Modes alone, we have programmed each slot with a unique **Prime Form Scale**[8] from which all its Modes (and if Chiral, its reflections) are available via Strange's **Transform** function. This provides 343 different scales and arpeggios not including their 12 transpositions. The complete list is detailed in Appendix 1 but the most familiar are obtained as follows:

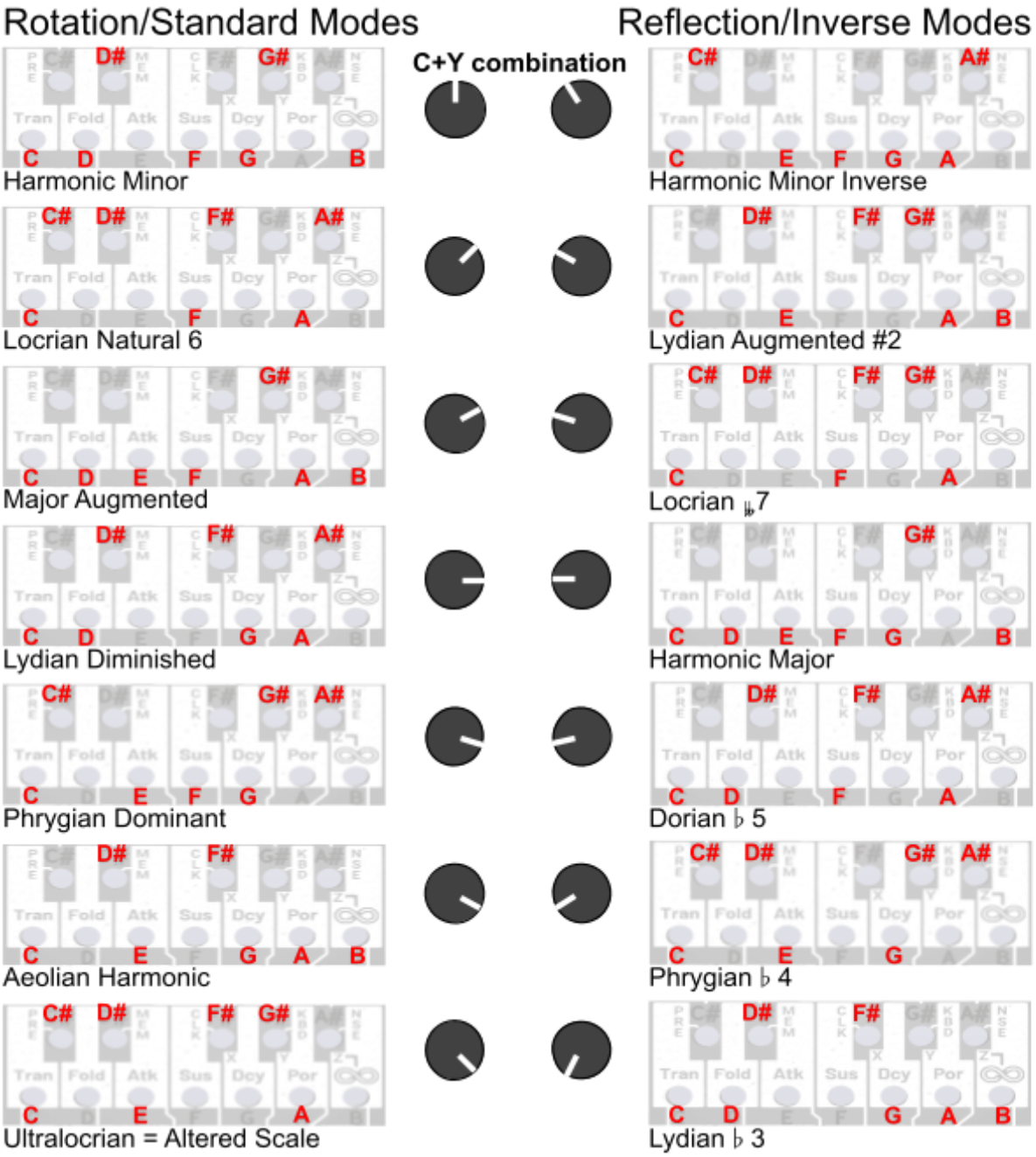
C#+X→C: The Major 7th arpeggio (+ Lothic, Phratic, Raga Lavangi modes)
C#+X→D: The Minor 7th arpeggio (+ Maj 6th, Lahuzu 4/4 and Karen 4/3 modes)
C#+X→E: The Pentatonic scale (+ Maj Pent, Sus Pent, Scottish Pent and Blues Minor modes)
C#+X→F: The Major/Ionian (+ Dorian, Phyrgian, Lydian, Mixolydian, **Aeolian/Minor**, Locrian)
C#+X→G: The Harmonic Minor scale (+ the 13 other modes and reflections listed above)

[7] Gloves are chiral (a left glove can't fit a right hand), socks are achiral (they'll fit either foot).
[8] Scales have a **Prime Form** (the most mathematically 'compact' rotation) and in strict terms we consider all other rotations Modes and Reflections (technically 'Enantiomorphs') of *it*.

Example: Harmonic Minor Rotations & Reflections

Strange-R handles the transformations automatically while keeping the root note constant, and only applies the transformation when you release the note button so you can apply a modal shift directly in performance[9]—an instant shift of musical colour or perspective.

Noon is always the 'untransformed' position, i.e. however the quantizer was before you started transforming it. If the scale is chiral the reflection modes will be identical to the rotation modes.



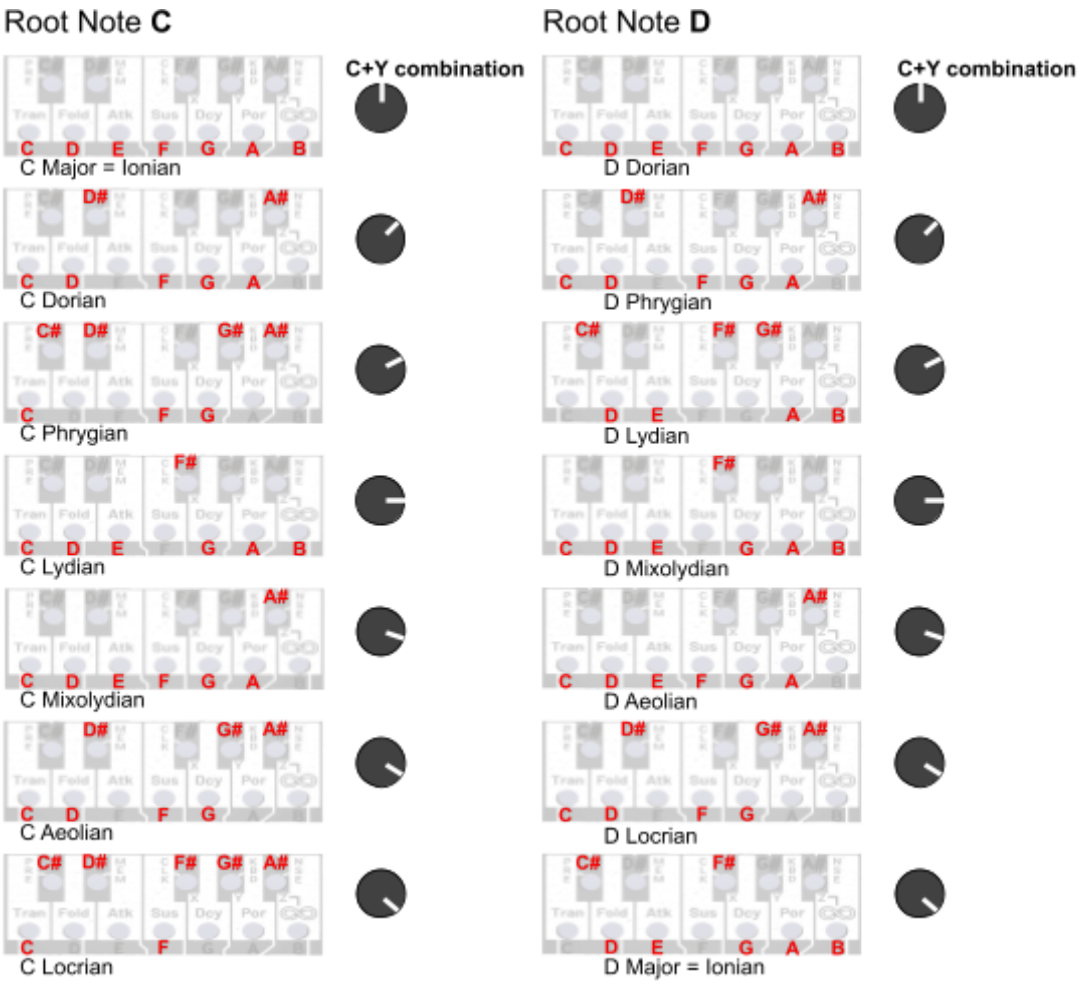
With 343 available modes you're not going to remember every pattern by name(!) so we suggest exploring to find some you like and then storing that as a memory location (see below)

[9] We're aware the enharmonic spellings are not correctly displayed for every scale e.g. A# vs B ♭ vs C𝄫 etc. The available panel space alas precludes their display so we plead dispensation from the Theory Police.

Transposition vs Root Note vs Rotational Centre

Modal transformations retain the root note of the scale, and this defaults to C. Strange can easily block transpose the quantizer to shift all the engaged notes up or down: for example, to obtain the scale of D Major/Ionian you would load *C* Major by turning C#+X to select the major scale preset (lights F) and then transpose this up two semitones to D by turning C+X two places.

However, there's another sense of 'root note' to understand. Lighting all the 'white' notes gives the *Major Scale (Ionian Mode) of C* and transforming that gives all its modes centred on that C. However, that *exact same collection of white notes is also the Dorian Mode of D*—and the Phrygian of E, Lydian of F, G Mixolydian, A Aeolian and B Locrian, each in relation to *its* root note. That is, the modes *of* C are the patterns of intervals applied *up from C,* which is why C is constant in each transform. If we wanted *D Dorian and its family* of modal transformations we would need to set the same white notes *but also set the root note to D so D stayed constant for each transform.*

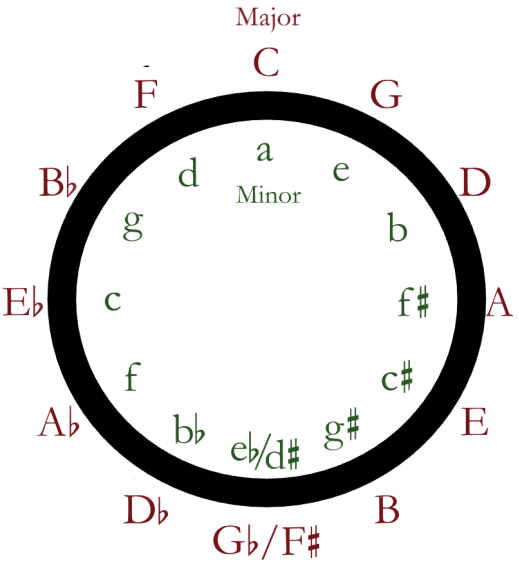


Setting the root note in performance is easy: just longhold the note for 2 seconds until the quantizer blinks. Most interestingly, because the root is also the mathematical 'centre of rotation', engaged quantizer notes can be rotated around *non-engaged* (unlit!) notes! Just set a *dis*engaged note as the 'root' and rotate—the results are well worth exploring…

Circle of Fifths Transformations

In music theory terms Modal Rotations are thought of as *parallel* operations in that they create different versions of the same key centre (*C* Major, *C* Dorian, *C* Phrygian etc.). However, almost all western music thinks in terms of *relational* operations: shifts of key centre to different but related keys. Another way to think about this is that 'transformational harmony retains key but changes character' where 'relational harmony retains character but changes key'.

The chords of a key are drawn entirely from the notes of its scale so are all essentially contained in the quantizer scale but because *modulation*[10] to a new related key centre (and thus scale) is such a common operation in music that we have provided a mechanism to achieve this also.

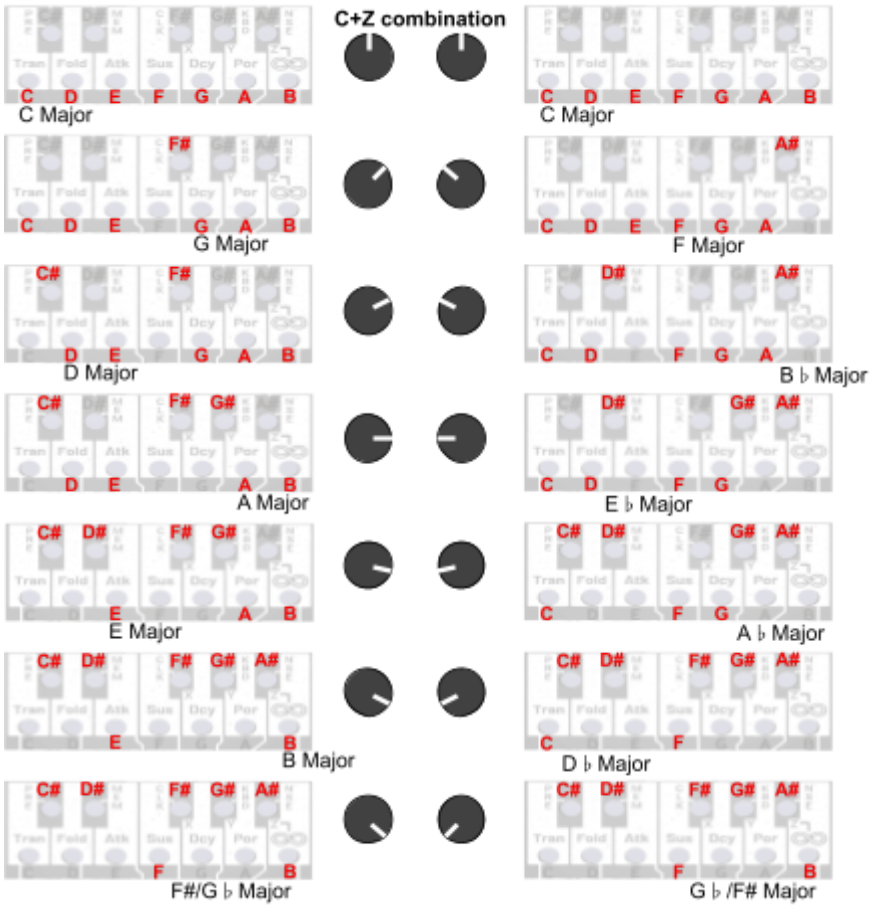


You may be familiar with the Circle of Fifths (borrowed in the DJ world by Mixed in Key as the Camelot Wheel for Harmonic Mixing). This represents the similarity of scales by their proximity on the circle. For example, we have already met the idea that C Ionian/Major shares *all* its notes with A Aeolian/Minor: the Circle of Fifths represents that with the inner 'minor' ring (C Major and A minor occupying the same position). Major scales a Perfect Fifth apart share all *but one* of their scale notes (e.g. C and G Majors share all except F/F#, C and F Majors share all except B/B ♭). They are thus neighbours on the Circle of Fifths and are said to be closely *related.* In simplest terms, this means transitioning between them *just sounds good!*

NB The Circle of Fifths works as originally intended (in Nikolay Diletsky's 1677 Grammatika) for 'standard' major/minor keys containing a perfect fifth but there's no reason why you can't apply the modulation to every kind of scale/arpeggio. Again, this is territory for you to explore!

Strange-R lets you navigate the Circle of Fifths just like Modal Transformations, but this time by rotating C+Z. Clockwise turns takes you clockwise on the wheel, anticlockwise takes you anticlockwise, and you will notice that between each step only one note will change:

[10] In Music Theory, 'modulation' refers to the process of changing key through the Circle of Fifths. In Synthesis 'modulation' means controlling something (more strictly, imposing the characteristics of one waveform upon another). As this is a synthesiser manual, to avoid confusion we use the 'synthesis' sense of the word unless otherwise stated.



Related Circle of Fifth keys are also modes of the original 'home' key (e.g. G Major/Ionian has the same notes as C Lydian). The difference is their *root:* for G Major it's G and C Lydian it's C. This relates very closely to Modal Brightness theory from jazz:

Diatonic Major Key	Equivalent Mode of C	Modal Brightness
G Major	C Lydian	7
C Major >>>>>>>>	**>>>>>>>> C Ionian**	**6**
F Major	C Mixolydian	5
B ♭ Major	C Dorian	4 = Neutral
E ♭ Major	C Aeolian	3
A ♭ Major	C Phyrigian	2
D ♭ Major	C Locrian	1
(D ♭ Major ♭ 3)	(C Altered)	(0)

Again and as always, no understanding of this is necessary: Strange-R does the theoretical thinking for you! Just twist, listen, explore and enjoy!

7) Portamento

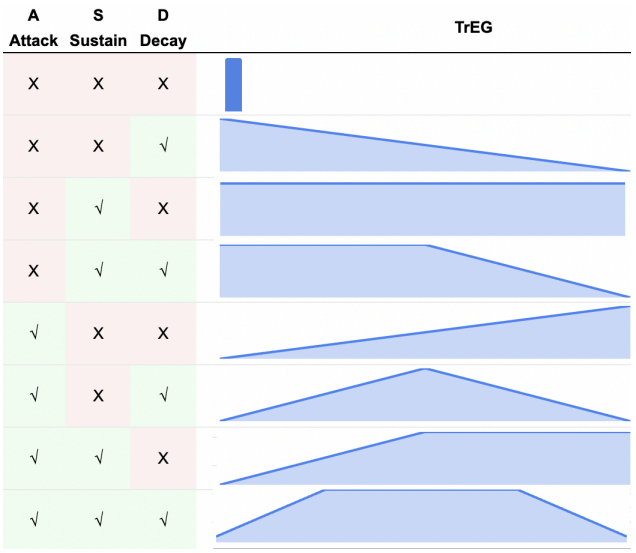
Portamento is applied stochastically, post-quantizer, and defaults to 0% chance: as you introduce probability, notes become more likely to glide between quantizer steps. You can also change the slope of the glide from exponential, through linear, to logarithmic, or a random selection of the three. This applies performative inflections to the melodic material Strange-R produces but sits 'outside' the main melody *generating* system.

Portamento settings are stored in Loop X but applied on the fly to Loops Y and Z (see below).

8) TrEGs

We believe Eurorack sequencing in general has missed an important idea best captured by a line of Stockhausen: "C played loud and C played soft are *two different notes*"—this is really the same idea from above about how our ear latches onto auditory regularities to form musical *objects*. The combination of pitch *and* articulation means we *hear* the 'loud C' and 'soft C' as two distinct musical *things* within the overall melody. Put differently, we think 'notes' are defined by more than their pitch. Just like SIG therefore, Strange-R does not output simple triggers: it produces envelopes to control a VCF, VCA or LPG directly.

Strange-R manages the Attack, Sustain and Decay articulation of each note by placing them under stochastic control[11]. As with Portamento you can also control the slope of Attack and Decay, and the overall Level of the combined envelope, and can also set these randomly for each event. Attack, Sustain, and Decay combine multiply and independently to provide 8 complex articulations if all 3 controls are dialled in:



There is also a further setting whereby the TrEG will only fire if the quantizer note changes. This avoids too many repeated notes with small Δ settings and introduces rhythm.

TrEG settings are stored in Loop X but applied on the fly, per note, to Loops Y and Z (see below).

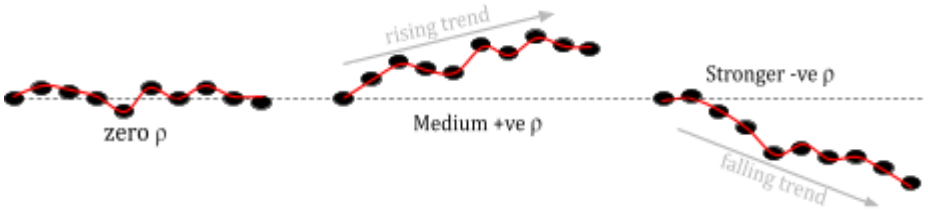
[11] Those familiar with SIG will note the omission of the Ratchet function. This is because there is no explicit duration control in Strange-R and thus Ratcheting becomes superfluous.

III. Operation Guide

Primary Functions

Correlation: ρ (Rho) — Direction

Correlation sets the overall rising/falling melodic 'trend' over multiple events: how strongly the next note will be in the direction of the previous few. +ρ steers the *overall* melodic trend up, -ρ down, and 0ρ static, while still permitting *individual* notes to rise and fall.

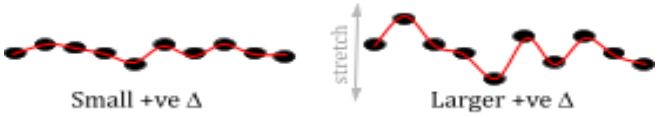


The ρ CV input allows external control of Correlation, and when the module detects an input the ρ **panel control acts as a CV attenuverter.** That is, +ve voltages will cause +ρ effects (and -ve, -ρ) when the panel control is +ve, and no effect when set 0 . *Negative* panel settings reverse the effect of the CV.

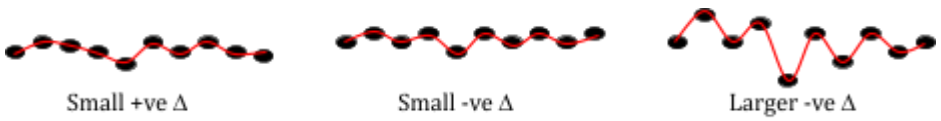
Deviation: Δ (Delta) — Jumpyness

Deviation sets the note-to-note step range: how smooth or jumpy the melody will be**.**

+Δ permits jumps in *either* direction: small +Δ produces relatively smooth melodies while larger +Δ relatively jumpier ones, à la Random Walk:



-Δ *always changes step direction:* small values produce effects like trills or melodic filigree while large -Δ produces bouncing ro alternating 'compound' melody (Random Hop!):



Small ρ values can be 'defeated' by large Δ values and large ρ values can 'defeat' small Δ values.

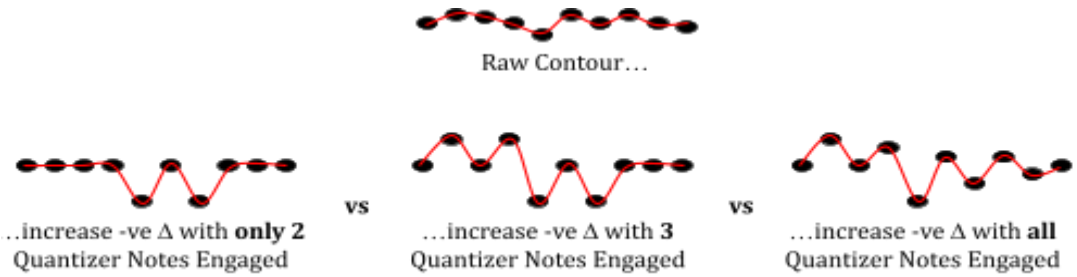
The Δ CV input allows external control of Deviation and as with ρ above, when the module detects a Δ CV input the Δ panel control acts as its CV attenuverter.

Quantizer

With no Quantizer notes engaged Strange behaves microtonally[12] and can be used as a unique and very effective unquantized stepped random source (and with 100% Portamento an unquantized *smooth* random source).

As soon as any key is pressed only that note[13] can then play, but can jump in octaves depending on ρ/Δ settings. Engaged notes light solid, and the currently playing note shines brightest (and when unquantized, the note nearest the current microtone lights).

Engaging or disengaging quantizer notes will interact with ρ and Δ so the Quantizer is an important part of *playing the instrument live*: it reveals or hides smaller melodic ripples. For example, while small Δ values can jump semitone gaps in a *scale*, if you remove notes to make an *arpeggio* your Δ value might no longer make the jumps so will stick on one note:



Quantizer keys access several additional functions when held in combination with other controls. This is covered below but in basic operation they are just on/off note selectors.

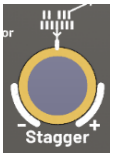
Rate/Freq

Internally clocked, Rate varies from 1bpm to around 180bpm (1 event every 15 secs to 12 in a second) at around noon, and then rapidly increases from there into audio rate for gestural tempo effects. Strange always responds to External Clocks via its Gate In jack *in addition to its internal clock* but you can **switch off the internal clock (hold F#, turn X fully clockwise) to sync Strange to the rest of your patch** (see below). Note too that in VCO Mode (see below) Rate acts as fine tune.

Rate responds to its CV input on the V/Oct standard allowing Strange to operate like a VCO when the clock is high enough (see below) but also to permit simple external tempo control where +1v doubles the clock and -1v halves it.

[12] Strange's 16 bit outputs mean its resolution is very high indeed.
[13] Technically, that *Pitch Class*: any C, any C#, any D etc., irrespective of octave.

Stagger



Stagger causes the tempo to change as the melody rises or falls. Positive Stagger means higher notes are faster/lower, slower, negative stagger means lower notes are faster, higher, faster. At small values the effect is subtle and 'off clock' (like classical tempo rubato) and at higher values it's rhythmical (x2 /2 etc.). Stagger is very useful for more experimental music and/or when using Strange as a random source, but is less appropriate for music requiring strict regularity like Techno so in those contexts ensure you disengage it with the centre-detented '0' noon point.

As with ρ/Δ, Stagger's panel control attenuverts its CV input when an input is detected.

Centre

Centre sets the overall centre pitch of the algorithm so can be used to 'brute force' the melody up or down, into a particular range and/or overlay a manual or CV controlled (see below) shape onto it.

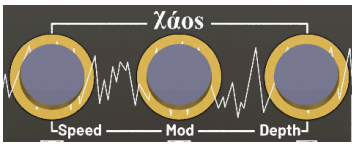
As with ρ/Δ, Centre's panel control attenuverts its CV input when an input is detected.

NOTE: min or max Centre can overwhelm ρ/Δ: this isn't a bug, it's just a natural effect of the scaling. If you notice happening just pull back on the amount of Centre you have dialled in.

Χάος

The Χάος section of Strange can be thought of as a separate sub-module: its two outputs are always available from the Χάος 1 and Χάος 2 CV outputs but its behaviour is completely separate to the main Strange Algorithm and will only affect the Strange's main CV Output if the Χάος Depth control is above 0 (or if you self-patch from one of the Χάος CV Outs to a different CV in). The underlying chaotic equation we've invented is a little bit secret but as far as has been possible its behaviour has been designed to provide unpredictable transitions between relative quiescence and freneticism so that 'higher level' or more 'macro' structural time scales can be applied to Stange's output behaviour.

Χάος Speed



Controls (broadly!) the overall speed of the Χάος function. By its nature it is unpredictable and fluctuates considerably and rapidly in frequency over short time periods. Longer vs shorter settings must therefore be considered as akin to the Pirates' Code: more a guideline than a rule.

As with ρ/Δ, Χάος Speed's panel control attenuverts its CV input when an input is detected.

Χάος Mod

Controls the amount of internal feedback within our Χάος function. This in turn translates to how erratic vs stable the function is likely to be but, again, this is Χάος so those are relative terms. However, in general terms, slow speed/low mod will produce more gentle undulations, fast speed/high mod more erratic spiky behaviour.

As with ρ/Δ, Χάος Mod's panel control attenuverts its CV input when an input is detected.

Χάος Depth

This controls the amount of Χάος applied to Strange's melody. With 0 Χάος depth, no Χάος applies: Strange's output is generated purely from the main algorithm determined by the panel controls (and/or their respective CV inputs): stochastic melody. Conversely, with 0ρ and 0Δ but some Χάος depth, Strange's output will produce purely chaotic melody. And, obviously, with a combination of both, the output will be stochaotic melody.

Χάος defaults to applying to the algorithm output itself, but can be set to probabilistically apply to the pitch window within which it occurs (see below).

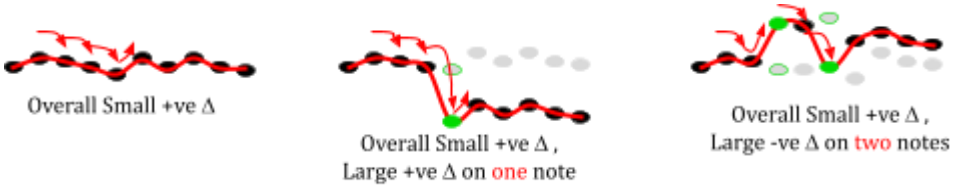
As with ρ/Δ, Χάος Depth's panel control attenuverts its CV input when an input is detected.

If Χάος suddenly becomes too much, you can tame it by either turning down Depth or pulling Speed/Mod down to 0. Alternatively, if you Χάος is being a little too dormant, you can wake it up by a quick flick of Speed or Mod.

Per-Note Functions: ρ, Δ and Stagger

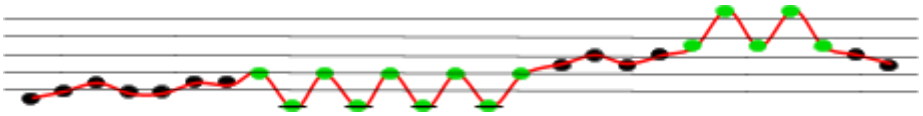
ρ, Δ and Stagger can also be applied on a per-note basis by holding the note while setting the respective control. In the case of Stagger, that specific note will simply play faster or slower (to cancel, hold the note and set Stagger to noon).

In the case of ρ and Δ, much more complex. Consider a large Δ on one note given an otherwise small masterΔ :



This behaviour is even more striking when ρ is applied to single notes, particularly where we combine *opposing directions* (+/- ρ) with small amounts of Δ . Given the following settings…

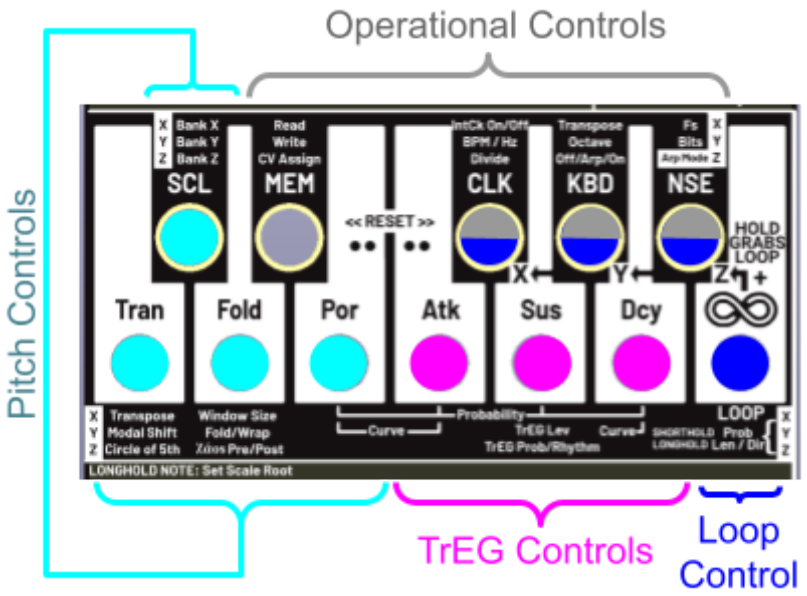
- +ρ of about a rising Perfect 5th on **C**
- -ρ of about a falling Perfect 5th on **G**
- +Δ of about a whole tone
- C scale on the quantizer



…the equal-but-opposite ρ between C and G means as soon as the contour meanders onto either note a short-term cycle is set up whose stability depends on Δ! A large Δ 'breaks the cycle' but a small one will allow it to continue indefinitely. Mixtures of small and large note-specific Δ and several opposite-but-not-quite-equal ρ settings produce even more interesting chaotic effects.

Secondary Functions

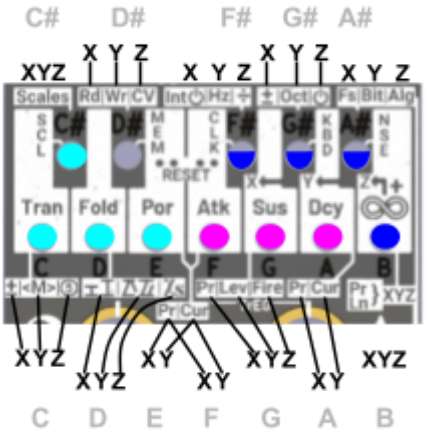
Secondary functions in Strange are all accessed through the combination of one of the Quantizer buttons and a panel control. They fall into two broad families, Operational Controls and Output Controls, with the latter dividing into settings for Pitch, TrEG and Loop.



Most have three parameters accessed by the Χάος panel control knobs doubling as secondary '**X**', '**Y**', and '**Z**' whenever one of the Quantizer buttons is held. Panel graphics helpfully show what X, Y, and Z do in each case. For example, 'Tran' controls Quantizer Scale Transformations so holding note 'C' and turning 'X' (Χάος Speed) *Transposes* the quantizer scale. Holding 'C' and turning 'Y' (Χάος Mod) *Modally Shifts* the scale, and 'C' + 'Z' moves it through the Circle of 5ths.

Loop adds further depth because it has to control Strange's *three* loops (see below). Some controls (ρ, Δ and Stagger) can also be applied to specific notes so in those cases holding Quantizer applies them to that note (see above). Finally, the Quantizer lets you apply a root note to a scale by longholding for 2 seconds (see below) so it's important not to wait *too* long between pressing the note and turning X/Y/Z so Strange knows to apply a second function rather than set a root.

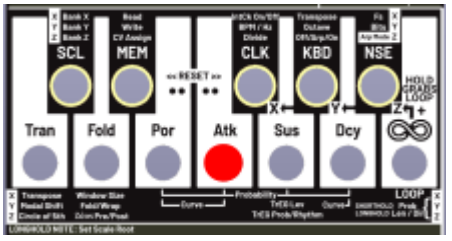
SmallHat follows the identical designations although with inevitably more compacted labels. Again, in *learning the instrument* the additional functions towards which you find yourself gravitating will become second nature and no longer require the mnemonics—they're just there to start you off if you need them.



TrEG Controls

As with SIG, every event is accompanied with an independent 'TrEG' (Trigger/Envelope Generator) output. This is SI's innovative replacement for the more usual (and much less versatile) trig outputs usually found in Eurorack sequencers. TrEG are designed for maximal versatility: they will trigger 'digital trig' modules (with an on/off response), 'analogue trig' modules (vactrol LPGs that convert a trig to an Decay envelope) and 'CV in' modules (VCAs/VCFs with the full range of A/S/D voltage control functions). The raw Triggers themselves are actually 125ms linear A/D envelopes so Strange-R can mimic LPG functionality with VCAs/VCFs (see Specifications)[14]. TrEG controls therefore only affect the TrEG output.

Strange defaults to firing raw triggers, so to access other kinds of TrEG you need to dial in some probability for Attack, Decay and/or Sustain. In addition, TrEG output jack is TRS having the TrEG on the tip and a 100% square wave on the ring intended for clocking out and in VCO mode.



Atk: TrEG Attack

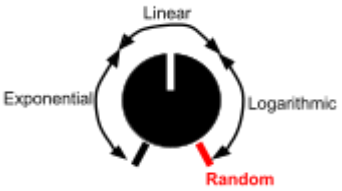
Attack controls the rising portion of the TrEG envelope and has two controls.

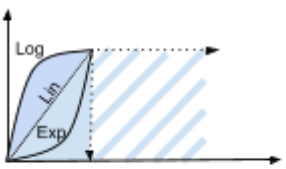
Atk Probability (hold F, turn X)

Holding note 'F' and turning Χάος Speed ('X') sets the *probability* (0–100%, default: 0%) that Strange will will apply a rising TrEG to each event. The rise time is defined by the clock and whether or not any other TrEG functions stochastically occur. With no other TrEG randomly applied (i.e. only Attack applies) the rise time is the length of the clock. If one other TrEG function applies (i.e. Sustain *or* Decay) the rise is half the clock. If *both* other TrEG functions apply (i.e. Sustain *and* Decay) the rise is ⅓ the clock.

Atk Curve (hold F, turn Y)

Holding note 'F' and turning Χάος Mod ('Y') sets the *type* of Attack curve Strange applies. Fully counterclockwise is exponential rise, centre is linear rise, 99% clockwise is logarithmic rise and fully clockwise is a special function where Strange randomly assigns the curve so there is a ⅓ change of it being either exp, lin or log *if* Attack fires (set by F+X).

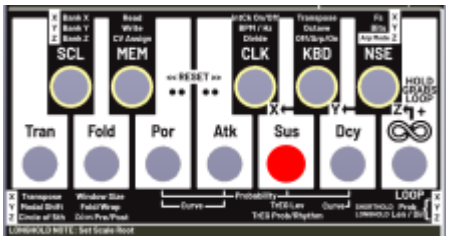




As a general rule, either no Attack (prob=0) or a Log curve will work best for more rhythmic melody at faster tempi because it has a harder edge. Lin an Exp gives more 'behind the beat' or 'reverse audio' effects but at much slower tempi can be very useful producing slow crescendi and filter sweeps.

[14] If you need shorter square triggers, just externally trigger-process the TrEG^ output.

Sus: TrEG Sustain



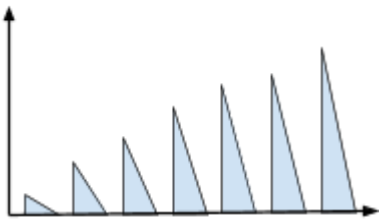
Sustain controls several aspects of the TrEG envelope: whether it holds, whether the TrEG fires at all, its overall level, and whether it fires on every clock or only when the notes changes.

Sus Probability (hold G, turn X)

Holding note 'G' and turning Xáos Speed ('X') sets the *probability* (0–100%, default: 0%) that Strange will will hold the TrEG high on each event. With no other TrEG randomly applied (i.e. only Sustain) the event will hold for the full length of the clock. If one other TrEG function applies (i.e. Attack *or* Sustain) the TrEG holds for them remaining half of the clock. If *both* other TrEG functions apply (i.e. Attack *and* Decay) the TrEG holds for the remaining ⅓ the clock.

Sus Level (hold G, turn Y)

Holding note 'G' and turning Xáos Mod ('Y') sets the *level* of the TrEG Strange applies, whatever it is (i.e. irrespective of Attack, Sustain and Decay probabilities). Fully counterclockwise is 0, through to 99% clockwise for maximum. The very last part of the turn to fully clockwise is a special function where Strange randomly sets the TrEG level from 0–100%.

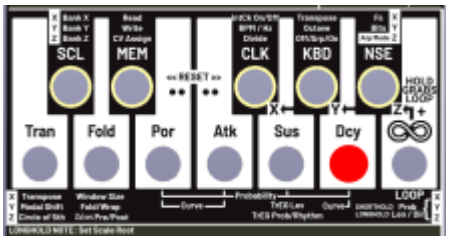


TrEG Probability & **Rhythm Mode** (hold G, turn Z)

Holding note 'G' and turning Xáos Depth ('Z') sets the *probability* (0–100%, default: 100%) of a TrEG firing *at all,* whatever it is (i.e. irrespective of Attack, Sustain and Decay probabilities). Fully counterclockwise is 0, through to 99% clockwise for 100%.

The very last part of the turn to fully clockwise is a special function where Strange only fires a TrEG when the Quantizer note changes. This simple idea can be *extremely* performative because it allows you to 'play the algorithm', controlling not only melodic trend and jump but *when notes fire* all from the panel: as such, it can be thought of as a Rhythm Mode.

Dcy: TrEG Decay



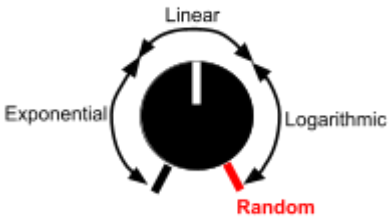
Decay controls the rising portion of the TrEG envelope and has two controls.

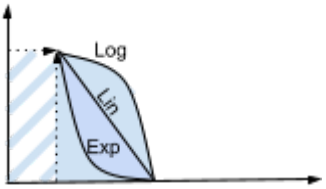
Dcy Probability (hold A, turn X)

Holding note 'A' and turning Χάος Speed ('X') sets the *probability* (0–100%, default: 0%) that Strange will will apply a falling TrEG to each event. The fall time is defined by the clock and whether or not any other TrEG functions stochastically occur. With no other TrEG randomly applied (i.e. only Decay applies) the fall time is the length of the clock. If one other TrEG function applies (i.e. Attack *or* Sustain) the fall is half the clock. If *both* other TrEG functions apply (i.e. Attack *and* Sustain) the fall is ⅓ the clock.

Dcy Curve (hold A, turn Y)

Holding note 'A' and turning Χάος Mod ('Y') sets the *type* of Decay curve Strange applies. Fully counterclockwise is exponential fall, centre is linear fall, 99% clockwise is logarithmic fall and fully clockwise is a special function where Strange randomly assigns the curve so there is a ⅓ change of it being either exp, lin or log *if* Decay fires (set by A+X).



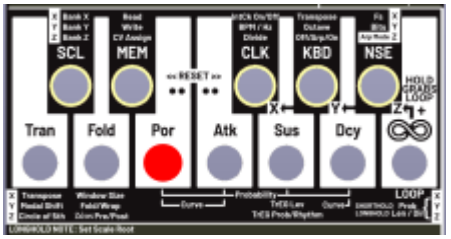


The combinations of the three TrEG functions are therefore as shown here (all set to linear):

A Attack	S Sustain	D Decay	TrEG
x	x	x	
x	x	√	
x	√	x	
x	√	√	
√	x	x	
√	x	√	
√	√	x	
√	√	√	

Pitch Controls

Strange has a range of general functions all broadly concerned with the CV output. These are more specifically divided between pitch-based controls (Portamento and Fold/Windowing), and Quantizer controls (Scale and Transformation functions).



Por: Portamento

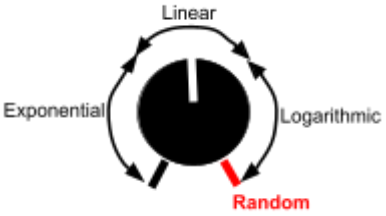
Portamento controls the pitch transition between notes and has two controls.

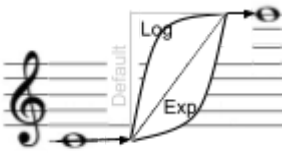
Por Probability (hold E, turn X)

Holding note 'E' and turning Χάος Speed ('X') sets the *probability* (0–100%, default: 0%) that Strange will will apply a portamento glide from the last pitch to the next. The glide time is defined by a combination of the clock and an absolute limit maximised for musical usefulness and the function is modelled on the classic Roland TB-303 slide.

Por Curve (hold E, turn Y)

Holding note 'E' and turning Χάος Mod ('Y') sets the *type* of Portamento curve Strange applies. Fully counterclockwise is exponential, centre is linear, 99% clockwise is logarithmic and fully clockwise is a special function where Strange randomly assigns the curve so there is a ⅓ change of it being either exp, lin or log *if* Portamento fires (set by A+X).





Tilt Controls

Strange also lets you 'tilt' the probabilities of the TrEGs and Portamento with respect to pitch, so that higher/lower notes will be more/less likely to apply Attack, Sustain, Decay and Portamento. In each case there needs to be >0 probability of the basic probability set for Tilt to do anything.

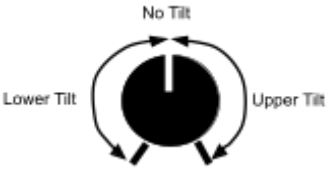
Tilt can set up rules that might say, for example, "All low notes must slow attack, all high notes must portamento".

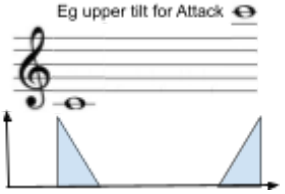
Attack Tilt (hold F, turn Centre)

Sustain Tilt (hold G, turn Centre)

Decay Tilt (hold A, turn Centre)

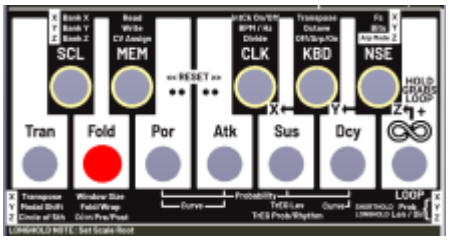
Portamento Tilt (hold E, turn Centre)





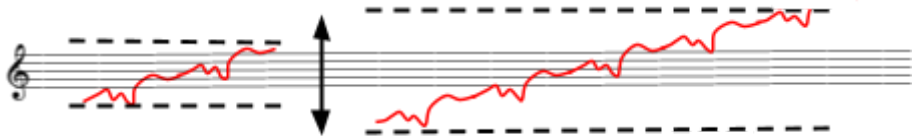
Fold: Fold & Windowing

Fold sets the range of pitches within which Strange's final output will operate and has 3 controls.



Window Size (hold D, turn X)

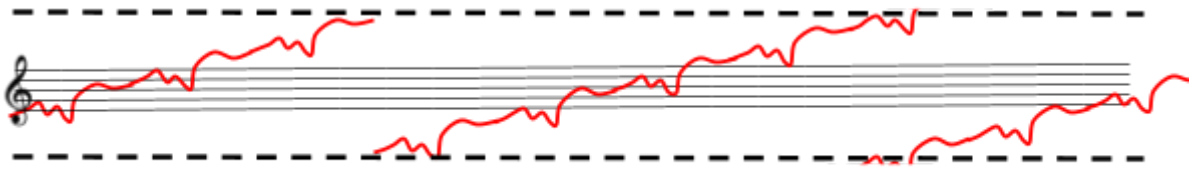
Holding note 'D' and turning Χάος Speed ('X') sets the width of Strange's output range and thus the range of pitches within which the contours sound. For use as a random source this is usually best left to its default of 100% so Strange can output voltages of +/-10v. However, most human melody sounds within a much tighter range, typically two octaves so +/- 2v. The Window size is therefore a very quick way to constraint Strange's output to you intended use. Note also that the window is applied *after* Strange's internal generative algorithm, which internally may still be generating large values but which are reigned in by the window before they leave the output. Note that Center conveniently shifts the window up and down.



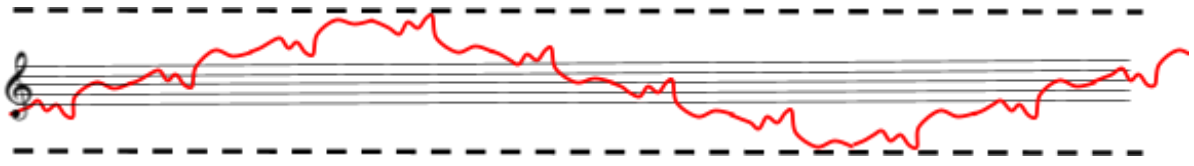
Fold vs Wrap Probability (hold D, turn Y)

Holding note 'D' and turning Χάος Mod ('Y') sets the behaviour Strange applies when its internal algorithm hits the limit of the window. This generates many useful musical effects and is why we apply the window *after* the generative algorithm, not directly *to* it.

The default setting is 100% **Wrap**, so as the internal algorithm continued to rise, when it hit the upper limit of the window it would wrap and come back up from the bottom. At speed, this would produce a sawtooth wave.

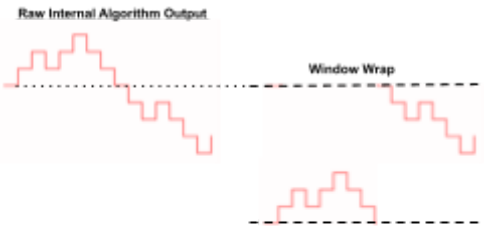


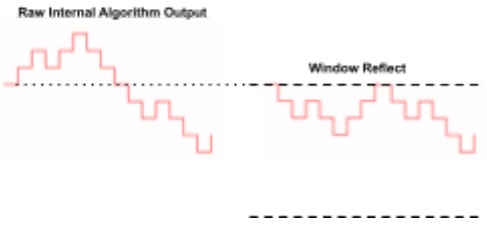
The other setting is **Fold**, so as the internal algorithm continued to rise, when it hit the upper limit of the window it would reflect and come back down from the top. At speed, this would produce a triangle wave. Note that because Fold is a reflection, the effects of ρ may appear to be reversed!



These are both very useful for setting up different arpeggiator behaviours: +ρ & Wrap gives you 'up', -ρ & Wrap gives you 'down' and +ρ & Fold gives you 'up-down'.

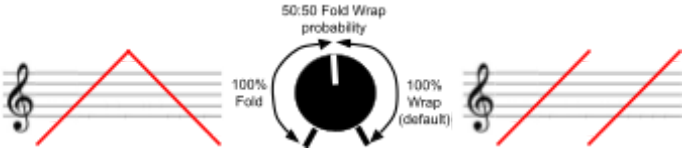
A more detailed view of the limit behaviour is shown to the right. The amount of time it takes to traverse the entire window width (from lowest to highest note) is fixed by the clock. That means *wider* windows jump more notes, and *narrower* windows step through their notes more slowly.





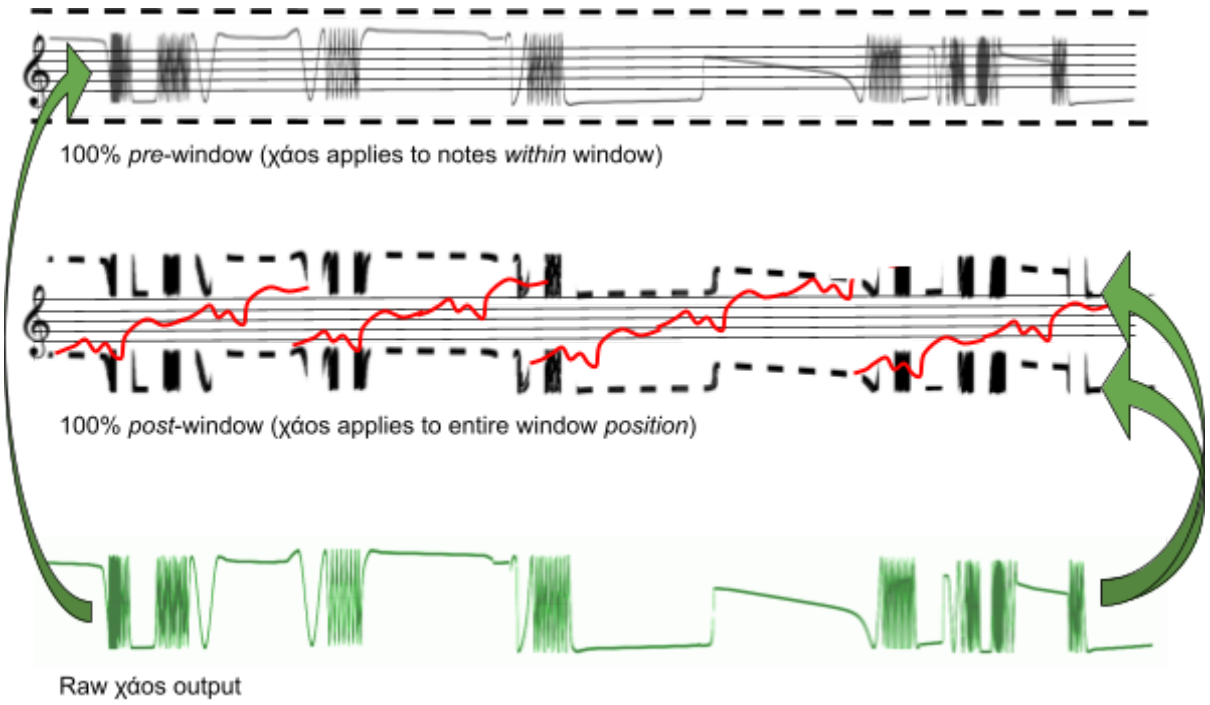
GOTCHA: in Wrap, at very low or high ranges where the output hits a 'floor' or 'ceiling', the output will *appear* to be random and/or moving in the wrong direction relative to ρ when the clock is fast. This is effectively an aliasing effect, because the output is jumping to the next value *outside the (truncated) range,* and then wrapping back within the *available* range. If you slow the clock down you will see that it is in fact wrapping correctly!

Strange also lets you set a probability for the likelihood that one or other behaviour will apply at each clock. At 50% any internal algorithm output exceeding the window limits would be equally likely to wrap to the bottom or fold from the top (and vice versa).

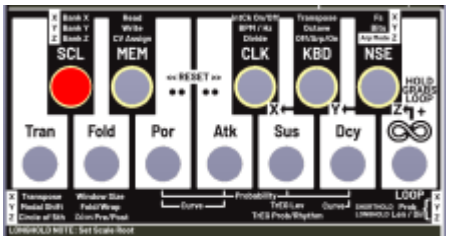


Χάος Pre/Post Contour (hold D, turn Z)

Holding note 'D' and turning Χάος Depth ('Z') sets the probability of Strange applying the Χάος Depth output to the algorithm *directly* (i.e. *before* the Window) or to the centre position of the Window *itself.* The default is 100% Pre-Window.



Scl: Scale Presets



Strange includes 36 *Prime Form* Scale Presets arranged in 3 banks of 12, denoted by the note that lights on selection. For technical information about Prime Forms please see Section 6 above (Music Theory Aware Quantizer). Basically, other quantizers/sequencers provide presets for each mode (Dorian, Phygian, Lydian, etc.). but these are really just members of the *same underlying family of scales.* Therefore, to provide you with the biggest range of musical material we have very carefully curated 36 Prime Form *families* and then allowed you to access their family *members* (their Modes) as Transformations (see next section). **This is a much more useful solution because it permits something previously very challenging to achieve in Eurorack: harmonic progression. The full list of scales is found in Appendix 1 but each is designated a note so you can remember your favourites.**

In each case, you can cancel the present load by turning the control fully counterclockwise until no note lights are lit. Strange will return to whatever was previously loaded into the Quantizer.

Scl Bank X—Arpeggios and Standard Scales (hold C#, turn X)

Holding note C# and turning Χάος Speed ('X') selects one of 12 'standard' scale/arpeggio families and includes the main arpeggios (Major 7th, Minor 7th, Dominant 7th, Sus 4), the Pentatonic, and the Diatonic Set (from which you can access the Major Scale, Natural Minor Scale and Church Modes, Ionian, Dorian, Phrygian, Lydian, Mixolydian, Aolian and Locrian, as Transformations (see below).

Scl Bank Y—Extended Scales (hold C#, turn Y)

Holding note C# and turning Χάος Mod ('Y') selects one of 12 rarer scale families, including the Double Harmonic, Whole Tone, Octatonic and Chromatic scales—and of course all of their modes as Transformations (see below).

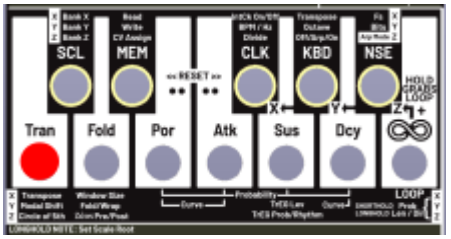
Scl Bank Z—Composers' Scales (hold C#, turn Z)

Holding note C# and turning Χάος Depth ('Z') selects one of 12 rarer composer scale families, scales invented by (or associated with) prominent composers. These include the Messiaen Modes of Limited Transposition, Verdi's Enigmatic, Debussy's Heptatonic and the Schoenberg Hexachord—and of course all of their modes as Transformations (see below).

NB There are also a few composer-related scales hidden in modes of other scales for you to explore: for example, Stravinsky's Petrushka scale is actually the 6th Inverse Mode of the Hungarian Major[15].

[15] It's useful to memorise facts like these to impress at dinner parties.

Trans: Quantizer Transformations



Strange's scale presets are not themselves modes but Prime Form scales, *families* of modes whose individual members are accessed by a simple control turn. There is also a simple 'blunt transpose' function which offsets the Quantizer by semitones, and a Circle of Fifths control to change to other keys. **This means Strange can uniquely achieve something not usually associated with Eurorack performance: instant changes in harmony. All transformations apply on *release* so you can time them in performance.**

The quantizer is also constantly aware of its current state so you can freely add or subtract notes on the fly. Strange always applies your new transformations to whatever notes are dialled in at the time. It makes no distinction between scales, arpeggios or any other collection of notes: it transforms *whatever* is dialled in even if we don't usually think of an arpeggio as having modes!

The important point is that all Strange's Transformations are extremely easy to achieve and related by music theory, so even if you didn't know how to achieve them without Strange's help, and don't know exactly what they are going to sound like ahead of time, there is a very high probability that the results will be musically interesting, so dive in and experiment!

For a graphical representation and deep explanation of the transformations see Section 6.

Transpose (hold C, turn X)

Holding note 'C' and turning Χάος Speed ('X') shifts the entire contents of the Quantizer up or down by up to 12 semitones to achieve a simple up/down transposition.

Modal Shift Rotation (hold C, turn Y)

Holding note 'C' and turning Χάος Mod ('Y') shifts the contents of the Quantizer to each possible mode of the selected notes, based on the current root note (see Section 6 above). The 'notes' selected are really an *interval series,* for example the Major Scale series of Tone-Tone-Semitone-Tone-Tone-Tone-Semitone. We form modes by shifting (or more correctly, *rotating*) these, removing the first interval (**T**TSTTTS > TSTTTS) and adding it on the end (TSTTTS**T**). Strange does this for you and can apply it to anything dialled into the Quantizer. Best of all, because all the modes share the same common root note the result is a set of *parallel scales of the same root.* That is, the root note will always be common ('on' in the Quantizer).

Some ('chiral') scales have the additional property that, as well as standard modes, they can also be *reflected* to provide an additional set of *inverse modes.* Strange archives this by representing the unrotated contents of the Quantizer at noon, positive (standard) modes as clockwise of noon, and inverse modes as counterclockwise of noon (and where the Quanstizer notes do not have an inverse you will just hear the standard modes again).

Circle of Fifths Rotation (hold C, turn Z)

Holding note 'C' and turning Χάος Depth ('Z') shifts all engaged notes in the Quantizer up or down by a Perfect Fifth. For the Major and Natural Minor Scales that means rotating clockwise or counterclockwise through the Circle of Fifths and thereby moving to the nearest related key centre. Noon is whatever the present setting of the Quantizer, clockwise adds #s and counterclockwise adds ♭ s to the key signature. If music theory isn't your bag, that's totally fine, Strange does the heavy lifting: all you have to do is twiddle and listen!

Due to the unique properties of the Diatonic Set, the Circle of Fifths only really makes sense to apply with these two scales (and their arpeggios) if the aim is a harmonic progression to the most 'related' keys. Obviously you are free to apply Circle of Fifth rotations to *anything* dialled into the Quantizer but the results will not necessarily maximise harmonic relation. They may very well still be musically interesting!

Setting Root Note (longhold note for 2sec).

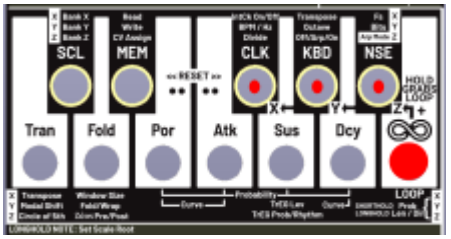
As above, modal rotations maintain the root note of the scale or arpeggio, defaulting to root note 'C'. To switch the root note/key centre to a different note just longhold the note you want as the new root. After 2 seconds the whole quantizer will flash and the new root is set. You will then note that all further modal rotations will be with respect to *that* root note and that will be the note that remains constantly on for all modal rotations.

Circle of Fifth shifts are not *themselves* affected by root note BUT shifting by Circle of Fifth *does* also shift the root note for Modal Rotation.

Combining Scale Preset, Transposition and Modal Shift

Presents and Transformations can be combined to 'dial up' any mode of any scale in any key with considerable ease. Suppose you wanted to play in F# Phrygian. You would first load the Diatonic Set (Scl: C#+X to present 'F') giving you C Ionian, then Transform to C Phrygian (Trans: C+Y to Phrygian), then Transpose up to F# (Trans: C+X to F#).

∞: Phrase Loop Controls



As detailed above, Strange's concept of Looping differs from typical sequencers in that it has *three* independent Phrase Loopers. Each one can hold different melodic material generated by the algorithm which can then be moved through the quantizer scale using the ρ/Δ controls you used to generate the material in the first place. Each loop has its own probability of playing, can be manipulated in length and playback direction, and has unique properties in terms of how they are affected by TrEG/Portamento and ρ/Δ.

Phrase Loop	Contour	TrEG/Portamento	ρ/Δ
X	Fixed at Capture	Fixed at Capture	Moves Contour
Y	Fixed at Capture	Alters Notes	Moves Contour
Z	Varied from Capture	Alters Notes	Moves & Alters Contour

While the 3 Loop lengths are extremely long by CV standards—well into the thousands of events—for the most part you're more likely to capture just a handful of events to make a recognisable motif.

Phrase Loop X Capture (hold B then hold F#)

Holding note B and then holding F# captures the algorithm output for the duration of the press and loads it into the Phrase Loop X location. As soon as the F# is released, Strange starts to play the newly captured loop X with 100% probability. The ρ/Δ controls change function to move the *phrase loop* around the quantizer scale rather than the raw algorithm output. What they do however remains the same: +ρ moves the phrase loop up, -ρ moves it down, +Δ jumps it around and -Δ jumps it with alternation. The loop plays its full length before moving on to the next cycle at the different pitch. TrEG and Portamento functions do *not* change the captured Loop X: it plays back exactly as captured with whatever TrEG and Portamento settings applied during capture.

Phrase Loop Y Capture (hold B then hold G#)

Holding note B and then G# captures output into Phrase Loop Y. As soon as G# is released loop Y plays with 100% probability and as above, ρ/Δ now move the *loop* not the algorithm. However, TrEG and Portamento functions *do* affect Loop Y: the pitch contour always plays the same but with whatever new TrEG and Portamento settings apply during each playback.

Phrase Loop Z Capture (hold B then hold A#)

Holding note B and A# captures Phrase Loop Z which plays with 100% probability when A# is released. However, as well as new TrEGs and Portamentos affecting each event, ρ/Δ now slightly change the *loop contents* as well as moving it up and down: theme and variation.

Loop from Loop Capture (hold B then hold A#)

Whatever Strange is doing with one loop in loop play can be recaptured into one of the other two loop locations on the fly with exactly the same procedure. This includes all the effects of new TrEGs/Portamento and ρ/Δ movements.

Loop Probability (hold B then turn X, Y or Z)

Each loop can contain different material of different lengths. As soon as each is captured it will start to play with 100% probability. This can be changed between 0-100% by holding note B and turning either Χάος Speed (for loop X probability), Χάος Mod (loop Y) or Χάος Depth (loop Z).

Strange balances loop playback against raw algorithm output to produce a melody-with-refrain effect at lower loop probabilities. Loops at 100% play constantly, and are balanced proportionately when more than one is at 100% (i.e. with all 3 at 100% each is heard approximately ⅓ the time). Strange also automatically balances loop *length* with probability so that longer loops do not dominate playback[16].

Overwriting Loops: a Conceptual Clarification

The way Strange thinks about loops is not exactly how we normally think of them in a standard sequencer (or even SI's SIG for that matter). Ordinarily a loop is an endlessly repeating short sequence of notes—but remember that the Strange Algorithm doesn't produce *notes*. It actually produces a melodic contour primitive, a very highly resolved (i.e. microtonal) arbitrary voltage curve with respect to time. The melody comes when you then put that curve through the quantizer to fit whichever notes are currently selected to it.

When you loop in Strange you are looping the *curve not the notes*. In addition, a crucial part of the design concept is then that, while playing the loop, the two controls you used to make that curve, ρ and Δ, then move that curve up and down through the quantizer notes, how human composers use melodic phrases.

At the point you capture the Phrase Loop we made the design choice that the joystick should immediately disable and set to 0 so you just hear the Phrase Loop as it was captured, without it moving. As soon as you touch the joystick again it comes back into operation so you can move the phrase around. However, at the point of capture Strange ignores the physical joystick position and sets the two controls, ρ and Δ, back to 0, as if you had a vertical stick.

Therefore, if you try to overwrite loop X on itself *without touching the joystick first* Strange thinks ρ and Δ are still at 0 (=vertical stick) so no pitch movement, which means you will get a constant stream of the same note! Recall too that, while Strange's loops are probabilistic, even if you dial your loop probability back to 0% to return to generative play, if you haven't touched the joystick Strange *still* thinks the stick is at 0 so you will still get that constant note stream.

Therefore, to overwrite a loop just just make sure you move the joystick first! You might also want to come back to generative play too because the joystick range is greater in that mode (another design choice: notes make bigger jumps than whole phrases do!) but it's not essential.

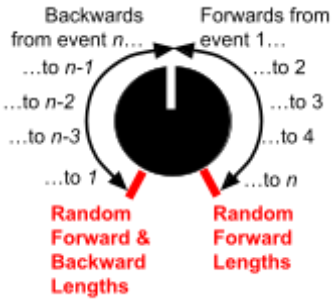
[16] If Loop X were a 4 notes, Y 16 notes and Z 32 notes, with all set to 100%, over the long run, X, Y, and Z would be playing for approximately the same amount of total time (i.e. X plays 4 times more often than Y, and Y twice as often as Z).

Loop Length & Direction (hold B until it flashes, then turn X, Y or Z)

Holding B until it flashes (1 sec) and then turning X, Y or Z sets the length and direction of the captured loop. There is a slight knack here: you need to hold long enough so Strange knows you're setting the *length* and not the *probability:* this is what the flashing shows. However, once it starts to flash, turn the X/Y/Z control you want within 1 sec so Strange knows you don't want to set B as the root note (*2* seconds, see above).

As above, Strange plays the entire captured loop of *n* events in the forwards direction as soon as the capture key is released (F# for X, G# for Y, A# for Z). The X/Y/Z length/direction control then sets the loop to play from the start to its n^{th} event between noon to 99% clockwise. The further the control is turned the more of the captured loop you hear, up to its full length. At full clockwise, Strange plays a random amount of the loop, from the start, in the forwards direction.

Between noon to 99% counterclockwise, the further the control is turned the more of the captured loop you hear, up to its full length but *backwards from its last event* (its retrograde). At full counterclockwise random fragments of the loop are played both forwards and backwards.

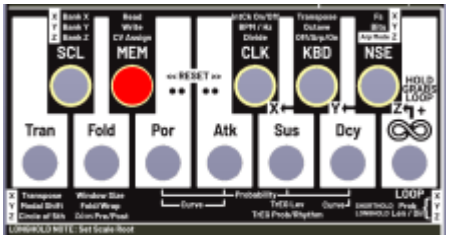


GOTCHA: Recall that Rhythm Mode (hold F, turn Z to 100%, above) only fires a TrEG when the note *changes.* If you have a Loop that you truncate back to only one note then of course the note never changes so Strange will appear to go silent and stop firing TrEGs.

Operational Controls

Strange has a range of operational settings which affect aspects of its overall function.

MEM: Memory Control Functions



Strange has 7 memory locations available for storing the entire state of the machine, including all controls, Quantizer notes, CV allocations and Loops. These can then be easily recalled after power cycling. In addition, on top of all Strange's primary functions having dedicated CV input, it also lets you assign its 3 assignable CV inputs to any 3 secondary functions.

Read Memory Location (hold D#, turn X)

Holding note D# and turning Χάος Speed ('X') recalls one of the 7 available memory slots designated by the white notes C, D, E, F, G, A, B and loads all settings from that location. If you accidentally engage this function you can turn fully anticlockwise until no notes are lit to cancel the load, and the module will continue with whatever it is doing at the time.

Write Memory Location (hold D#, turn Y)

Holding note D# and turning Χάος Mod ('Y') writes into one of the 7 available memory locations designated by the white notes C, D, E, F, G, A, B. If you accidentally engage this function you can turn fully anticlockwise until no notes are lit to cancel the save, and the module will continue with whatever it is doing at the time.

The entire state of the machine, including all controls, Quantizer notes, CV allocations and Loops is saved so this is a very useful way to recall the previous day's work and/or use scenes in a live performance.

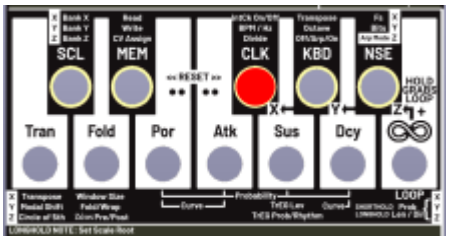
Assign CV (hold D#, turn Z)

Holding note D# and turning Χάος Depth ('Z') allocates the most recently accessed control to one of the 3 assignable CV inputs, X, Y, and Z. For example, if you wanted to put Modal Rotation under CV control, you would first apply the function (hold C, turn Y). You don't even need to actually commit the change—you could turn the control so whatever was in the Quantizer is preserved, Strange just needs to detect that you touched that secondary function. Without doing anything else, then just assign the CV to your chosen input, X, Y, or Z and any voltage sent to the corresponding input will then control the function.

ProTip: try self-patching one of Strange's Χάος outputs, or even mult its main output into an assignable CV!

GOTCHA: When primary functions are under CV control via their dedicated jacks (e.g. ρ, Δ, Centre etc.) the panel knob becomes an attenuverter for the CV in. For example, if you had ρ and Δ under external control but left the joystick at centre Strange would not respond because the CV ins would be being multiplied by 0 so you would need to have the JS at about half past one (i.e. ρ CV x1, Δ CV x1). This also means you can easily reverse or trim the CV effect.

CLK: Clock Control Functions

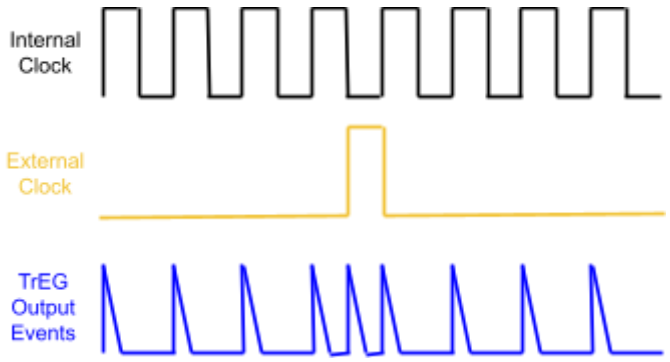


Strange defaults to its internal clock on startup but can be clocked externally to sync with the rest of your rack. It also offers some options relevant to its 'extended functionality' utilities.

Internal vs External Clock: Internal Clock On/Off (hold F#, turn X)

Holding note F# and turning Χάος Speed ('X') clockwise *switches the Internal Clock off.* Turning anticlockwise returns the default of Internal Clock on. This requires a little explanation.

Even in (default) Internal Clock operation Strange will respond to an External Clock input. If no ExtClk is present, or it's just unpatched, then the module simply acts as you would expect. However, if it detects an ExtClk this will also trigger Strange 'in parallel' with the IntClk ie an extra event will fire between the IntClks.



This means that 'External Clocking Mode' is really a case of *switching the Internal Clock* ***off***. In ExtClk mode Strange, like SIG, has no minimum so will wait however long it needs to receive the next clock. It can also be clocked well into the auditory range, up to low kHz. **The clock state is saved into memory when you switch it so will retain after power off.**

The reason Strange works this was is twofold: first, it permits the expiration of interesting and unusual rhythms by combining Int and Ext Clocks. Second, at high clock rates, it mimics the effect of oscillator sync (see below)!

NOTE: the duration of a TrEG is based on the *last* event so at very slow clock rates turning the rate up won't take immediate effect because the last (slow) TrEG will have to run its course.

Normal BPM vs VCO Hz Clock (hold F#, turn Y)

Holding note F# and turning Χάος Mod ('Y') selects normal BPM range (anticlockwise, default) or high VCO Hz range (clockwise) for the Internal Clock.

Internal BPM range varies from 4bpm (250mHz) to 2250Hz (135kbpm) defaulting to 120bpm on startup. The range is quite large and is more designed for arhythmic gestures rather than very precise two-decimal-point-correct bpm (although is of course very stable). If you need careful selection/synchronisation of bpm you would use the External Clocking.

Internal VCO Hz range is designed for two purposes: very rapid audio-rate randomness from the CV Out, and v/oct VCO control from its TrEG output. In the first case, Strange does everything that it normally does, just much faster, and this forms a kind of stochatoic noise source. The second case needs more explanation.

VCO Mode

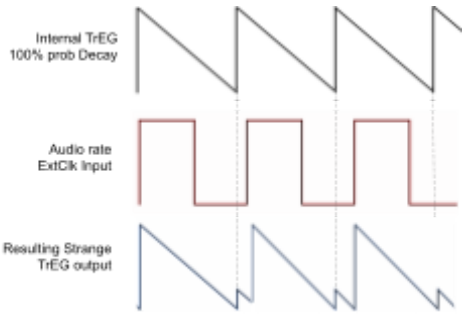
With the IntClk set to VCO/Hz mode, the TrEGs first so rapidly that they can act as a perfectly viable audio VCO: simply patch them into a filter or VCA and listen to the output. On switching to VCO mode, the default frequency is Concert A 440Hz (also allowing you to use Strange as a handy tuning source!).

Rate control now sets tuning, Centre sets Octave, and because Rate CV In is scaled to v/oct you can now 'play' Strange as a VCO from your favourite standard sequencer (as of course you can in standard mode except you would be thinking in terms of 1v causing a 'doubling in tempo' rather than a 'jump of an octave'—they are of course, exactly the same thing at different time spans).

The initial sound is Strange's default TrEG 'trig' (a thin saw wave) and because it's length is fixed there is an upper frequency limit on it. However, if you select 100% probability Decay then you will hear a 'full length' sawtooth which scales with frequency so the upper range will be much higher. If you combine this with 100% Attack then the two will combine to make a triangle wave. Less than 100% probabilities will make increasingly 'noisy' VCO effects.

100% Sustain on its own won't work because, being designed for legato playing in normal operation, it lacks rising edges. Sustain *will* however combine with Attack and/or Decay to provide trapezoid waves and of course also at low frequencies for triggerable probabilistic trapezoids. Genuine square waves are available via the ring of the TrEG TRS jack output.

Most interestingly, because Strange is always listening to its External Clock input, injecting audio rate waveforms into ExtClk in will produce VCO sync effects!

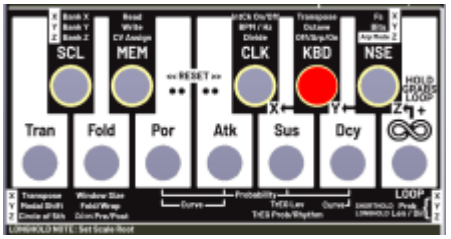


GOTCHA: remember that external clocks get through at any frequency which, if they are at ordinary clock bpm rates, will manifest as clicks on the TrEG (=VCO) output.

Clock Divide Mode (hold F#, turn Z)

Holding note F# and turning Χάος Depth ('Z') divides the clock, be it IntClk or Ext Clk by integers up to 16. The Quantizer will indicate the division by which light is shown: the default is /1 (i.e. no division) with note B lit. Divide by two (half time) is A# two notes down, /3 is A (three notes down) and so on, to /12 (C), and then /13 (C–C#), /14 (C–D), /15 (C–D#) and finally /16 (C–E). This also combines with VCO Hz mode to provide the Utonal Subharmonic series!

KBD: Keyboard Control Functions



Strange offers the additional functionality of acting as a very usable performance keyboard and clocked arpeggiator, via its Quantizer. Because this is an entirely separate mode we place the 'on' switch on Z rather than X. Moreover, because the keys are now of course 'live', secondary functions are not available[17] in Keyboard Modes and its additional functions are accessed by *directly* turning X and Y, not in combination with G#.

Keyboard Mode On/Off (hold G#, turn **Z**)

Holding note G# and turning Xάos Depth ('Z') engages the Keyboard Modes, switching off the Strange Algorithm (although the external Xάos and Noise outputs still operate). Anticlockwise (default) is normal Strange operation.

Noon is Arpeggiator Mode, where a TrEG is generated in sync with the clock (Int/Ext) whenever a note is held. If more than one note is held, these are played in an up/down pattern.

Clockwise is Pseudopolyphonic Mode: pressing one key will play it normally but any additional keys will alternate very rapidly to produce 'chiptune' style pseudopolyphony.

TrEGs and Portamento still apply to both Keyboard Modes, using whatever probabilistic settings were set when Strange was in normal operation, but must be set before switching into the mode.

Keyboard Mode Transpose (turn X while in Keyboard Mode)

Turning Xάos Speed ('X') while in Keyboard Mode transposes +/- 12 semitones. This does *not* need to be combined with a key press because the keys are live: you only need to turn the knob.

Keyboard Mode Octaves (turn Y while in Keyboard Mode)

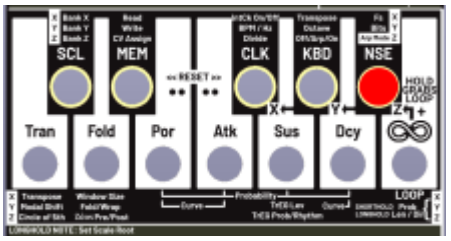
Turning Xάos Mod ('Y') while in Keyboard Mode transposes +/- 3 octaves. This does *not* need to be combined with a key press because the keys are live: you only need to turn the knob.

ρ & Δ + Keyboard Mode

Applying ρ & Δ while in Keyboard Mode introduces some fascinating effect. +ρ cycles up in octaves from 1 to 3 octaves depending on the amount, and -ρ does the same in the opposite direction, so equivalent to standard 'Up' and 'Down' arpegiator settings. Δ applies alternating motion but in a unique Strange formulation of the standard 'Up/Down'.

[17] As a result, you also can't save the mode as a preset, or CV allocate it—but would not need to anyway.

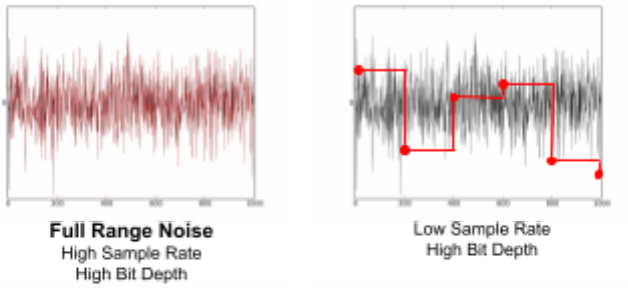
NSE: Noise Source Control Functions



Strange's algorithm processes randomness from one of two kinds of underlying noise source which you can switch between, alter and tap it off at the external Noise output. Since the Noise source feeds the Strange Algorithm, changing the NSE settings not only alters how the Noise output sounds, it also subtly affects Strange's behaviour as well (most easily heard at lower clock rates).

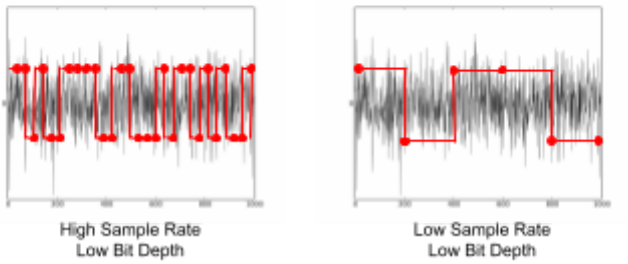
Noise Sample Rate Fs (hold A#, turn X)

Holding note A# and turning Χάος Speed ('X') changes the rate at which the noise source is sampled (default: maximum). Changing this creates super digital noise effects ideal for noise music, percussion and chiptune nostalgia!



Noise Bit Depth (hold A#, turn Y)

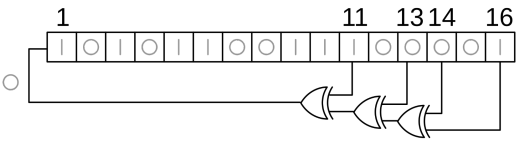
Holding note A# and turning Χάος Mod ('Y') changes the bit depth, from default maximum down to 1 bit (hi/lo). At low sample rate, minimum bit depth Strange's Noise Output therefore produces random gates.



Noise Algorithm (hold A#, turn Z)

Holding note A# and turning Χάος Depth ('Z') selects one of the two onboard noise sources. The default is true analogue noise but the alternative is our take on XOR Pseudo Random Number Generator noise, designed to include some semi-chaotic cycling, again ideal for chiptune retro!

Example XOR LSR PRNG where some positions in a Linear Shift Register are tapped off, XOR'd, and fed back into the start, creating a Pseudo-Random Noise Generator (*pseudo* random, because while 'random' from moment to moment, the output is deterministic and cyclic over longer periods).



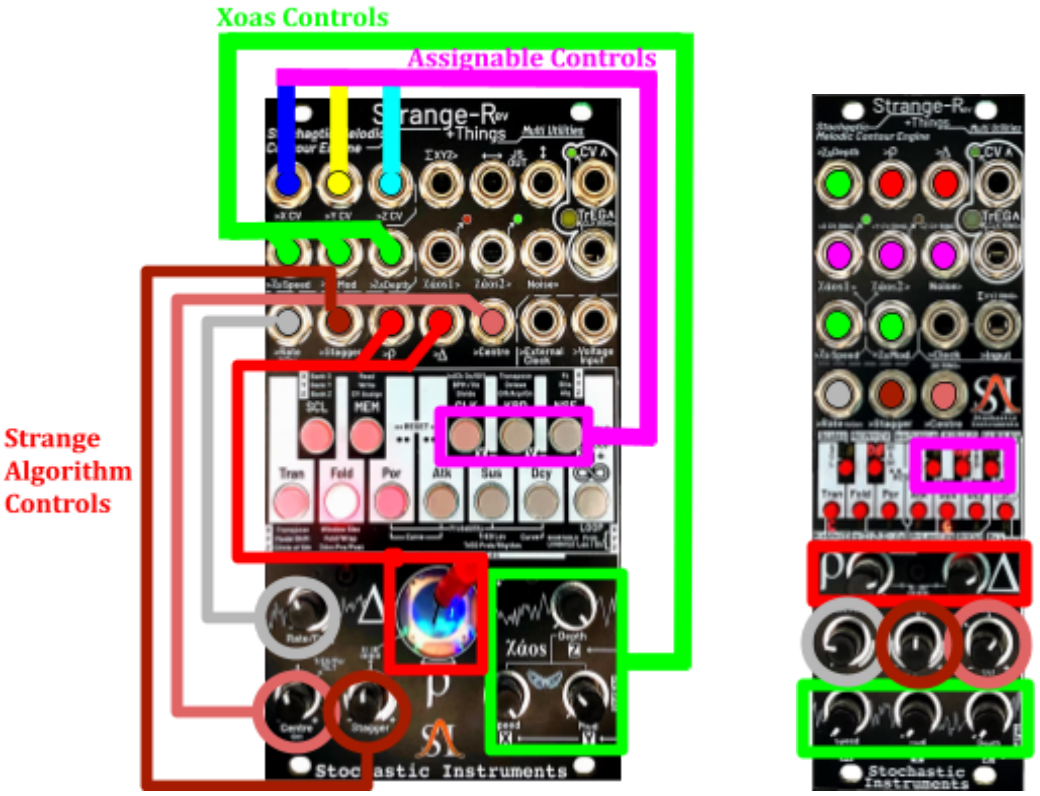
GOTCHA: because the PRNG noise distribution is completely different to the much more even (and non-repeating!) analogue default, and because this sits beneath the calculations of *all* the probabilities in the unit, TrEG and Portamento probabilities will operate slightly differently as well as the basic ρ/Δ algorithm controls.

External CV Control

Strange-R detects when a CV jack is inserted into a CV socket. For Primary Controls (those with a dedicated knob/joystick dimension) once it detects the jack the respective control switches to being an attenuverter for the incoming CV signal. This means, in order for the CV to take effect, the panel control needs to be non-zero: positive direction means the CV will have its effect in the 'natural' direction (e.g. +ve voltage = +ve effect), and negative direction means the effect of the CV will be in the opposite direction (e.g. -ve voltage = -ve effect).

Gocha: the CV will have no effect if the panel control is at zero (including the JS being vertical).

The three assignable CV inputs, X, Y, and Z control any of the secondary functions (TrEG, Protamento, Quantizer manipulations, Noise types etc.) but do not have attenuverter effects.

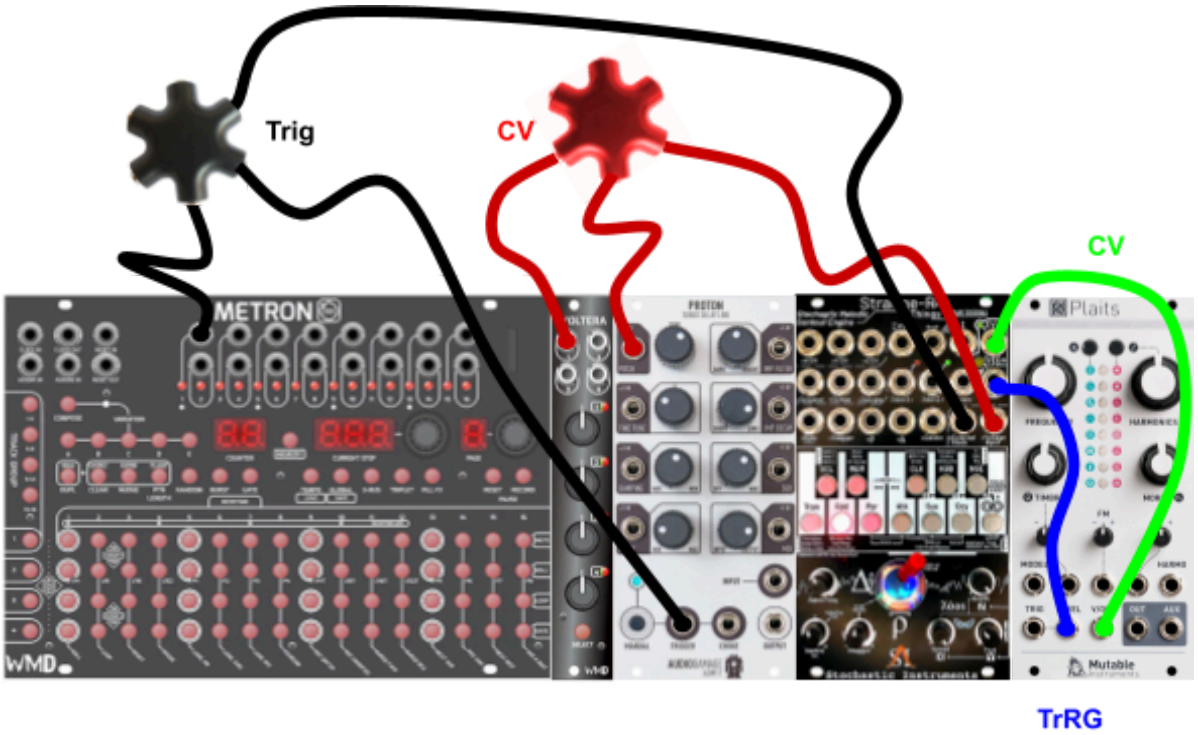


Extended Functionality

Live Input Variation/Recomposition Mode

One of Strange-R's most powerful features is its ability to recompose an external CV input live in performance. Route an external sequencer's output to Strange's Voltage Input and Strange bypasses its internal Strange Algorithm and replaces it with the external CV. You can also clock Strange from a common central clock, or from the sequencer's trigger output to retain the rhythm of the original sequence. Strange then acts as a live variation machine, spontaneously recomposing and riffing on your own original pre-composed material from the external sequencer, all under your performative control.

First, patch your favourite sequencer's CV output to Strange's Voltage Input, and its clock output to Strange's External Clock input. Then start your 'source' sequencer's sequence running. A common setup would be to mult to two voices so you have 'theme' and 'variation' running in parallel:



Even with Strange's quantizer disengaged you will notice it still does a pretty good job measuring the raw input voltage and passing it to CV Out[18]. However, the magic happens when you engage the quantizer with the scale used in the source material. Most obviously this ensures perfect tuning but much more interesting is the effect of ρ, Δ, Χάος, TrEG and other functions.

When Strange detects a voltage input, ρ subtly changes function, now offsetting up to +/- 1v to the CV across its range. This means the incoming melodic contour is walked through the degrees of the quantizer scale, so for example can harmonise the original at the 3rd above.

[18] See Calibration instructions, below.

Centre adds +/- 3v to the input so you can offset a baseline up to a complementary lead or a lead down to a baseline.

Δ adds increasing random offsets to the notes in its 'standard' +ve and 'alternating' -ve offsets, just as in Strange's usual operation.

Χάος adds offset over longer time spans, TrEGs still apply meaning some notes can attack, decay, and sustain, Portamento adds slides, clock divisions add differing rhythmic emphases and of course the input can still be Looped, varied, truncated, reversed and randomised!

Strange can even capture and manipulate the incoming CV as a Loop. However, there is an important point here:

GOTCHA: In External Input mode, we want the loop to play initially *as it is* and not be further affected by the ongoing input. Remember that in normal operation, ρ and Δ first generate the contour and then move the loop around. With an External Input, you probably don't want the ongoing input CV to immediately start moving the loop around. Therefore, **as soon as you capture a Loop in External Input mode, the input is then ignored so the Loop doesn't move around. At that point, if you don't want the input to offset the Loop as it plays you need to disconnect the input. However, as soon as you touch either ρ or Δ Strange *will* start listening to the input again and it *will* start to offset the Loop (if that's what you want).**

Obviously this mode is dependent upon having a well calibrated unit so please refer to the Calibration section. Remember that Δ applies random offset, and that the input voltage might be slightly above or below the '0' point on the quantizer, so extreme high or low notes in the original sequence may nudge to the note either side. Since this is a variation machine, this represents a feature not a bug. 🙂

Audio Sampling

While not quite a Fairlight, Strange is, believe it or not, also an audio sampler. By clocking the External Voltage Input into audio, any signal fed toVoltage Input is sampled (although of course with no anti-aliasing filtration). Grabbing a loop repeats the audio.

Negative Sampling

Strange actually 'invents' a new concept in sampling, *Negative Sampling.* Traditional digital samplers use a fixed sample rate and alter the *playback* speed of the audio they record to obtain different pitches. However, because Strange permits you to separate the 'clock' from the 'sampler' you can sequence the *sample rate* as the audio goes in! This produces some very interesting results because a fixed input pitch sampled at a *lower* sample rate will play back *higher* and sampled at a *higher* rate will playback *lower.* The effect is exactly akin to musical inversion but is fascinating when applied to full range audio rather than just fixed tones, (although they work very musically too!)

+Things: Utility Functions

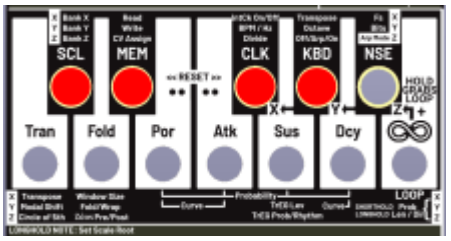
We have designed Strange to offer maximum utility even when not being used in its 'main' role by bringing the outputs of its 'micropatch' submodules to the front panel. These, alone, or in self-patched combination permit Strange to offer a very wide range of utility functions.

- Change Detector
    - TrEG set to Rhythm Mode (G+Z fully CW)
    - External Input connect
    - When input changes, Strange outputs a TrEG
- Random Gates
    - Set Noise Fs to minimum
    - Set Noise Bit Depth to minimum
    - Send Noise Output to chosen EG/Drum module/Sequencer…
    - …or self-patch back to Strange!
- Envelope Generator
    - Connect external sequencer Gate out to External Clock
    - Set to External Clock
    - Set TrEG Attack/Sustain/Decay to 100% to taste
    - Connect TrEG out to your external sequencer voice's VCF/VCA
- Stochastic Envelope Generator
    - As above but with <100% probability
- Chaotic LFO
    - Connect Χάος 1 (slower) and/or Χάος 2 (faster) to your desired destination
- Noise Source
    - Noise Output
- Drum Module
    - Change Noise Fs (A# + X)
    - Assign to CV In X
    - 100% Decay Prob
    - Mult TrEG out to external VCA/VCF
    - Mult TrEG out self-patched to CV In X
    - Send Noise Output to VCA/VCF
    - Externally Clock Strange from your drum sequencer channel
- Transposer
    - Connect Voltage Input
    - Set all Quantizer notes lit
    - Use CV Output
    - Transpose to taste
- Sample/Track & Hold
    - Connect Voltage Input
    - Set to External Clock
    - 100% Sustain
    - CV Out to VCA In
    - TrEG Out to VCA CV In
- Slew Limiter
    - Connect Voltage Input
    - 100% Portamento

    - 100% Sustain
    - CV Out to destination
- VCO with Sync
    - Audio rate clock
    - Output to VCF/VCA from TrEG
    - Audio rate input to Clock In
- Sampler
    - Audio rate clock
    - Audio input to External Voltage Input
    - Loop
- Quantizer
    - External Voltage Input
    - CV Out
- CV Recorder
    - External sequencer gate to External Clock
    - External sequencer CV out to Voltage Input
    - Loop
- Stepped Random Source
    - +ve Δ
    - Portamento 0%
- Smooth Random Source
    - +ve Δ
    - Portamento 100%
- Arpeggiator
    - +ve ρ
    - Quantizer set to arpegio
    - Fold as Wrap or Fold
- Clock Divider
    - External clock input
    - Clk Divide (F#+Z)
- Summer
    - CV inputs to X/Y/Z in
    - Output from Σ Out
- Manual Trig/EG
    - Set TrEGs to desired probabilities
    - Push joystick (BigHat only)

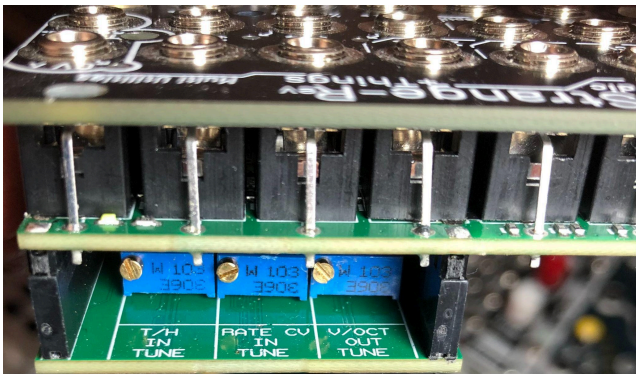
Reset & Calibration

Reset



To reset the unit simply press and longhold notes C# + D# + F# + G#. After 5 secs all notes will flash on and the unit will reboot.

Calibration



Strange has three controls which need to be accurately tuned: its CV Out, Voltage Input and v/oct Rate input. We calibrate Strange before we ship but to protect against drift over time we include trim pots in case you need to re-cal. The unit has to be powered to achieve this so you need to be very careful not to short anything on the board.

We strongly recommend powering off before you remove it from the case (in case you touch anything in extracting it), *then* powering on with it safely out but connected. We'd also recommend a plastic trimmer tool because they don't fall out of the slot on the screwhead, and it doesn't matter if you touch it on anything .

Strange has special calibration modes to simplify the procedure, and should be done in order:

1) CV Out

This ensures Strange's CV Out is properly scaled in perfect octaves over its range.

a) Start by connecting Strange's CV Out to a well calibrated VCO (or oscilloscope).
b) Set the Centre control to noon (0)
c) Enter CV Calibration Mode by Resetting and then holding **C** (for **C**V Calibration!) while the lights are counting up from the left
   i) This sets all controls to zero and disables them (ρ / Δ / Stagger / Χάος)
   ii) Disables the Quantizer
   iii) Automatically sends incrementing volts (octaves) out of CV Out at 2Hz
   iv) Centre is now 'absolute' tune but don't touch this yet
d) You should be hearing perfect octaves and/or seeing perfect 1v increases on your oscilloscope.
e) If not, turn the V/OCT OUT TUNE pot until all the octaves are in tune with each other and/or you read voltages increasing by exactly 1 volt each step. Turning clockwise will make the higher notes sharper, anticlockwise, flatter.
f) If there is any absolute offset (e.g. 1.02, 2.02, 3.02, 4.02, 5.02) this can be tuned out with the Centre pot.

Your CV Out is now calibrated!

2) Voltage Input

This ensures the voltage sent to Strange's External Input (for Live Input Variation Mode) is reproduced correctly by Strange. We are checking this from Strange's CV Out, which is why the *output* has to be cal'ed first.

a) Disengage all notes from the quantizer
b) Set to External Clock
c) Re-zero ρ (JS to centre-centre on BigHat)
d) Confirm Δ, Centre, Stagger and Χάος are still all zero.
e) Send a constant mid range voltage to External Input and use this to exactly centre ρ & Δ (i.e. so Strange outputs as close to a constant voltage as possible).
f) Create an external sequence of ascending octaves, from 0v up (0v, 1v, 2v etc.) on a well calibrated sequencer.
g) Connect its CV Out to Strange's Voltage Input, and its Trigger/Gate output to Strange's External Clock input.
h) You should be hearing (near) perfect octaves and/or seeing (near) perfect 1v increases on the oscilloscope.
i) If not, turn the "VOLTAGE IN TUNE" pot until the highest notes are in tune with the lowest.
    i) Some random drift is still acceptable because the quantizer is disengaged and there will inevitably still be some microtonal Δ jittering the output. This doesn't matter: in normal operation you'll be engaging the Quantizer so this procedure is only designed to ensure the *Quantizer* is as correctly aligned as possible with the raw input voltage.

Your Voltage Input is now calibrated!

3) Rate v/oct Input

This ensures the voltage sent to Strange's Rate v/Oct Input (to control its Internal Clock) tune correctly. This is most relevant when using Strange as a back up VCO. This is done last because it re-uses the octave sequence from Step 2.

a) Connect the octave sequence you made for Step 2 now to the Rate v/oct input
b) Disconnect the external clock from step 2
c) Disconnect the CV Out from your VCO from Step 1
d) Connect Strange's *TrEG output* (not CV) to an audio source/oscilloscope
e) Reset Strange
f) Ensure it's set to Internal Clock
g) Switch to VCO Hz mode (F# + Y 100%)
h) Set 100% TrEG Decay (A + X 100%).
i) You should be hearing perfect octaves and/or seeing perfect 1v increases on the oscilloscope.
j) If not, turn the RATE CV IN TUNE pot until the highest octaves are in tune with the lowest.

Your Rate CV v/Oct is now calibrated!

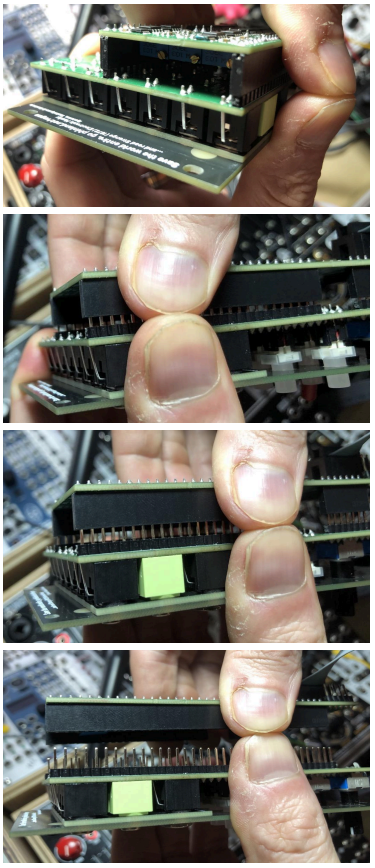
Swapping Hats

If you have both hats and one MCU Board you can swap between them as follows:

1. Turn off the power

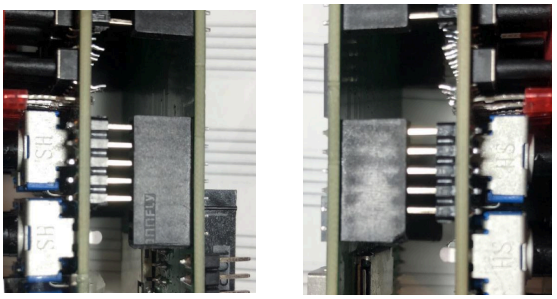
2. Extract the module from the case and remove the power ribbon

3. Very carefully prize apart the MCU board from the hat along the lines of the two black pin connectors. They fit fairly snuggly so require a little *but not too much* force.
    a. The MCU board has 'female' sockets, hats have 'male' pins.
    b. You need to take care not to bend the pins on the hat when pulling the MCU away.
    c. This means ensuring you are always prising them apart in line with the pins (i.e., not at an angle).
    d. The best way to achieve this is to hold the hat in one hand, the MCU in the other, and line up your two thumbs and two index fingers against each other.
    e. Then, lever your thumbs and indexes *together* while gripping the edges of the black rail so the pads of your thumbs/indexes pull *apart.*
    f. Go more gently as the two come apart and make sure you don't 'roll' the boards away from each other at the last moment—always keep them in line with the pins.



4. If you do bend the pins they will most likely be the top or bottom 1 or 2.
    a. Carefully bend these back with some needle nose pliers and they should be fine.

5. Now line up the other hat with the pins and carefully push the two halves together.
    a. SmallHat does not use all the pins of the bottom connector, so you need to align from the *top* of the long *top* connector.
    b. The bottom sockets of the shorter connector don't have pins to mate with on the SmallHat—this is absolutely fine and correct.
        i. The right side misses two, the left side misses one.

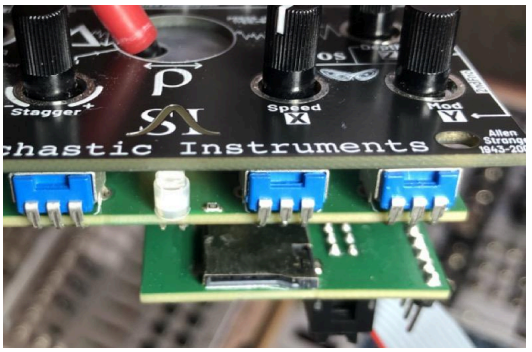


SI is not responsible for damage resulting from changing hats so please take great care with this procedure.

Firmware Update

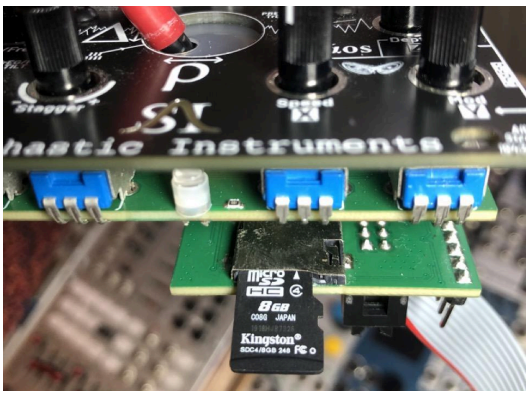
The module includes a Micro SD Card slot on the main MCU board (the board with the large microchip) to accommodate potential future Firmware updates. These are made available on the Site free of charge. To update your firmware:

1. Using a compatible adapter, ensure on your computer that your Micro SD Card is empty and formatted to MS DOS Fat32
    a. In Apple OSX use Disk Utility
    b. In Microsoft Windows use Disk Management



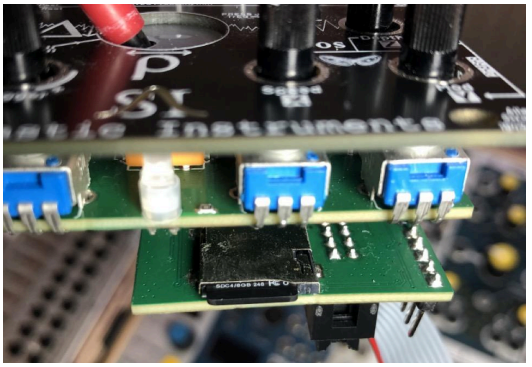
2. Copy the strange.bin file to the card
    a. It must be named exactly strange.bin and be the only file on the card

3. Remove the Micro SD Card from the adapter and, with the power to Strange disconnected, insert it into the slot on the main MCU board.
    a. The card needs to be orientated text-up with the module the right way up.

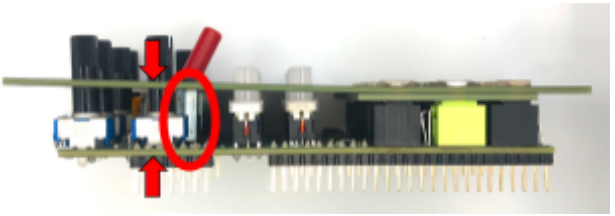


4. Whilst holding F power up the module.
    a. Take great care not to interrupt the power of remove the card during the update process.

5. The new Firmware will load over a few seconds and then the unit will automatically operate normally.
    a. Once loaded on, the SD Card does not need to be present for future startups.

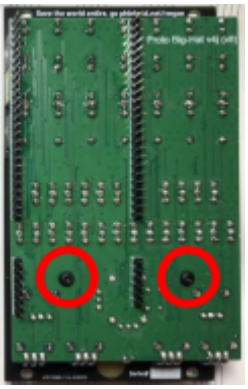


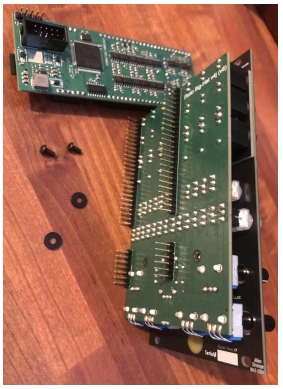
Joystick Resistance Adjustment



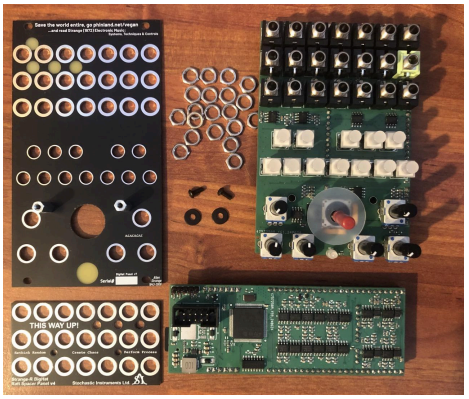
BigHat users looking side on to the module may notice a very slight flex in the top board. This is entirely by design and does not show when the unit is screwed into your rack. The purpose is so you can adjust the resistance of the joystick from 'whisper light' to heavy. Turning the two 1.5mm hex screws either side of the joystick hole on the faceplate changes the pressure on the translucent joystick gator: think of it as the 'fine' control.

For 'coarser' resistance you need to remove the washers spacing the hexagonal standoffs from the hat's PCB. To simplify this we manufacture the BigHat with these washers at the *bottom* of the standoffs so you don't need to remove the front panel. Instead, remove the MCU driver board and unscrew the two 1.5mm hex screws *from underneath,* give a little shake and the two washers will drop out: then just re-insert the screws. As you tighten them up the joystick will give you increasing resistance. Some units have black plastic ones like here, others, shiny metal ones, but the function is identical.

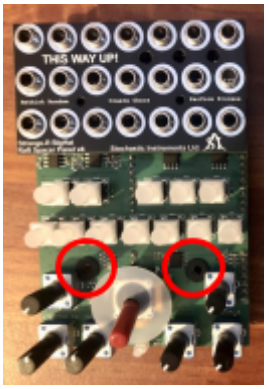




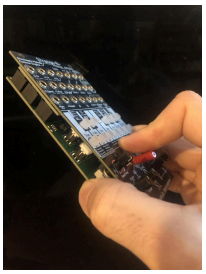
Putting the washers back is a *little* trickier but still a sub-5 minute job. Remove the front panel by unscrewing the 8mm jack nuts and the two 1.5mm hex nuts *from the underneath* to disassemble like this. Note that the jack nuts are quite close together and some 8mm bits can be too large to easily rotate.

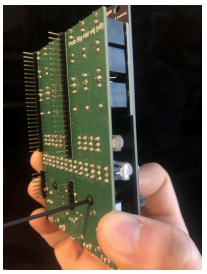


Next, place the two washers over the holes in the bottom PCB, not forgetting the spacer "Jeoff" board that fits over the jacks (named after my father-in-law whose idea it was!).



Place the front panel back on ensuring you don't disturb the washers' position when you guide it over the 12 note buttons. Then push the panel down with your index finger to hold them in place while you turn it over to replace the hex screws.





Credits and Acknowledgements

2018 conceptual design & software, 2024 functionality/panel layout: **Phineas Head**
2024 Hardware & Firmware design: **Stuart MacVeigh**

Mechanical design consultation: **Jeffrey Cook** & **Anthony Head** (shoutouts to the Dads / Oracles!)

Manual: **Phineas Head**

Beta Testing: **Matt Ward, An On Bast, Dan Legg**

UK production fabrication, consultancy & testing to April 2025: Dan Legg / Fully Wired Electronics
UK production fabrication, consultancy from June 2025: Isaac Chasteau

Stochastic shoutouts to the hero composers that inspired it: JSB JMC KS IKX SMR MLN MF BPGE
Special thanks to Allen Strange

Phin shoutouts to Lucy, Madds, Marsha, Tony, Dusty, Beansprout, Mr. Whizzah 🐾

No animals were harmed in the making of SI products. On the contrary, it's kept Beansprout in chewsticks.
[Save the World Entire: phinland.net/vegan]

Appendix 1: Preset Scales, Modes and Reflections

Root note and transformational centre are assumed to be identical (non-root centres will produce different results). All names are hyperlinked to https://ianring.com

Bank X (C#+X)		
Note	**Scale**	**Available Modes / *Reflections*** (popular/useful transformations/scales shown in **bold**)
C	**Major 7th Arpeggio**	Lothic = **i6**, Phratic = **vi sus4**, Raga Lavangi = **N⁶**
C#	**Dominant 7th Arpeggio**	Lonic = ♭ **VI7**, Phradic = **viº sus4**, Bolic = **II7** *Raga Haripriya = **iiº7**, Aeraphic = **iø7**, Byptic = **iM6**, Saric = **vi sus#4***
D	**Minor 7th Arpeggio**	Lahuzu 4Tone 3 = **I6**, Karen 4Tone 3 = **VI sus4**, Lahuzu 4Tone 4 = **ii7**
D#	**Major sus4 Arpeggio**	BOBian = **VI sus4**, NEMian = **V sus4**, JOCian = **IV sus4** *COPian = **iv sus2**, BABian = **i sus2**, GIKian = ♭ **vii sus2**, PUNian = **vi sus2***
E	**Minor Pentatonic**	Major Pentatonic, Scottish Pentatonic, **Blues Minor**, Suspended Pentatonic
F	**Major / Ionian**	**Dorian, Phrygian, Lydian, Mixolydian, Aeolian, Locrian**
F#	**Melodic Minor (ascending)**	Dorian ♭ 2, **Lydian Augmented**, **Acoustic (Lydian Dominant, Overtone, Bartók)**, **Major-Minor**, Minor Locrian, **Superlocrian (Altered Scale / Ravel)**
G	**Harmonic Minor**	Locrian♮6, Major Augmented, **Lydian Diminished**, **Phrygian Dominant**, Aeolian Harmonic, **Ultralocrian** *Harmonic Minor Inverse*, *Lydian Augmented #2*, *Locrian 𝄫7*, ***Harmonic Major***, *Dorian ♭ 5*, *Phrygian ♭ 4*, *Lydian ♭ 3*
G#	**Blues Scale I**	Gycrimic, Pyrimic, Raga Hamsanandi, Mixolimic, Dadimic *Gathimic, Raga Bhanumanjari, Phrynimic, Raga Gurjari Todi, Raga Brindabani Sarang, Ionacrimic*
A	**Blues Scale II**	Spanish Phrygian, Gycryllic, Lyryllic, Magen Abot 2, Gregorian Nr.4, Moptyllic, **Bebop Major**
A#	**Bebop Minor I**	Stylyllic, Dalyllic, Ionyphyllic, Zaptyllic, **Shostakovich**, Gathyllic, Mixopyryllic *JG Octatonic, Mocryllic, Aeolyryllic, Baryllic, Stogyllic, Ionidyllic, **Bebop Locrian**, Stalyllic*
B	**Dorian Bebop (Mixodorean)**	Quartal Octamode, Quartal Octamode R10, **Prokofiev**, **Dominant Bebop**, Raga Mukhari, Phrygian/Locrian Mixed, Ichikotsuchô

Bank Y (C#+Y)		
Note	**Scale**	**Available Modes / *Reflections*** (frequent/useful transformations shown in **bold**)
C	**Double Harmonic**	Lydian #2 #6, **Ultraphrygian**, **Double Harmonic Minor**, Asian, Ionian Augmented #2, Hungarian Romani Minor 4th Mode
C#	**Hungarian Major**	Alternating Heptamode, Harmonic Minor ♭ 5, Hungarian Major 4th Mode, Hungarian Major 5th Mode, Mela Sadvidhamargini, Nohkan Flute Scale *Moravian Pistalkova, Epylian,* ***Jeths' Mode****,* *Kathian, Lylian,* ***Stravinsky's Petrushka****, Pogian*
D	**Neapolitan Minor**	Lydian #6, Mixolydian Augmented, **Ukrainian Dorian**, Locrian Dominant, Raga Trishuli, Porian *Raga Supradhipam, Golian, Raga Senagrani, Harmonic Lydian, Minor Romani Inverse, Sabach ascending, Pynian*
D#	**Neapolitan Major**	Leading Wholetone, Aeroptian, Lydian Minor, Major Locrian, Storian, Leading Wholetone Inverse
E	**Chromatic Hypodorian**	Chromatic Mixolydian, Chromatic Lydian, Chromatic Phrygian, Chromatic Dorian, Chromatic Hypolydian, Chromatic Hypophrygian *Ch Hypodorian Inv, Ch Hypophrygian Inv, Ch Hypolydian Inv, Ch Dorian Inv, Ch Phrygian Inv, Ch Lydian Inv, Ch Mixolydian Inv*
F	**Phradian**	Aeolorian, Raga Nagabharanam, Dalian, Raga Bhāvani, Zolian, **Mixolydian #2** *Thonian, Jhankara Bhramavi, Stadian, Thodian, Dogian, Ratnangi, Marva That*
F#	**Sarian**	Zoptian, Mela Varunapriya, Byptian, Darian, Lonian, Raga Cudamani *Phrolian,* ***Maj Bebop Heptatonic****,* ***Modified Blues****, Kycrian,* ***Epygian****, Zaptian, Kagian*
G	**Donian**	Aeoloptian, Panian, Lodian, Solian, Ionolian, Blues Heptatonic II, *Ionopian, Aeologian, Ceiling Scale, Sygian, Phralian, Phrogian, Rathian*
G#	**Lygian**	Logian, Lalian, **Verdi's Scala Enigmatica Descending**, Phrocrian, **Foulds' Mantra Of Will Scale**, Laptian *Zarian, Phrythian, Rorian, Bolian, Bothian, Mela Divyamani, Kodian*
A	**Chromatic**	-
A#	**Messiaen 1 (Whole Tone)**	-
B	**Messiaen 2 (Octatonic I)**	**Diminished (Octatonic II, Pijper's Scale)**

Bank Z (C#+Z)		
Note	**Scale**	**Available Modes / *Reflections*** (frequent/useful transformations shown in **bold**)
C	**Messiaen 3**	Messiaen 3 R1, Messiaen 3 R2
C#	**Messiaen 4**	Messiaen Mode 4 R1 (Tcherepnin Octatonic), Messiaen Mode 4 R2, Messiaen Mode 4 R3
D	**Messiaen 5**	Messiaen Mode 5 R1, Messiaen Mode 5 R2
D#	**Messiaen 6**	Messiaen Mode 6 R1, Messiaen Mode 6 R2, Messiaen Mode 6 R2 (van der Horst Octatonic)
E	**Messiaen 7**	Messiaen Mode 7 R1, Messiaen Mode 7 R2, Messiaen Mode 7 R3, Messiaen Mode 7 R4
F	**Verdi's Enigmatic**	Aeracryllic, Epygyllic, Half-Diminished Bebop, Lanyllic, Phrynyllic, Lycryllic, Ionyptyllic, *Dathyllic, Boptyllic, Bagyllic, Mathyllic, Styptyllic, Raga Bhatiyar, Bacryllic, Aerygyllic*
F#	**Scriabin's Prometheus (Mystic)**	Bythimic, Padimic, Boptimic, Stogimic, Eskimo Hexatonic 2 ***Takemitsu Linea M2**, Gygimic, Aeragimic, Epothimic, Salimic, Lyptimic*
G	**Debussy's Heptatonic**	Sadian, Dothian, Moptian, Aeryrian, Boniana, Badian *Katoptian, Ponian, Kadian, Gynian, Thyphian, Polian, Thanian*
G#	**Schoenberg Hexachord**	Raga Lalita, Ionalimic, Raga Kalagada, Mydimic, Thyptimic *Raga Jivantini, Bogimic, Mogimic, Docrimic, Raga Suddha Mukhari, Raga Bauli*
A	**Espla's Scale**	Spanish Octamode R1, Spanish Octamode, SpOcta R10, Sp Octa R9 (**Hamel**), Sp Octa R8, Sp Octa R6, Sp Octa R4 *Ionocryllic, Lydian/Mixolydian Mixed, Epiryllic, Aeradyllic, Staptyllic, Magen Abot 1, Raga Mian Ki Malhar, Epocryllic*
A#	**Bartók Beta Chord**	Phrothitonic, Katycritonic, Ionalitonic, Raga Manaranjani I, *Pathitonic, Raga Priyadharshini, Thyritonic, Thoptitonic, Bycritonic*
B	**Persichetti**	Epacrian, Sathian, Lathian, Elephant Scale, Kynian, Stynian *Stagian, Lothian, Phrycrian, Kyptian, Ionylian, Ionanian, Aerothian*

Appendix 2: Busoni's Definition of Melody

One vital ingredient to Strange-R was Ferruccio Busoni's definition of melody, which formed the basis of one of the most scintillating lectures on music Phin has ever heard, given by the composer Michael Finnissey with whom he was ridiculously lucky enough to have studied:

*"**Absolute melody**: a series of*

(0) repeated
(1) rising and falling intervals
(2) which, rhythmically articulated and set in motion
(3) contains within itself a latent harmony
(4) and renders a state of mind
(5) which can exist independent of a text as expression
(6) and independent of an accompaniment as form
(7) and whose execution effects no change in its nature
(8) through choice of key or timbre."

Ferruccio Busoni, 22nd July 1913

Let's unpack that:

(0) "*Repeated*": not so much an essential *ontological* component of melody—a melody is still a melody if it never repeats—but rather an important *practical* one in terms of the formation of a 'melodic object' or 'primitive' in the listener's mind and memory, which can be repeated, perhaps in transposition, and which permits the formation of micro-, meso-, and macro- scale structures within the complete structural form.

(1) "*Rising and falling intervals*": importantly, *not 'notes' per se but intervals*. It's the intervallic relations, the *steps* and *jumps*, which are fundamental, and their diatonic or non-diatonic manipulation and transposition. For example a melody in C Major may start *absolutely* as C-D-E-D but can be better thought of *relativistically* as Start-Up-Up-Down, with that meso-pattern then transposed, reversed, inverted into phrases (e.g. D-E–F-E).

(2) "*rhythmically articulated and set in motion*": melody requires horizontal movement in *pitch over time* but this also requires the understanding that *melodically* (although not *acoustically*), pitch is an *inseparable* dimension from rhythm. That is, the pitch series and duration series of a melody, while individually extractable, are inextricably linked. That means the same series of pitches with a different rhythm (or the same rhythm with different pitches) would form a different melodic percept in a listener's mind (even if it were heard as 'related' to the original).

(3) "*contains within itself a latent harmony*": probably the most technically complex point, and well beyond the scope of a user manual(!) but the point is that we derive chords (harmony) from scales (melody) perhaps very anciently via the reverberant acoustics of ritual spaces. The way we write a melody (specifically where certain 'structural' rather than 'passing' notes fall) can *imply* the chords through which it moves *without having to actually play those chords.* This, and much else besides, is best understood by listening to Bach—specifically in this case the Suites for Unaccompanied 'Cello, BWV 1007—1012.

(4) "*renders a state of mind*": Busoni probably meant the *affective* (=emotional) implication of this, whether by some intrinsic acoustic property or, much more likely, social construction, but we can also take this to mean the formation of the 'understanding' or 'concept' of the whole melodic form in the listener's mind.

(5) "*exist independent of a text as expression*": again, Busoni probably meant that melody can convey meaning(?) although we know from modern computational linguistics that it can't do so *propositionally* or *syntactically* like real language can (i.e. a melody can make you feel 'happy' or 'sad' but can't say "The cat sat on the mat" or "The mat sat on the cat" or "Colorless green ideas sleep furiously".)

(6) "*independent of an accompaniment as form*": i.e. it has internal *structure* which is why repetition is an important practical element (see above).

(7) "*whose execution effects no change in its nature*": implies its 'implementational invariance'—that is, a melody is the same melody whether played on Eurorack or euphonium, loud or soft, fast or slow. It is therefore primarily 'music' (*organisation*) not 'sound'.

(8) "*through choice of key or timbre*": …or registration or starting note—'implementational invariance' as above. It may be worth noting that (extremes of) tempo *may* change the musical implication, and modal interchange will create a related but distinct melody.

He continues: "Melody, at first independent, was combined over the years with an harmonic accompaniment and later blended with this harmony into an indissoluble unity from which the constantly evolving poly-harmony has recently sought to free itself. In contradiction to well-established points of view, it must be insisted here that melody has continued to evolve, has grown in line and expressive potential, and must eventually attain universal dominance in composition. Its quality was lowest, and the importance accorded to it most negligible, during the age of 'melodious' operas and salon compositions, for melody was then fashioned to be as readily intelligible as possible and as separable from the other components in the musical work of art—**a corrupt folk song**. **It is characteristic of melody's period of decline that it always appeared isolated and confined to the leading voice, usually the upper voice.** The attempts to contrast it with one or more other melodic voices were at first extremely feeble, for the latter were invariably made subordinate to it and remained secondary."

Considered in the light of some pop music's proclivity for adding a hermetically sealed 'topline' over a 4-chord cycle this can make slightly depressing reading for we old farts of the older generation…but I may just be being nostalgic. Either way, listen to Doktor Faust, not the Mephistophelian Pact of a cheap topline. 🙂

Take time to experiment with per-note ρ & Δ and their interaction with global ρ & Δ: therein lie rich rewards.

Appendix 3: Performable Process in Strange-R & SIG

SI's Performable Processes ethos is founded on the principle that we ascribe expressive power, and perhaps therefore artistic value, as a function of the depth of input of the performer. Triggering a recording (perhaps pressing play on a shop's muzak system) is less artistically interesting than performing a Bach Trio Sonata at the organ[19]. A *purely* generative system which requires only that it be set into motion might fall closer to the muzak end of the distribution. However, a partially generative system yet whose parameters are sufficiently multiple and interdependent to afford wide and deep human control over the micro-, meso- and macro-aspects of a performance might be considered to bear a close enough family resemblance to 'playing an instrument' to count as such—even if the raw materials and constraints are different. More succinctly, the process becomes the instrument, its parameters the keys, frets, and bow.

Part of Strange-R design ethos is thus what we've dubbed *Regelmusik* (Rule Music), essentially a stricter take on Systems Music (as defined in Nyman 1974) where mathematical rules *directly* link a note's pitch to some other parameter, for example articulation or tempo, or, via note ρ & Δ, the next pitch—but *in performance*. That's critical because it provides the recursion necessary for a complex system to blossom. Complexity happens when simple things cause complex results and we're very pleased with the amount of melodic power that can be wielded by a performer and Strange-R working together.

Moreover, we've worked hard to make its approach 'sound' complementarily distinct from SIG, and indeed, using the two in combination demonstrates this very well. A typical use-case would be Strange-R as a lead line with SIG providing harmony and/or drums/samples/counterpoint, but all these roles are of course interchangeable. What's certainly the case is that Strange-R is *not* just a single channel SIG—in several respects, the pure melody engine of Strange-R is much more complex than 'SIG in Linearity mode', while at the same time, SIG offers much more in terms of durational control and of course per-note probability. The way to compare and contrast them is to consider the relationship between, and differing capabilities of, a saxophone and a piano: they excel at different things but often work best in combination.

[19] We leave it up to the reader to decide where, and on what terms, they wish to place all other kinds of musical activity on that continuum.

Appendix 4: A Word on (Design) Complexity

In common parlance 'complexity' is used in two ways within our discipline, to mean either positively 'functional breadth/depth' or negatively 'overly dense ergonomics'. At the same time, the formal *mathematical* definition really concerns the interaction of *simple* elements to produce deep output behaviour.

By way of example, consider the question "Is a violin complex?". In one sense it's a small wooden box with a short plank of wood you press the strings against, and makes a noise when you scratch them with horsehair. Those are *very* simple ideas. In another sense it's an instrument that takes years to play at just a basic amateur level, a lifetime to master professionally, and on which the world's greatest composers are still exploring its full capabilities, the instrument having survived in one form or another for thousands of years, in highly varied musical cultures (from rebab, to ektara, to masenqo, to erhu).

Strange's design—conceptual, mechanical, hardware and firmware—took a long time to perfect but we're very proud of the results. It uses simple and immediate performative controls to output a huge range of rich musical output and which can be radically changed in character by simple and readily perforable actions. While there are many additional secondary functions each is a further *simple combinatorial* addition to the overall system. They should be considered in terms of the 'possibility space' of the instrument, in the same way we would consider 'hand weight', 'bowing point', 'finger position', 'string length' and so on as simple elements contributing to an overall complex output.

In terms of the additional functions (VCO, Noise, Keyboard etc.) these are there *if you need them* but are by no means compulsory to its use. We do argue though that hp is a valuable resource to the working modular synthesist and so the ability to solve additional musical problems even when not in its primary mode of operation is a valuable asset which easily justifies the inclusion of additional functionality.