PedalScope Methods, Explained

A plain-language companion to the technical reference

PedalScope

Contents

1 What this companion is

The technical reference states each measurement's method exactly, with equations and every parameter. This companion says the same things in plain language, one measurement per chapter, using the same defined terms — the glossary at the end is generated from the same source the technical document uses, so a word means one thing in both. It never says what a measurement sounds like: the instrument measures, and listening is yours.

The chapters follow the technical document's headings in the same order, so the two can be read side by side: where this document says "the app cuts each blip out with a window", the technical document gives the window's shape and width. Every number quoted here is read from the same methods manifest the technical document reads, which the app's own code writes out; nothing is typed by hand. Each defined word is a link to its entry in the glossary at the end.

2 Harmonic Distortion

2.1 What this measurement tells you, and what it cannot

Play one clean note into a distortion pedal and more than one note comes out. The extra notes are not random: they sit at exact whole-number multiples of the note you played — twice its frequency, three times, four times, and so on. Those multiples are the note's harmonics. The Harmonic Distortion measurement lists how much of each harmonic the pedal adds, for every note across the range, at one loudness.

Two ideas make that sentence work, and the rest of the chapter leans on both.

Why a changed shape is new notes. A pure tone, drawn against time, is a smooth sine wave: one frequency and nothing else. Any wave that repeats at the same rate but has a different shape — flatter tops, a kink, a lopsided bottom — is made of that frequency plus its multiples. That is not a way of speaking; it is what "shape" means for a repeating wave. A wave is exactly a stack of harmonics, and changing its shape changes how much of each harmonic is in the stack. The fundamental is the note itself; each harmonic order names which multiple an added tone is.

What clipping does to a sine. A distortion circuit has a ceiling. Below it the wave passes through unchanged; at it the peaks are flattened or rounded — that is clipping. Figure 1 shows three versions of it, computed exactly, with each harmonic's level in decibels relative to the note that went in. A soft clipper (the top row, the tanh curve that serves as the worked example throughout) rounds the peaks and produces a ladder of harmonics that falls away quickly; a hard clipper flattens them and produces a ladder that falls away slowly, so the high harmonics stay strong. Both of those treat the top and bottom halves of the wave the same way, and both produce only the odd harmonics (third, fifth, seventh, . . .); the even ones are exactly absent. The third version clips the bottom half less than the top, and the even harmonics (second, fourth, sixth, . . .) appear. That is the whole rule: symmetry gives odd harmonics only, and any difference between the two halves adds the even ones.

The measurement, then, is a table: for each harmonic (the app reads 9 of them, the note itself and eight above it) and for each note in the range, how loud that harmonic is compared with the note that went in. It is drawn as one curve per harmonic across the notes — the harmonic response — with a single-number summary, the total harmonic distortion, beside it. Everything is measured relative to the strength of the test tone (input-normalized), so a plain wire reads as "0 dB" for the note itself and nothing for any harmonic, and two measurements made at the same drive on different rigs can be laid over each other.

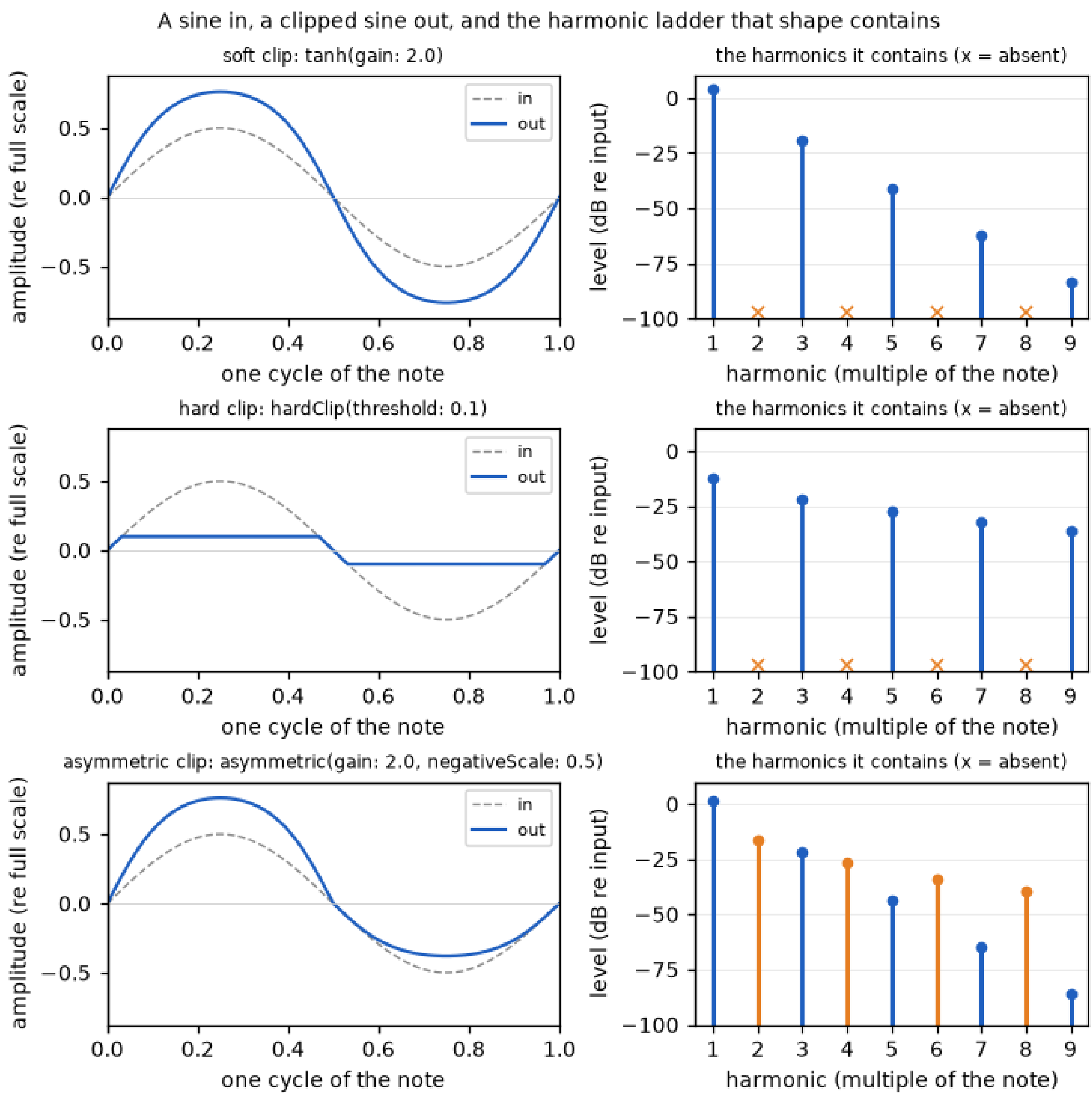

Figure 1: Three clippers, computed from their formulas at the amplitude the worked example uses (0.5 of full scale). Left: one cycle of a sine in (grey, dashed) and what comes out (blue). Right: the exact harmonic content of the output, in decibels relative to the input — each step of 20 down is a tenth of the size. Odd harmonics in blue, even in orange; a cross marks a harmonic that is exactly absent.

What it cannot tell you. Stated once, so nothing below has to hedge.

1. **Only one loudness.** One sweep is one drive level. A pedal that cleans up when you play softly, or hardens when you dig in, does so at other loudnesses this measurement never visits; the Gain Map and the Compression measurements own that axis.

2. **Only one note at a time.** Two notes played together make the pedal produce sums and differences of their frequencies as well as harmonics. That is a different measurement — Chord IMD — and nothing here predicts it.

3. **A number can be hiss, not harmonic.** The analysis reads whatever lands in each harmonic's slot, and the pedal's own noise lands there too. A separate check (harmonic shaping, below) says whether a reading is something the circuit did or just its noise; a reading is not automatically a harmonic.

4. **Nothing below the rig's own floor.** The interface, the cables and the converters have noise and distortion of their own. Below that level the app is measuring the rig, and it says so rather than quoting a figure (measurement floor, below).

5. **Nothing about listening.** The chart says how much of each harmonic there is. It says nothing about what the distortion sounds like.

2.2 The test signal

The test tone is a single sine that glides smoothly from a low pitch to a high one — from 20 Hz to 10 000 Hz over about 10 seconds in the standard plan — the synchronized swept sine. One signal visits every note in the range, so one recording holds the answer for the whole neck. Figure 2 shows its first quarter second and how its pitch climbs over the whole sweep: the pitch rises by the same musical interval in each second, which is why it draws as a straight line on the pitch axis of the lower panel. The instantaneous frequency is the pitch the tone is at, at any given moment.

The speed of the glide is not free to choose. The trick that makes the whole measurement possible is this: when a pedal clips a gliding tone, the second harmonic it produces is itself a gliding tone, at twice the pitch — which is to say, it is a copy of the test tone that is running ahead of it by a fixed amount of time. The third harmonic is another copy, further ahead; the fourth further still. For those copies to line up exactly, the tone's climb has to be timed so that it completes a whole number of cycles before each copy's head start — the synchronization condition, met by nudging the requested length slightly (the standard plan's 10-second request becomes 9.943 s; its rate constant is 1.6 s). The nudge is what lets each harmonic be pulled out afterwards as its own clean copy. In the standard plan the second harmonic runs 1.109 s ahead of the note, the third 1.758 s and the ninth 3.516 s.

The first and last moments of the tone are faded in and out rather than switched, so nothing bangs. During those fades the pedal was driven by a tone that was still rising or already falling, and a reading there describes the fade, not the pedal. The range of notes the tone played at full strength — the excited band, from 20.63 Hz to 9937.7 Hz in the standard plan — is where every number in the chapter is meant.

2.3 How the answer is pulled apart

The recording that comes back from the pedal is a gliding tone with its harmonics riding on it, all overlapping in time. Separating them takes three steps.

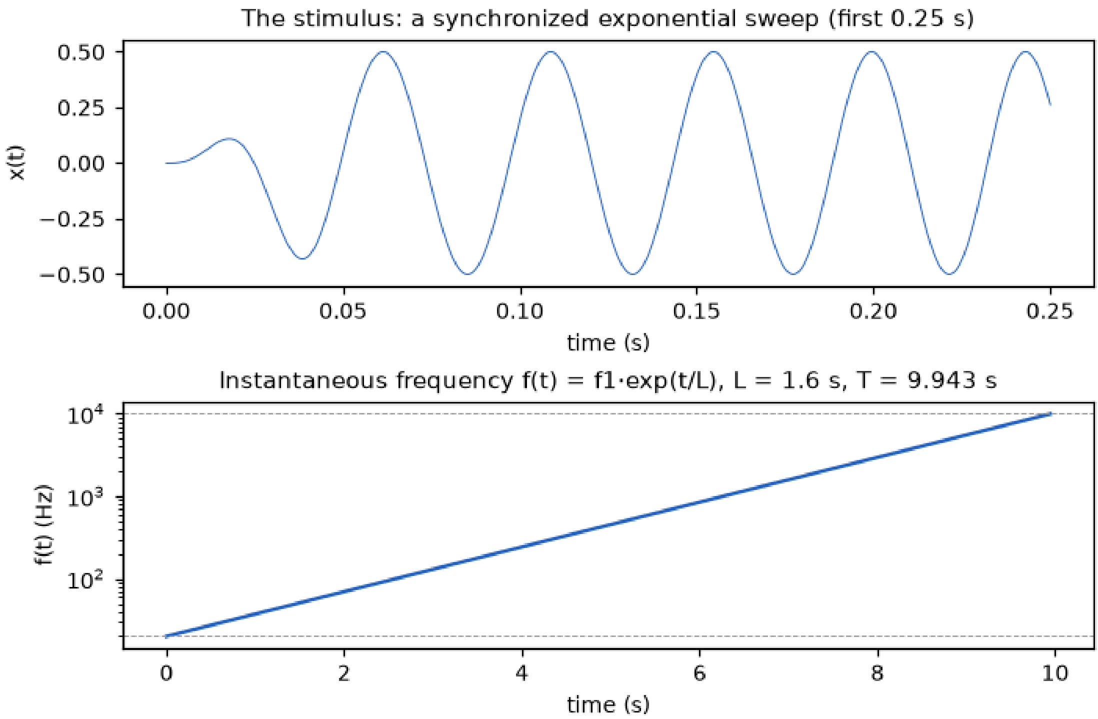

Figure 2: The test tone: its first quarter second (top) and its pitch over the whole sweep (bottom).

Undoing the glide. The first step is deconvolution: applying the exact mathematical undo of the test tone — its inverse filter, written down from the tone's formula rather than measured — to the recording. Undoing a glide turns it into a single sharp blip at one instant. Because every harmonic is a copy of the tone running a known amount ahead, each harmonic turns into its own blip, that same amount earlier in time. The result, the impulse response, is shown in Figure 3: the note itself is the blip at zero, and the harmonics stand in a row to its left, each at its own head start. Each one is a harmonic pulse. The example is a symmetric soft clipper, so only the odd harmonics have blips; at the even positions there is nothing, as there should be.

Cutting each blip out. The second step cuts each harmonic's blip out of that row with an extraction window: a slice of time centred on the blip, tapered smoothly to nothing at both ends so that the cut itself adds no edges. Each slice is as wide as it can be without reaching into the neighbouring blip, up to a cap of 0.2 s on either side — so the note and the first few harmonics get the full 0.4 s, and higher harmonics, whose blips crowd closer together, get less (the ninth gets 0.1785 s). Figure 4 shows the nine slices over their blips. A wider slice can tell finer pitch differences apart, so the high harmonics are read a little more coarsely than the low ones. If a sweep is so short that a slice would shrink below 16 samples on either side, that harmonic and every one above it is not reported at all: a short sweep simply carries fewer harmonics.

Reading a harmonic at a note. Each slice, once cut out, holds one harmonic's behaviour across every note at once. The app turns it back into a curve against frequency, and reads the harmonic for the note you care about at that harmonic's own frequency: the third harmonic of a note is read at three times the note's frequency, the ninth at nine times. (The curves are drawn against the note you play, but each point on the ninth-harmonic curve was read nine times higher up.)

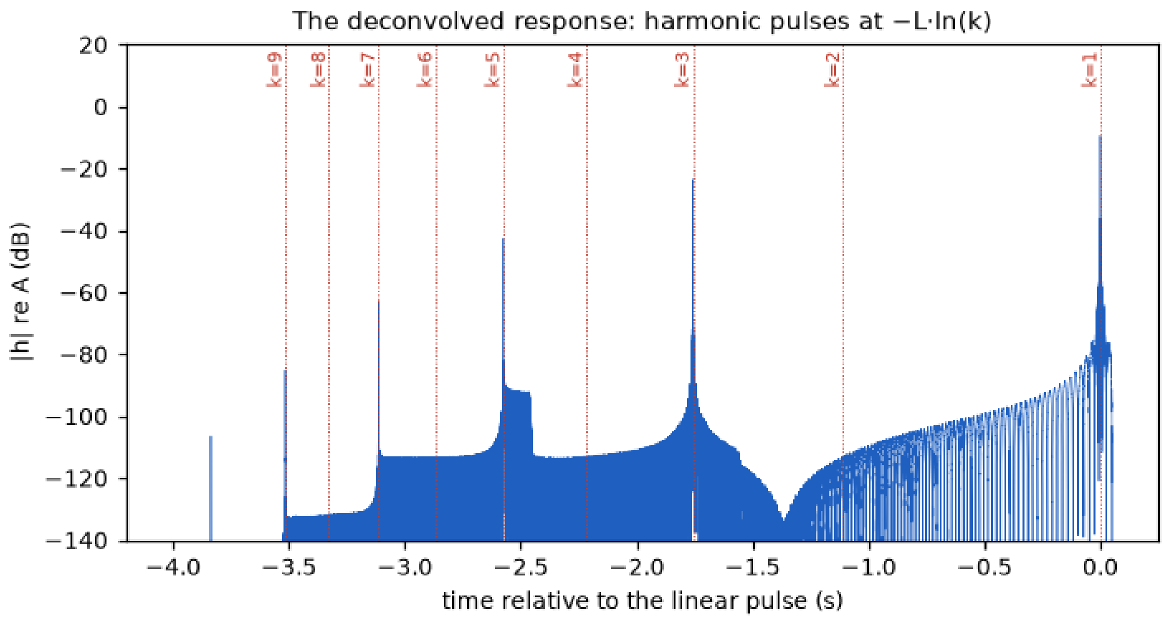

Figure 3: The recording after the glide is undone, for the worked example (a symmetric soft clipper). The note is the blip at time zero; each harmonic is a blip that much earlier. The even positions are empty because the clipper is symmetric.

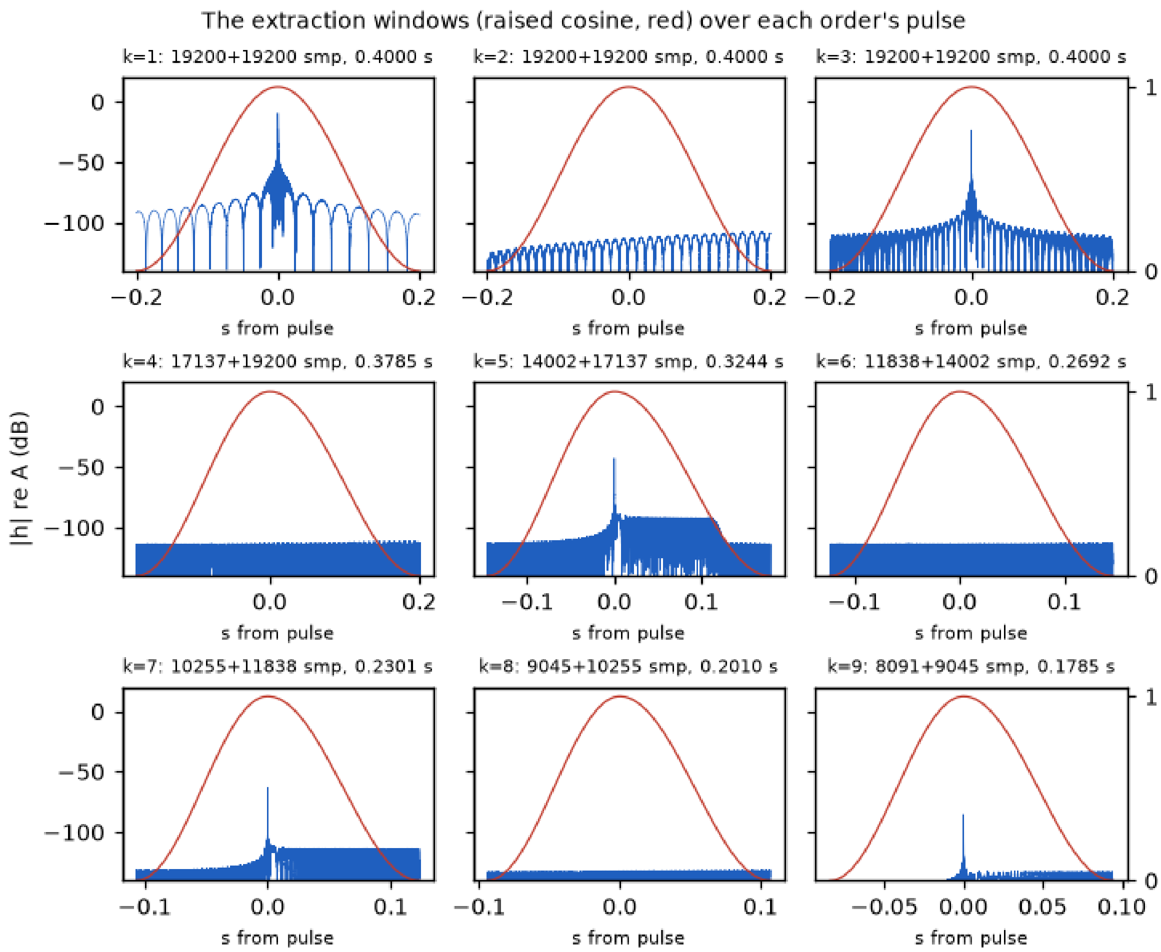

Figure 4: The nine slices (red) over their blips (blue). The first three take the full width; from the fourth on, the gap to the next blip sets the width.

Where a reading stops being honest. A digital recording cannot hold a tone above half its sample rate; anything higher folds back down to a lower frequency instead of vanishing (an alias image). For a high enough note, the tenth harmonic — which the pedal produces whether or not the app reads it — folds back onto exactly the frequency where the ninth is being read, and from that note upward the ninth harmonic's reading is contaminated. Every harmonic has such a note, and the higher the harmonic, the lower the note at which its next neighbour lands on it. That note is the harmonic's validity bound, and past it the app draws the curve faded and uses none of the values. In the standard plan at 96 000 Hz, the ninth harmonic is readable up to 5053 Hz (around D#8), the fifth up to 8727 Hz — all far above the top of a guitar's range, so for the notes you actually play every harmonic is inside its bound.

harmonic	arrives earlier by (s)	slice kept (s)	readable up to (Hz)	nearest note
1	0	0.4	10 000	D#9
2	1.109	0.4	10 000	D#9
3	1.758	0.4	10 000	D#9
4	2.218	0.3785	10 000	D#9
5	2.575	0.3244	8727.27	C#9
6	2.867	0.2692	7384.62	A#8
7	3.113	0.2301	6 400	G8
8	3.327	0.201	5647.06	F8
9	3.516	0.1785	5052.63	D#8

2.4 THD, in one sentence

The total harmonic distortion figure adds up everything the pedal added at multiples of the note — every harmonic from the second up that lies inside the audible band (20 Hz to 20 000 Hz) and inside its validity bound — and divides the total by the note itself, so that "ten percent" means the added harmonics together are a tenth the size of the note. It is a proportion of the note, not a verdict, and it is computed the same way at any sample rate because the audible band is fixed. It is THD and not THD+N: noise is deliberately not in the sum, which is why the checks in the next section matter — a harmonic slot that holds only hiss would otherwise count as distortion. Toward the top of the neck the sum loses its harmonics one at a time as each leaves the audible band or its validity bound; from the first note where that happens the app draws the curve broken, because it is then counting fewer harmonics, not measuring less distortion. The curve is sampled at 127 notes from 30 Hz upward (the very lowest notes are fade-in territory and are skipped).

2.5 Three checks the app makes before it believes a number

The floor. Every rig has a measurement floor: a level below which what the app reads is the interface's own noise and distortion, not the pedal's. The app measures it during calibration, with a cable in place of the pedal, and reads it against every harmonic at every note. A reading that sits at or below the floor is reported as a bound — "at most this much" — never as a figure, because nothing quieter than the rig can be seen through the rig. The floor is not one number: it depends on the note, on how loud the sweep was played, and on how many sweeps were averaged (playing the sweep 4 times back to back and combining the results, the app's default, lowers the noise part of the floor by about 6 dB; eight sweeps by about 9 dB). That lowering applies only where what the rig contributes is noise; at the faded ends of the sweep, where the residue is the measurement's own, averaging buys nothing and the floor stays put.

Fundamental presence. The THD figure, and everything else that compares a harmonic with the note, divides by the note. A few devices — an octave-up fuzz turned all the way up, a ring modulator — remove so much of the note that what comes back at the note's frequency is no longer the pedal but the analysis machinery's own leakage. The app has measured how far below the loudest harmonic a reading of the note can sit before that happens — 76 dB, established on devices whose answers were known in advance — and where a record crosses that line it stops quoting anything that divides by the note (fundamental presence). The harmonic levels themselves are still real and are still shown; the ratio to a note that is not there is not.

Shaping: circuit or hiss? The analysis reads whatever landed in each harmonic's slice, and the pedal's own hiss lands there too. For a symmetric clipper the even harmonics are exactly absent, so what the app reads at the even positions *is* its hiss — often well above the rig's floor, and still not distortion. The harmonic shaping check separates the two by looking at how a harmonic's reading behaves across neighbouring frequencies. Something a circuit did varies smoothly and consistently from one frequency to the next; noise does not — its readings scatter, each one unrelated to its neighbours. The app computes a coherence score for that orderliness over 32 neighbouring readings (pure noise scores about 0.18 on it, a perfectly orderly response scores 1), converts the score into how far the reading stands above the noise in it, and calls the reading *shaped* only when that margin is at least 6 dB. With the score comes a read scatter: how far a single reading might be off purely because of the noise in it, stated in dB alongside the level (a reading that is pure noise carries about 5.6 dB of it). A reading that fails the check is drawn dotted and faded, the legend says "at the floor — noise, not a response", and nothing downstream — the THD sum, the comparison distance, the summaries — treats it as a harmonic.

Averaging, since it came up. When the sweep is played more than once, each repeat is analysed on its own and then the repeats are combined. Before combining, each repeat is lined up with the first to a fraction of a sample (derotation) — a timing drift of even a fraction of a sample between repeats would otherwise blur the high harmonics into looking politer than they are. The combination is of the full readings, direction included, never of the levels alone: averaging levels would make a biased number more precise, not more right. This is sweep averaging; the app offers 1, 2, 4, 8 sweeps and defaults to 4, with a 0.5 s gap between repeats.

2.6 What the result looks like

Figure 5 is the worked example — a symmetric soft clipper whose exact answer is known — as the pipeline reads it. Each solid curve is one harmonic's measured level across the notes, in dB relative to the input; the dashed line under each is the exact answer, which the solid curve sits on. Where a curve turns dotted it has passed its validity bound, and the shaded strip at the far right is the fade-out, where the last point reads the fade rather than the device. The even harmonics are not drawn at all: for a symmetric clipper they are exactly absent, and the pipeline reads nothing there but the arithmetic's own dust, far below the chart. The note itself reads above zero because this clipper also adds gain.

The two charts the app draws for a real record are shown next, rendered from the app's own chart code on its simulated pedal (never a screenshot). The first is the Harmonic Distortion chart itself. Each curve is one harmonic; the legend names it and says what it is musically — the second harmonic is an octave up, the third an octave and a fifth, and so on — with the odd family in one colour and the even family in another, so the symmetry rule from the first section can be read straight off the chart. The horizontal axis is the note you play, labelled with note names and their frequencies; the vertical axis is the harmonic's level in dB relative to the input, and its label says how the axis was scaled. The shaded band marks a guitar's range. Where a curve fades and breaks near the top it has crossed its validity bound. The lines under the chart state

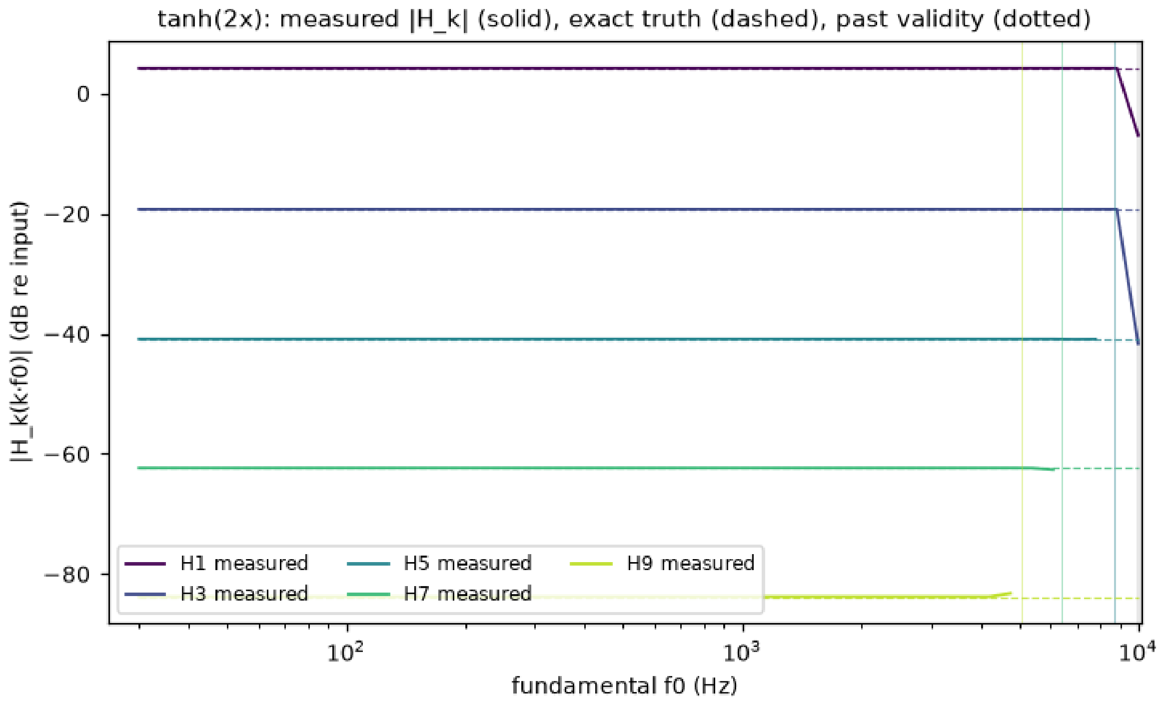

Figure 5: The worked example: measured level of each odd harmonic (solid) against the exact answer (dashed) across every note. Dotted segments lie past the harmonic's validity bound; the shaded strip is the fade-out.

the drive level the sweep was played at and, for a real capture, how many sweeps were averaged — read them before comparing two records, because a harmonic level only means something at a stated loudness.

The second chart is Distortion vs. Note: the THD figure, in percent, across the same notes. A flat curve means the pedal adds the same proportion of harmonics at every note; a curve that falls toward the top of the neck usually means the pedal's own tone filter is shaving the harmonics off. Where the curve turns dashed, the sum has begun losing harmonics to the band edge, one at a time, and the line under the chart names the note at which each one left — from there the curve falls because it counts fewer harmonics, not because the pedal cleaned up. The vertical axis is fitted to the notes before that point, so the dashed tail may run off the bottom of the chart; that is the honest picture of a sum with fewer terms in it.

2.7 How you can check that the app does what this says

Everything above is a description, and a description can drift from the code it describes. The pipeline behind these documents guards against that in a way anyone can rerun. The app's own measurement code writes out the methods manifest: every setting that determines a result, obtained by calling the code that owns it. From that manifest and the published method, a second program was written — in Python, independently, not by translating the app's code — and the parity test runs both on the same made-up devices and demands the same answer.

The made-up devices are the point. Three of them are formulas — the soft clipper of the worked example, a hard clipper and a gentle polynomial curve — whose exact harmonic content can be computed on paper, so both programs are held not only to each other but to the truth. The fourth is a phantom capture: a recording built to contain exactly a chosen set of harmonics and nothing else, so the analysis is asked to read back precisely what was put in.

The demands, in numbers the pipeline writes into this document: the two programs must agree to

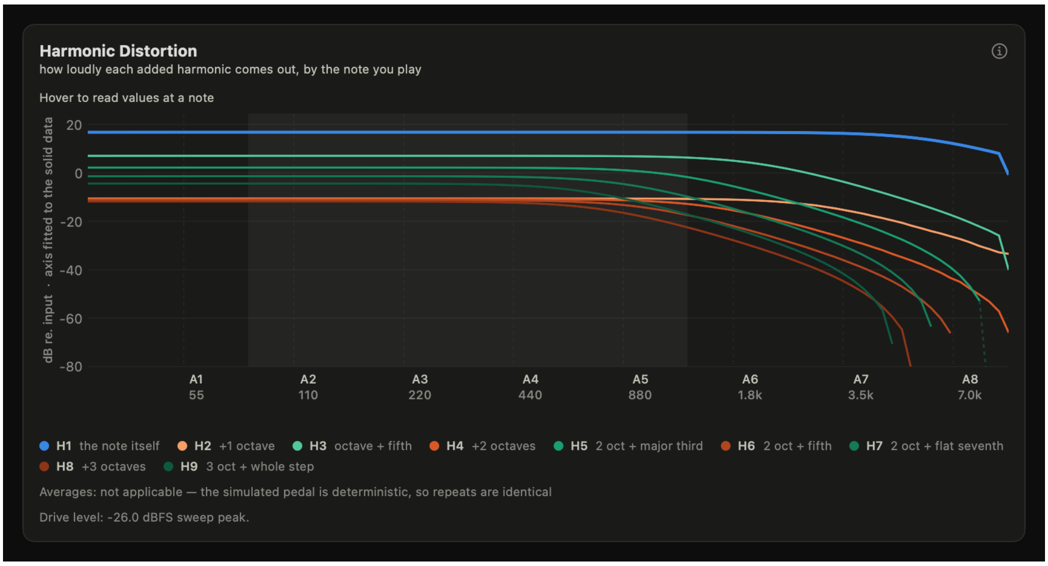

Figure 6: The Harmonic Distortion chart as the app draws it, for its simulated pedal.

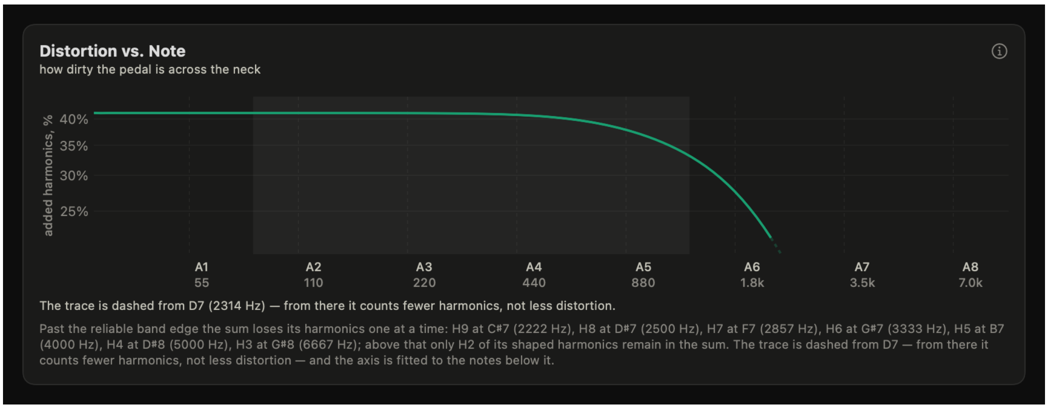

Figure 7: The Distortion vs. Note chart as the app draws it, for the same simulated pedal.

within a hundred-millionth of a decibel (1e-08 dB) at every harmonic and every note (on the run that built this document the worst disagreement was about two hundred-billionths of a decibel, 2.1e-11 dB — the two differ only in the last digits of floating-point arithmetic); on the phantom, both must be within 0.01 dB of the exact answer over the whole range; on the formula devices, within stated tolerances (tanh 0.01 dB, poly 0.01 dB, hardclip 0.3 dB) over the lower half of each harmonic's range — the upper half is where a computed device, unlike a real pedal through a real interface, folds harmonics from above the sample rate back into the band, and that folding is the device's, not the method's. For the worked example the exact answer is 4.2083 dB for the note itself, -19.292 dB for the third harmonic and -83.757 dB for the ninth, and both programs read them back within those tolerances. A fifth case adds noise and averages four sweeps, and requires every reading the shaping check calls shaped to sit within 3 times its own stated scatter of the truth — the check on the check.

The test runs automatically on every change to the app, and the tables and figures in both documents are regenerated on every change and compared with the committed copies: a method change that the description does not follow is a failing build, not a stale page. The scripts, the test and the instructions for running them are in the repository named on the first page.

2.8 Where it stops being a measurement

- **The faded ends.** The very first and very last moments of the sweep are ramps, and a reading there is a reading of the ramp. Numbers from those notes describe the measurement, not the pedal; the app excludes them where it can and draws them faded where it cannot.
- **A loud harmonic leaks a little into its neighbours' slices.** How much is measured, not assumed, and the app never claims a harmonic deeper below the loudest one than that leakage allows. Where the note itself is that far down, the fundamental-presence check withdraws every ratio to it.
- **A number can be noise.** A harmonic slice that holds only hiss still yields a number. The shaping check is there to say so, and a reading it rejects must not be read as a harmonic, however confidently it is drawn.
- **The simulated pedal folds its own harmonics.** The app's virtual pedal applies its clipping to the signal sample by sample, so the harmonics it makes above half the sample rate fold back into the band — an error a real pedal through a real interface does not carry. The user guide's Simulation Mode page states it; the parity numbers above measure it, and it is largest in each harmonic's top half-octave.
- **Short sweeps carry fewer harmonics.** Each harmonic needs a slice at least 16 samples wide on each side of its blip; a sweep much shorter than the standard plan runs out of room for the high ones, and they are simply not reported.
- **The range's top point.** The THD curve asks for 128 notes and can deliver 127: its topmost point lands a hair above the sweep's end and is dropped rather than read under the fade.

3 Transfer Curve

3.1 What this measurement tells you, and what it cannot

The Harmonic Distortion chapter inferred a pedal's clipping from the harmonics it adds. This measurement draws the clipping itself. The app plays one steady low note into the pedal, records what comes back, and plots the output against the input, moment by moment, over one cycle of the note: the transfer curve. A plain wire draws a straight diagonal line — what comes out equals what went in. A clipper draws a line that bends over at the top and the bottom, and the bends are its clipping shape, seen directly rather than deduced.

Reading a picture of output against input. Each instant of the recording is one dot: how far across is the input at that instant, how far up is the output. As the note swings up and down, the dot travels out to the right, back through the middle, out to the left and back again, and over a whole cycle it traces a path. If the pedal's output depends only on the input at that same instant — the same input always gives the same output, whether the wave is on its way up or on its way down — the outward and return journeys land on the same path, and the picture is a single line. Figure 8 shows the worked example, a soft clipper with no filter, both ways: the path as the app sees it on the left, and the same input and output cycles laid out against time on the right. The path retraces itself exactly, so the line is the device's clipping formula, and the exact formula drawn over it cannot be told apart from it.

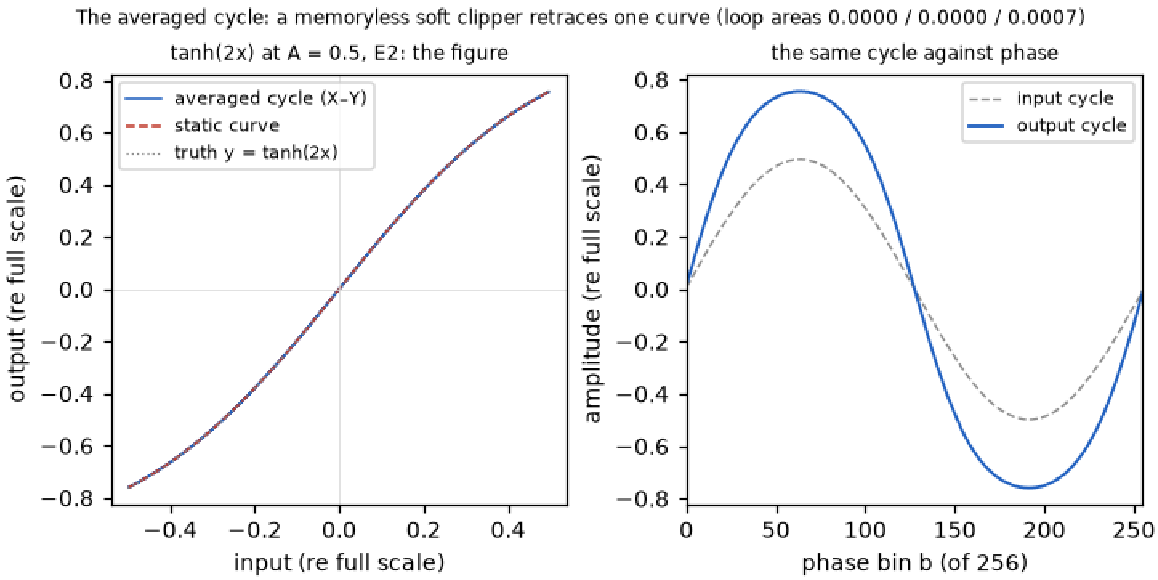

Figure 8: The worked example, a soft clipper with no filter, at the loudest hardware drive and the standard note. Left: the averaged cycle as the app plots it, output against input, with the app's static curve (dashed) and the exact formula (dotted) over it — the three coincide. Right: the same input and output cycles against position in the cycle. The title gives the three loop areas, all zero to the resolution of the slots.

Why a time shift draws a loop. Now suppose the output is an undistorted copy of the input that arrives a little late. At any moment the output is what the input was a moment ago, so the same input value now meets two different outputs — one on the way up, another on the way down — and the outward and return journeys separate: the path opens into an oval. Figure 9 builds one. On the left, a sine and a shifted copy of itself; in the middle, the oval the two draw when one is plotted against the other; on the right, what the worked example's clipper followed by the same shift draws — the bends are the clipper, the opening is the shift. How fat the oval is depends on the size of the shift as a fraction of the cycle: no shift is a line, a quarter of a cycle is a circle. A shift like this is what every tone filter does, because a filter passes each frequency with its own small delay, so a pedal with any tone shaping in it draws an oval even when it adds no distortion at all.

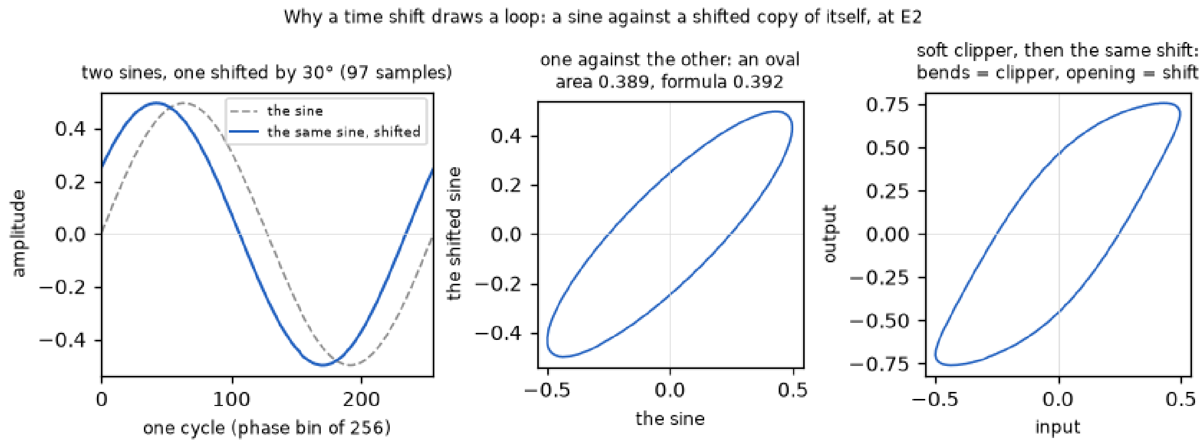

Figure 9: Left: a sine and the same sine shifted in time by a twelfth of a cycle. Middle: the oval the pair draws when one is plotted against the other, with its area as a fraction of the box and the value the formula for a pure shift predicts. Right: the worked example's soft clipper followed by the same shift — the bends are the clipper and the opening is the shift.

The word "memory". There is a second way to open the loop, and it is the one this measurement is really after. If the circuit's output depends not only on the input now but on where the signal has just been — a bias point that drifts with the signal, charge stored in a capacitor that leaks back out over the next few cycles — then the same input value again meets different outputs on the way up and on the way down, and the path opens. That is memory, and its loop looks like the filter's oval. Telling the two apart is the whole difficulty of this chapter, and the phase removal below is the app's answer.

The measurement's result is the averaged cycle itself, the line through the middle of it, and three loop areas, each stated as a fraction of the picture's box: the total, the part a tone filter's shift accounts for, and the part left over.

What it cannot tell you. Stated once.

1. **Only one loudness.** One note at one drive is one operating point. A pedal whose clipping starts above that drive draws a straight line — not clean, merely not driven into its bends. The Gain Map and the Compression measurements visit other loudnesses.

2. **Whether an open loop is memory or a tone filter.** Ordinary tone filters open the loop with no memory in the circuit. The phase removal exists to take the filter's part out, and even then the middle band of loop sizes is reported with a hedge, for a reason the next section gives.

3. **The clip levels of a lopsided clipper.** Every pedal has a capacitor on its output that removes the wave's average level. It centres a clipper that clips the top harder than the bottom, so the two clip levels come out equal in the picture even when the circuit's are not; the lopsided shape survives as even harmonics in the Harmonic Distortion measurement instead (the limits section).

4. **The rig's own timing error.** If the recording is lined up with the test tone a fraction of a sample late, the picture opens into an oval the measurement cannot tell from the device's own — and it is a property of the rig, not of the pedal (the limits section).

5. **Nothing about listening.** The picture shows the clipping shape. It says nothing about what the distortion sounds like.

3.2 The test signal

The test tone is a single sine at one note, held steady. The standard note is E2 (82.4 Hz, the low E string), and the standard loudness is -26 dBFS, the same drive the Harmonic Distortion sweep uses, so that the two measurements describe the pedal at one operating point. The tone lasts 1.5 s — about 124 cycles of the note — with 0.2 s of silence before it and 0.2 s after, and its first and last 20 ms are faded rather than switched, so nothing bangs. The first 8 cycles (about 0.097 s) are left out of the analysis while the circuit settles into the tone; everything from there to the end of the tone is used. The app measures the delay through the rig on every capture and lines the recording up with the tone before anything below happens.

The app offers 10 probe notes (E2, A2, E3, A3, E4, A4, E5, A5, A6, C7) and a note-range capture at E2, A3, A4; the drive can be set from -60 to -6 dBFS on hardware (a plugin can be driven to 0 dBFS, digital full scale). The worked example throughout this chapter uses the loudest hardware drive, -6 dBFS — an amplitude of 0.5 of full scale — because at the standard drive the reference clippers the method is checked against barely bend at all. That is the point of the level control: this measurement shows the shape only where the note reaches it.

3.3 How the picture is built

Seven steps, each in words; the technical reference gives each one as an equation.

Folding the cycles together. The recording holds about 115 usable cycles of the note. Each cycle is divided into 256 equal slots by position within the cycle, every sample goes into its slot, and each slot's samples are averaged — the input's and the output's separately. That gives one averaged cycle: 256 input values and 256 output values, one pair per slot. Noise, which is different in every cycle, averages toward nothing; the shape, which repeats, stays. The right-hand panel of Figure 8 is that cycle, and the left-hand panel is the same 256 pairs plotted against each other. Every sample from the skipped opening to the end of the tone is folded in, the closing fade included — the limits section returns to that.

The box. Loop areas need a scale, or a loud signal's loop would read larger than a quiet one's for no reason. Every area is divided by the area of the rectangle the whole picture fits inside, the box normalization, so that an area is a fraction: a loop filling the whole box would read one, and a line reads zero.

Counting the area lobe by lobe. A loop can cross itself. A figure-eight is two lobes traced in opposite directions, and the textbook way of measuring an enclosed area, which keeps track of direction, would let the two cancel and report nothing. The app instead cuts the path at every crossing and adds up the size of every lobe regardless of which way it was traced: the unsigned lobe area. A lobe smaller than 0.0002 of its own box is dropped as the jitter of the averaging.

The oval a tone filter draws. The best-fitting oval is drawn through the loop using only the note itself — the fundamental of each of the two cycles, nothing above it. That oval is the fundamental ellipse, and its area is exactly what a shift of the note in time would draw: the linear part of the loop, a filter's doing, never memory. Figure 10 shows the worked example's clipper behind a tone filter (a high-pass just below the guitar's range): the raw loop, the oval fitted to it, and what remains.

What is left. The oval's output is subtracted from the output cycle, and the loop that remains is measured the same way, lobe by lobe: the nonlinear loop area, the headline memory figure. It is the harmonics' own loop. In a clipper with no memory and no filter the harmonics keep step with the note and retrace; a harmonic that has been shifted — by a filter, or by memory — encloses area here.

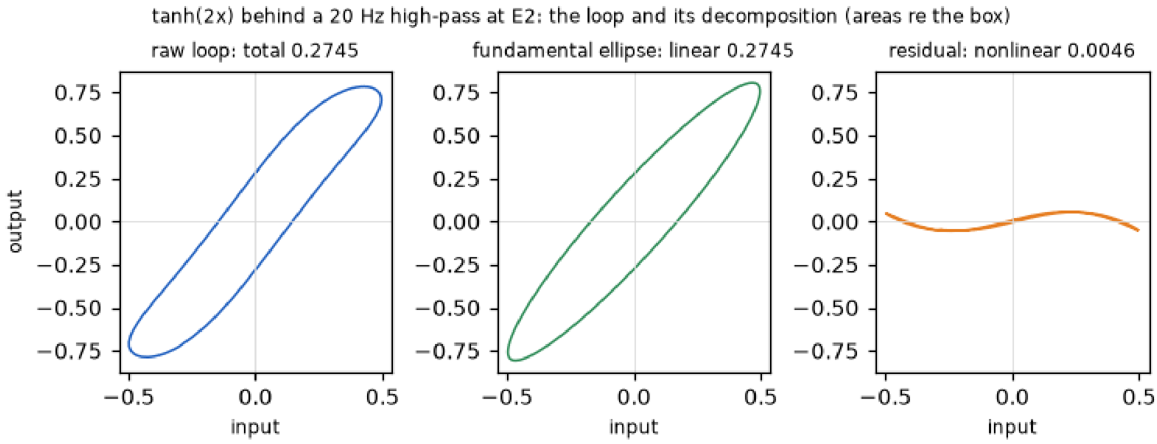

Figure 10: The worked example's soft clipper behind a 20 Hz high-pass at the standard note: the raw loop (left), the oval fitted to its fundamental (middle), and what is left once the oval's output is subtracted (right), each titled with its area as a fraction of the box. The filter draws the oval; what is left is small.

The line through the middle. The static curve is the app's estimate of the clipping shape with the loop ignored. The cycle is cut at the input's lowest and highest points into a rising pass and a falling pass; each pass is resampled onto the same 101 evenly spaced input values, and the two are averaged. It is done pass by pass, rather than by sorting every sample by its input value, because a sine spends most of its time near its peaks and hardly any near zero — sorting by input value would starve the middle of the curve.

The memory verdict. The memory verdict is the app's three-way call on the nonlinear loop area (of the corrected picture, when the phase removal has been applied): under 5 percent of the box the device reads memoryless-ish, from 15 percent it reads memory, and between the two the reading is hedged, deliberately. The reason is a tone filter placed *ahead* of the clipper. Such a filter shifts the note before it is clipped, and the harmonics the clipper then makes are shifted along with it — the second harmonic by twice the note's shift, the third by three times — so the harmonics no longer keep step with the note, and they enclose area with no memory in the circuit at all. From a single note the app cannot tell that from mild memory, and it says so instead of choosing.

Every one of these quantities is recomputed from the stored cycle each time a record is read; the record keeps the cycle and nothing that a later fix would have to migrate.

3.4 Three things the app checks before it believes the picture

All three lean on the pedal's paired Harmonic Distortion record — the sweep measurement of the previous chapter, of the same pedal at the same setting — when one exists. Nothing they decide is stored; each is re-decided from the two records every time.

The right way up. A plugin is measured inside the computer, and its recording is lined up with the test tone by finding the best match — which, for a steady tone, can land exactly half a cycle off. The picture then comes out upside down, and for a lopsided clipper worse than that: each input is paired with the output from half a cycle later. The orientation resolution check reads where the note sits in the captured cycle. When it sits within 2.9 degrees of exactly in step or exactly reversed, the alignment is known to have locked to the tone; if the pedal's polarity (next) then disagrees with the way the picture is drawn, the output cycle is moved on by half a period and the picture redrawn. In every other case the picture is shown as captured and nothing is implied. A hardware capture is lined up by a different method that keeps the pedal's true timing, and is immune.

Polarity. polarity is whether the pedal turns the wave upside down. It is read from the paired record, over the low notes: does the note come back the same way up or inverted, with the louder readings counting for more. The confidence is how consistently the answer comes out, on a scale where one is perfect agreement, and the reading counts as confident only above 0.35. A device that shifts the note by a quarter of a cycle is neither one nor the other; it reads as not confident rather than as a guess.

Is the note there? The memory verdict, the polarity statement and the pairing check below all depend on the note itself being in the recording. A few devices remove it (an octave-up effect, a ring modulator), and what is then read at the note's frequency is the analysis's own leakage. The app takes the fundamental presence reading from the paired record, which separates the note from its harmonics far better than one averaged cycle can (the 76 dB bound of the previous chapter), and where the note is absent it draws the raw picture and states none of the three.

The phase removal. This is the longest part of the chapter, and the part a reader most needs.

The problem. A tone filter shifts each frequency in time by its own amount. A filter *after* the clipper shifts each harmonic by the filter's own delay at that harmonic's frequency. A filter *before* the clipper shifts the note, and the clipper's harmonics inherit multiples of that shift — two, three, four times it. The two cases look alike in the picture and are corrected differently, and the obvious repair (correct each harmonic by the filter's delay at that harmonic's frequency) is wrong exactly when the filter sits before the clipper — which is where a pedal's tone control usually sits.

What the paired record knows. The sweep measurement read each harmonic at its own frequency as the pedal made it, filters before and after included. So the paired record already holds every harmonic's true timing, whichever side of the clipper the filters sit on. The phase removal rests on that: the picture's harmonic timings are compared with the paired record's, and the difference is what the filters did to the picture.

Lining the two recordings up. The two measurements were made separately and each measured its own delay through the rig, so a small time offset between them is legitimate, and it has to be found before anything else — treated as a filter shift, it would be "removed" wrongly. The frame fit finds the one offset that makes every harmonic agree at once, louder harmonics counting for more, by scanning every offset within half a cycle. Whatever disagreement remains after that, over the harmonics above the note, is the pairing residual: the test of whether the two records really describe one device at one setting. Above 0.25 radians (14 degrees of a cycle) the app declines.

Who votes. A harmonic takes part only when both records hold it clearly — at least -46 dB relative to the note on each side — and it is inside its validity bound in the paired record. Fewer than 3 voting harmonics is too little evidence, and the app says so rather than fitting through what it has.

The two allowed positions, and the fold. In a pedal with no memory and no filtering, every harmonic sits in one of two positions relative to the note: exactly in step with it, or exactly reversed. Which of the two is the device's own doing — a clipper's third harmonic is reversed — and the picture keeps it. That pair of positions is the memoryless lattice, and a measured harmonic's distance from the nearer of its two positions is what the filters added. But "nearer" is ambiguous once a filter has shifted a harmonic more than a quarter of the way round, so the app folds in order. The note itself goes first, anchored by the polarity: inverting means reversed; without a confident polarity the nearest position is taken, and the sign is then quoted only within a stated confidence. Each higher harmonic is folded against a prediction made from the ones already folded — a straight ramp with the harmonic number, which is the shape a filter before the clipper produces, plus the paired record's own low-frequency shape, which is the filter after it. The working assumption is that what a harmonic carries beyond that ramp is under a quarter turn: true of the resistor-and-capacitor networks pedals are built from, and weakest at the lowest harmonic, where it has held on every device class measured.

The correction. Once each harmonic's added shift is known, the whole picture is re-timed: a smooth correction across every frequency in the cycle that moves each one in time and makes none louder or quieter. It is measured up to the highest harmonic that voted — the support, that harmonic's number times the note — and extended smoothly above it, and the app's caption states the support and that the correction is extrapolated above it. Figure 11 is the case the obvious repair cannot handle: the soft clipper behind a filter placed before it. The raw picture is a wide loop with the clipping bends on it; the corrected picture is the clipper's own curve, laid over the exact memoryless answer.

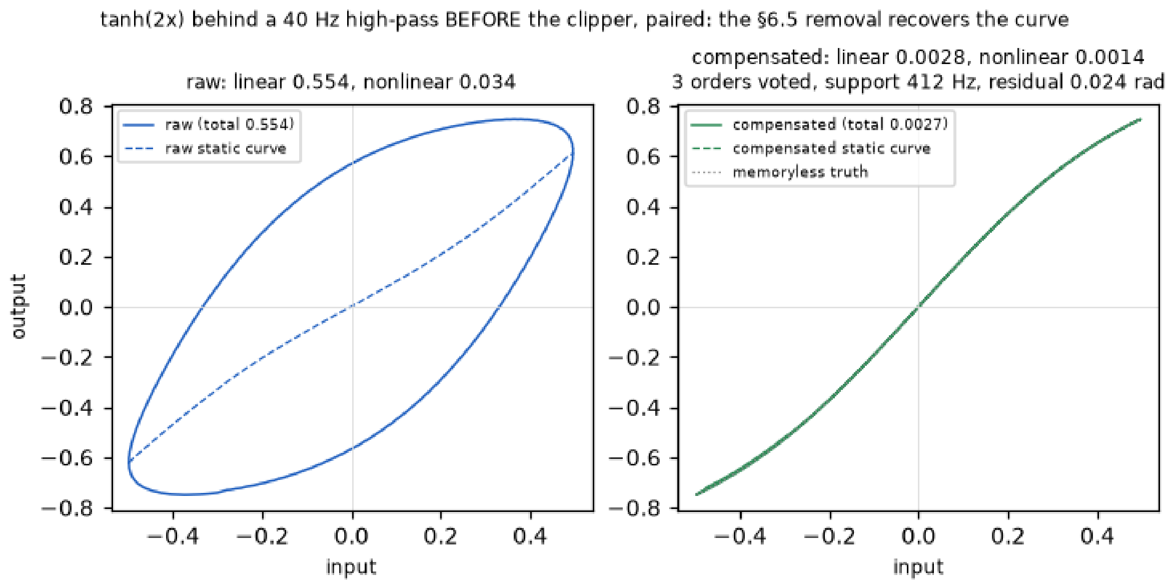

Figure 11: The soft clipper behind a 40 Hz high-pass placed before the clipper, paired with its own sweep measurement. Left: the raw loop and its static curve. Right: the corrected picture over the exact memoryless curve, with how many harmonics voted, the support and the residual.

Five outcomes. Corrected; a pairing mismatch, when no offset reconciles the harmonics; too little evidence, when fewer than 3 harmonics vote; a negligible loop, when the raw total is under 1 percent of the box — there is nothing to correct, and this outranks a mismatch, so a tiny noise-dominated loop never wears a warning (the fit still runs, so polarity can still be stated); and a tripped guard, when a correction that passed the residual nonetheless grew one of the loop areas by more than 0.02 of the box over the raw picture, in which case the raw picture stands. A pairing mismatch has two possible causes the residual cannot separate: the two records come from different settings or sessions, which measuring both again together resolves; or the device is one whose harmonics have no fixed timing relationship to the note (a frequency doubler), which no re-measurement resolves. The app names the class, not the cause.

Polarity, stated. The picture states the pedal's polarity only when the fit ran, its residual confirmed the pairing, and the note's position resolved; otherwise the tile stays empty. Absence is honest; a guessed sign is not.

3.5 The settings

Every setting is in the technical reference's Transfer Curve chapter, in a table read from the manifest; this chapter repeats only the handful a reader acts on. The standard note is E2 at 82.4 Hz and the standard loudness -26.02 dBFS (an amplitude of 0.05 of full scale); the tone lasts 1.5 s at 96 000 Hz, so one cycle is about 1 165 samples; the first 8 cycles are skipped; the cycle is folded into 256 slots and the static curve is drawn on 101 input values; the memory verdict's two thresholds are 0.05 and 0.15 of the box; and the phase removal declines above a residual of 0.25 radians, needs 3 voting harmonics, and does nothing at all when the raw loop is under 0.01 of the box.

3.6 What the result looks like

Figure 8 is the pipeline's own rendering of the worked example. On the left, the averaged cycle plotted output against input, with the static curve and the exact formula over it; the three coincide, which is what a memoryless device with no filter looks like here. On the right, the same input and output cycles side by side against position in the cycle: the output is the input with its peaks pushed in and rounded — the clipping — and taller, because this clipper also adds gain. The title gives the three loop areas, all zero to the resolution of the slots.

The app's own chart, rendered from its chart code on its simulated pedal (never a screenshot), is Figure 12. Four tiles run across the top: the drive note and its frequency; the memory figure — the nonlinear loop area, with the verdict's word under it; the linear phase loop — the oval's area, labelled as filter phase and not memory; and the polarity, with a plain-words gloss. Under the chart's title, the lines state whether the phase removal was applied, through how many of the measured harmonics it was fitted, up to what frequency the correction is measured (extrapolated above), which record it was removed against, and the alignment residual in radians. The chart itself draws the loop in blue and the static curve dashed in orange, labelled as both directions averaged; what goes in runs across, what comes out runs up. For the simulated pedal the loop is closed and the two lines coincide. On a real pedal with an open loop, the dashed line is the app's estimate of the clipping shape through the middle of it, and the limits section says where that estimate is weakest.

3.7 How you can check that the app does what this says

As in the previous chapter, a second program was written from the description — in Python, independently, not by translating the app's code — and the parity test runs both on the same made-up devices and demands the same answer. Here the made-up devices come in three kinds. Memoryless clippers with no filter, whose exact curve is their formula, so the picture can be

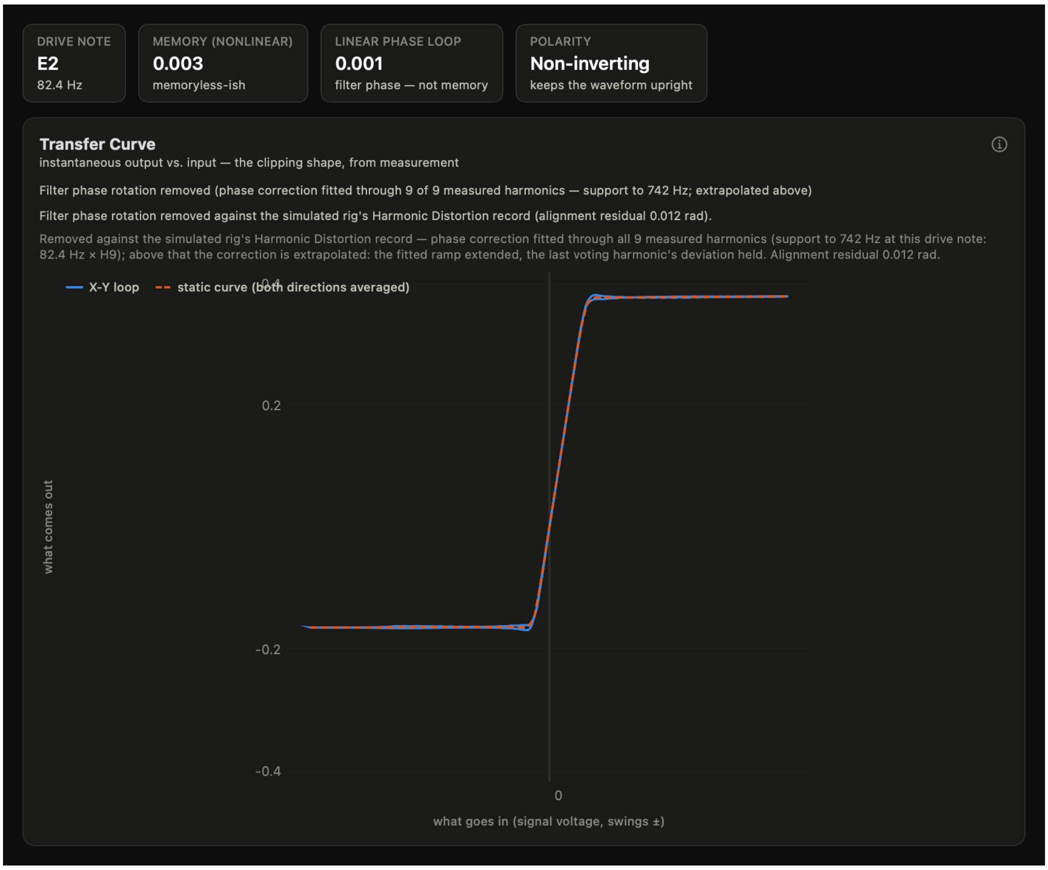

Figure 12: The Transfer Curve view as the app draws it, for its simulated pedal, with the phase removal applied.

checked against the truth point by point. Linear devices — a wire behind a tone filter, and a wire read with a deliberately planted timing error — whose oval must match the formula for a pure shift. And filtered clippers, with the filter after the clipper or before it, measured by the sweep of the previous chapter as well so that the phase removal can be run and its answer checked against the memoryless truth. That last kind is the one that matters: the removal is held not only to the app but to what the clipper would have drawn with no filter at all. In total the test runs 21 cases.

The demands, in numbers the pipeline writes into this document. The two programs must agree on every value in the cycle, on the static curve, on the corrected picture, on every area and on every fitted quantity to within a ten-billionth (1e-10; on the run that built this document the worst disagreement was about seven ten-trillionths, 7.4e-13, the last digits of floating-point arithmetic), and must find the same time offset to within a billionth of a sample (the measured difference was 0); every yes-or-no reading — the outcome class, the flip, the polarity, the verdict, the presence — must be identical. The oval drawn for a linear device must match the formula for a pure shift to within a hundredth at the plan's tone length (measured: at worst 0.73 percent short) and to within a thousandth when the tone is ten times longer (measured 0.05 percent) — the limits section says why the length matters. On a smooth clipper the raw static curve must sit within a hundredth of the output peak of the exact curve (measured three thousandths, 0.003); on a hard clipper within five hundredths (measured about three hundredths, 0.027, because 256 slots round a sharp corner); and a memoryless device's nonlinear loop area must stay under five thousandths of the box (measured two thousandths, 0.002). Every paired filtered device must come out corrected on both sides, with the planted time offset found to within a half of a sample (measured about four tenths, 0.39), a residual under 0.05 radians (measured 0.0237; the app's own accept bar is 0.25), the memoryless curve recovered to within a hundredth of peak (measured about six thousandths, 0.0055), and both corrected loop areas under five thousandths of the box (measured about three thousandths, 0.0028). The removal's parity is established: every paired filtered case compensates on both sides and the two implementations agree on the compensated figure to the Swift-vs-Python tolerance.

Two of those results are worth stating in words. The clipper behind a filter placed before it has no memory at all, and its raw nonlinear loop area reads 0.0343 — inside the memory verdict's hedged band; the removal takes it to 0.0014. The hedge in the pipeline section is exactly this case. And the half-cycle lock is undone: on the case built to land half a cycle off, the note's captured position read 3.144 radians, a half turn to within about three thousandths of a radian, the paired polarity read non-inverting, and the resolved picture rises from left to right.

The test runs automatically on every change to the app, and the tables and figures in both documents are regenerated on every change and compared with the committed copies; the scripts and the instructions for running them are in the repository named on the first page.

3.8 Where it stops being a measurement

- **The oval's formula is for a linear device only.** The rule that turns a shift into an oval's area assumes the device's peak is the note's own. A clipped output's peak is not, so on a clipper the oval's area can read larger than that rule gives. A reader who wants the shift as an angle should apply the rule to a linear-device measurement only, and the pipeline's own check reads the rule to 0.73 percent at the plan's tone length and 0.05 percent at ten times it — so a deficit of several percent on a hardware capture is in the measurement chain, not in the rule.

- **The closing fade is folded in.** The skipped opening cycles protect the start of the tone; the last 20 ms ride their fade into the averaged cycle. Their share of the average shrinks with the tone's length, which is the whole of the deficit above: it is why the pipeline checks the oval at ten times the tone length as well, where the share is gone.

- **A leftover timing error is a loop.** Any timing error left between the tone and the recording — the app removes a whole number of samples, and the fraction of a sample at least stays — shifts the note by an amount that grows with the note's frequency, and draws an oval on a plain wire (Figure 13). On a pedal that residual alignment error loop cannot be told from the pedal's own filter loop. A reader can test it: measure two notes on one rig, and a single timing error must account for both ovals. The nonlinear loop area is untouched by it on a linear device.
- **The output capacitor hides unequal clip levels, not unequal harmonics.** A clipper that clamps at different heights carries an average level, and the coupling capacitor every pedal has on its output strips it, leaving equal heights from an unequal circuit (Figure 14). The shape asymmetry remains, and the even harmonics in the paired sweep measurement are where it is read. Because every pedal has one, a transfer curve essentially never shows the raw asymmetry of an asymmetric clipper.
- **The correction is measured only as high as the last voting harmonic.** A harmonic too quiet to clear the floor on either side does not vote — the soft clipper at this drive seats 3 harmonics, the asymmetric clipper 7 — and above the support the correction is extended, not measured. A hard-clipped device has content far above it.
- **The paired timing is read at the nearest point.** The paired record's timing for a harmonic is read at the nearest frequency it holds, so a read can be off by up to half a step of that grid; the frame fit absorbs it as a spurious offset of a fraction of a sample (measured up to 0.39 samples on the check cases) and leaves the rest in the higher harmonics' residuals.
- **The slots round a corner.** With 256 slots per cycle a hard clip's corner is averaged over one slot's width of input, and the static curve departs from the exact curve there by a few percent of the clip level; the smooth devices read under one percent of peak.
- **The line through an open loop is weakest at the ends.** The static curve averages the rising and falling passes, and the two meet at the input's extremes, which is where the estimate is least trustworthy; on a wide oval it departs from the true line by a few percent of peak there. The app draws it dashed, and that is the reason to read its ends with care.
- **A declined removal names its class, not its cause.** The residual separates a genuine pairing from a mismatch; it does not separate a session mismatch from a structural one, and the two need different remedies (the phase removal, above).

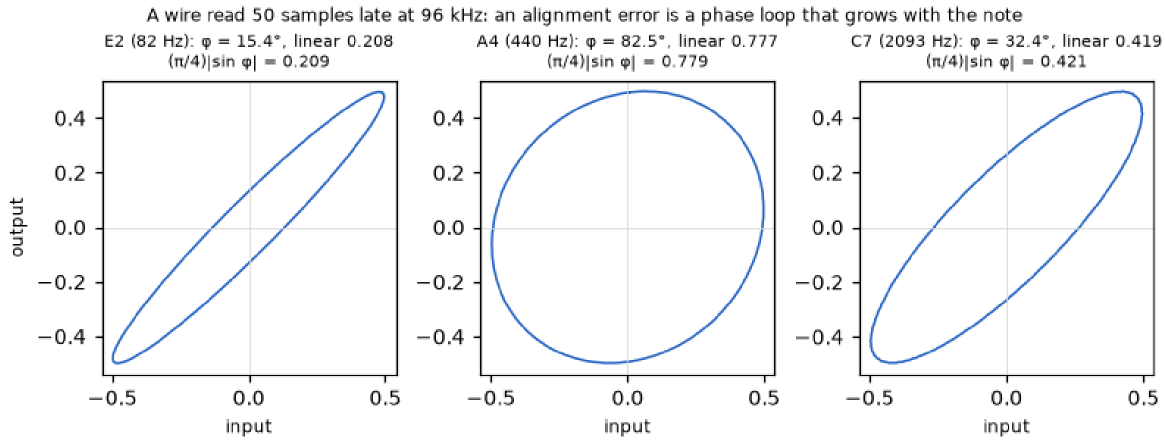

Figure 13: A plain wire, read 50 samples late, at three notes: a timing error is a loop that grows with the note. Each title gives the shift the error amounts to at that note, the measured oval area and the value the formula for a pure shift gives.

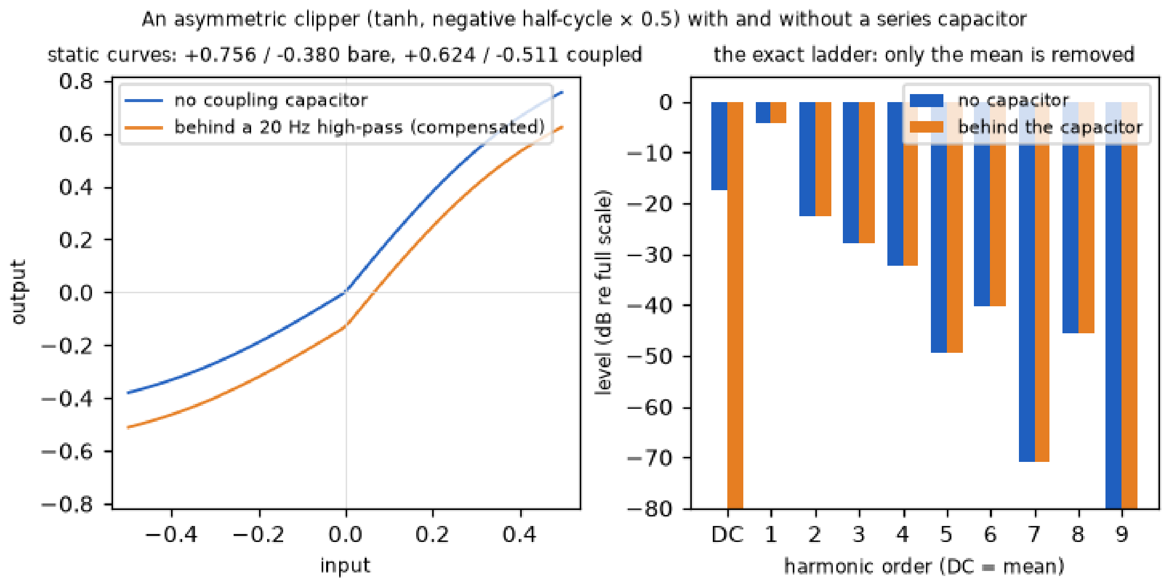

Figure 14: A lopsided clipper (the bottom half scaled to one half) with and without a series capacitor at its output: the static curve's unequal clip levels are centred by the capacitor (left), while the exact harmonic ladder loses only its average level — the even harmonics survive (right).

4 Glossary, in plain language

alias image A recording cannot hold tones above half its sample rate; higher ones fold back down and can land on a harmonic being measured.

averaged cycle Every cycle of the note laid on top of the previous ones and averaged, so that noise cancels and the shape stays — the figure is drawn from that one averaged cycle.

box normalization Loop areas are stated as a fraction of the box the whole figure fits in, so a tiny signal and a loud one are on the same scale.

clipping What a distortion circuit does to a note that is too big for it: the peaks of the wave are flattened or rounded off. A wave whose shape has changed is, by the same token, a wave with new harmonics in it.

coherence (Γ) A score for how orderly a harmonic's measurement is across neighbouring frequencies; real circuit behaviour is orderly, noise is not.

coupling capacitor The capacitor every pedal has on its output. It removes the wave's average level, so a clipper that clips the top and bottom at different heights shows equal heights after it — but the even harmonics that asymmetry makes are still there.

dBc A level relative to the note (or tone) that caused it, so that "how big is the distortion" does not depend on how loud the test was.

dBFS A level relative to the loudest a digital recording can be; 0 is the maximum and negative numbers are quieter.

decibel (dB) A way of writing how much bigger or smaller one level is than another, on a scale where each step of 20 is a factor of ten in size: 0 means the same, -20 means a tenth as big, -40 a hundredth, -60 a thousandth.

deconvolution The step that undoes the sweep, turning the recording into a set of separate blips, one per harmonic.

derotation Lining up repeated sweeps to the fraction of a sample before combining them, so that a tiny timing drift cannot smear the result.

excited band The range of notes the test tone actually played at full strength; the very first and very last moments are ramped and do not count.

extraction window (T_k) The slice of time around one harmonic's blip that is kept for measuring it; wide enough to hold the blip, narrow enough to exclude its neighbours.

frame fit Lining the two measurements up in time by one shift that makes every harmonic agree at once; how well they then agree says whether the two records really describe the same device at the same setting.

fundamental (f_0) The note being played. Distortion shows up as extra tones at whole-number multiples of it.

fundamental ellipse The smooth oval a plain tone filter draws by shifting the note in time: real, and not memory, so it is measured separately and subtracted.

fundamental presence A check that the note itself is still there in the output; if a device removes the note, ratios to it mean nothing and the app says so.

harmonic order (k) Which multiple of the note an added tone is: twice the note is the second harmonic, three times is the third, and so on.

harmonic pulse After the analysis untangles the recording, each harmonic shows up as its own separate blip, spaced apart in time so it can be measured alone.

harmonic response ($H_k(f)$) For each multiple of the note, how much of it the device adds, across the whole range of notes.

harmonic shaping A check that a measured harmonic is really the circuit's doing and not just the device's hiss showing up where a harmonic would be.

impulse response ($h[n]$) The device's fingerprint in time: a sharp blip for the note itself and earlier, smaller blips for each harmonic.

input-normalized Measured relative to the strength of the test tone, so that the numbers compare across drive levels and rigs.

instantaneous frequency ($f(t)$) The pitch the test tone is at, at a given moment.

inverse filter ($\tilde{X}^{-1}(f)$) The exact mathematical undo of the test tone, written down from its formula rather than measured, so it keeps working above the highest note the tone reached.

measurement floor How quiet a measurement can go before it is measuring the recording equipment instead of the pedal; numbers at that level are stated as limits, not values.

memory verdict The app's three-way call on the loop that is left once the tone filter's oval is taken away: small enough to be treated as no memory, large enough to be called memory, or in between — where the app hedges, because a tone filter ahead of the clipper draws the same kind of loop as mild memory and one note cannot tell them apart.

memoryless lattice The two positions a harmonic can sit in, relative to the note, in a pedal with no memory and no filtering: exactly in step with it, or exactly reversed. How far each measured harmonic sits from the nearer of the two is what the filters added, and the removal takes that out.

methods manifest The list of every setting the app uses to make a measurement, written out by the app's own code so that it cannot be typed wrongly. The documents read their numbers from it.

nonlinear loop area The loop that is left once the tone filter's oval is taken away: the part only a real memory effect (or a filter ahead of the clipper, see the phase removal) can draw.

orientation resolution A check that the picture is the right way up: a plugin measurement can come out upside down, and the pedal's own measured polarity is used to turn it back.

pairing residual A single number for how well the two records agree once lined up; too large, and the app declines to apply the correction and says so.

parity test A check, run automatically on every change, that a second, independently written program gets the same answer as the app — and that both get the right answer on devices whose behaviour is known exactly.

phantom capture A made-up recording whose distortion is known to the digit, used to check that the analysis reads back exactly what was put in.

phase removal Using the pedal's own measured harmonic timing to undo what its tone filters did to the picture, so that what is left open is memory and not filtering.

polarity Whether the pedal turns the wave upside down. Stated only when the measurement is sure; otherwise nothing is claimed.

rate constant (L) How quickly the test tone climbs; a larger value means a slower, longer sweep.

read scatter How far a single measured level might be off simply because of noise, stated alongside the level.

residual alignment error A leftover timing error between the test signal and the recording of a fraction of a sample or more; it opens the picture into a loop that is wider for higher notes, and is a property of the rig, not the pedal.

static curve The single line drawn through the middle of a loop, taken as the device's clipping shape: the upward and downward passes averaged together.

sweep averaging Playing the sweep several times in a row and combining the results so that random noise partly cancels while the device's behaviour adds up.

synchronization condition A small adjustment to the sweep's length so that the arithmetic that separates the harmonics works out exactly.

synchronized swept sine ($x(t)$) A test tone that glides smoothly from low to high pitch, timed so that the distortion each note produces can be picked apart afterwards.

tanh The hyperbolic tangent, the textbook soft clipper: a smooth S-shaped curve that passes small signals through unchanged and bends ever more gently toward a ceiling it never quite reaches. It clips the top and bottom of the wave alike, so it adds only odd harmonics, and they fall away quickly. The worked example in these documents is this curve with the input doubled first.

total harmonic distortion (THD) The size of everything the device added at multiples of the note, compared with the note itself, as a single number.

transfer curve A picture of what comes out against what went in, drawn while one steady low note plays: a pedal with no memory draws one line, and a loop means the output depends on where the signal has been as well.

unsigned lobe area How much area the loop encloses, counted lobe by lobe regardless of which way each lobe is traced — a figure-eight counts twice, not zero.

validity bound The range of notes over which a given harmonic can be measured honestly; beyond it the recording itself gets in the way.