

De-Sipper — DSP Manual

Carbonated Audio · v1.0.0 Hard-knee frequency-selective compressor for taming sibilance.

This document explains the signal processing inside De-Sipper from the top down: what each control does mathematically, why it's there, and how to reason about it on a vocal. It's written for engineers and producers who want to know what's actually happening to their audio, not just which knobs sound good.

1. Overview

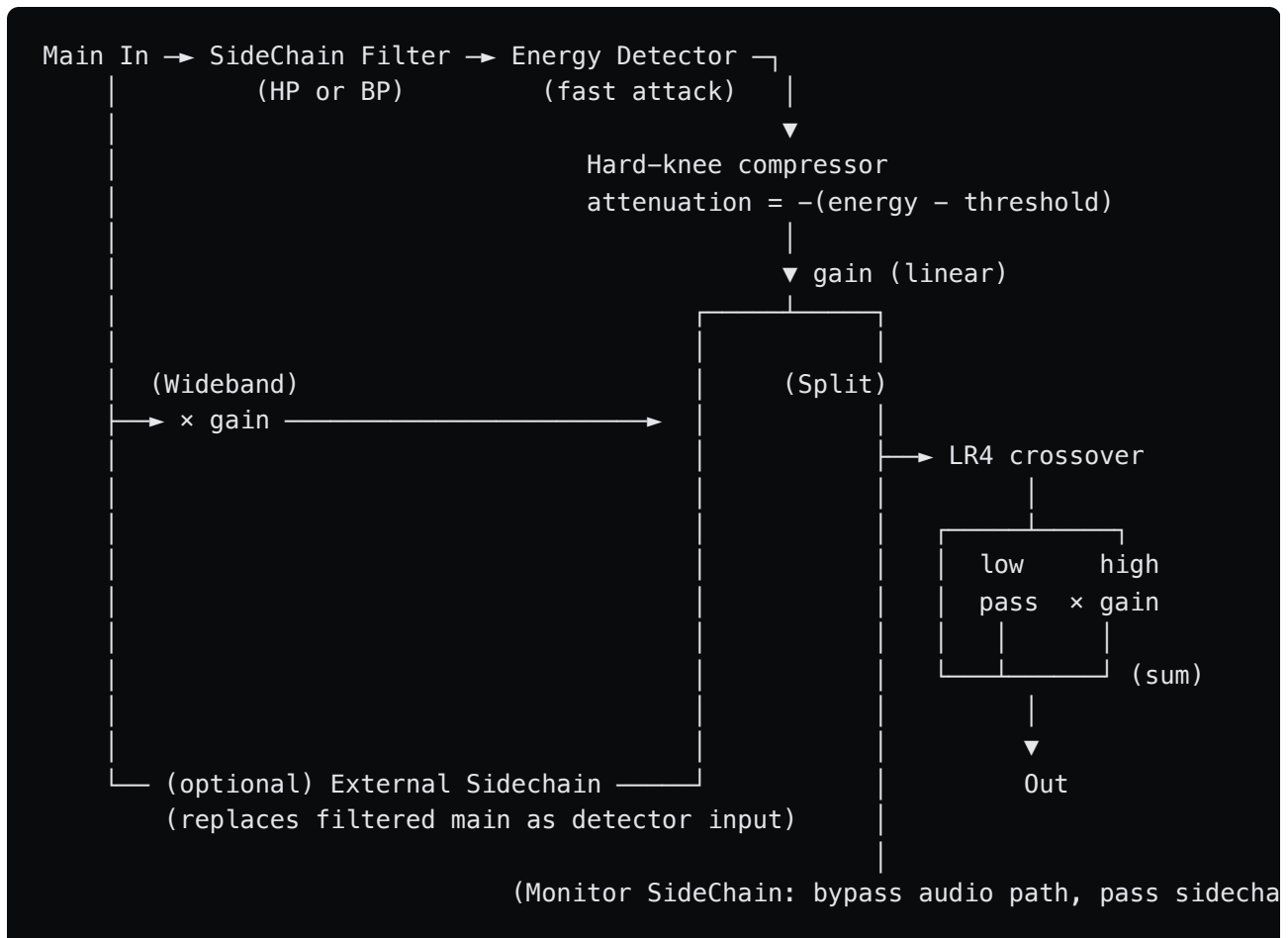
De-Sipper is a **wideband/split hard-knee compressor with a filtered detection path** — the canonical architecture behind every de-esser that's ever worked.

Everything in De-Sipper reduces to four core stages:

1. **Sidechain filter** — take the input, isolate the frequency range that contains the sibilance (usually 4–10 kHz).
2. **Energy detector** — measure the level of that filtered signal in dB over a short time window.
3. **Hard-knee compressor** — if the level is above the threshold, generate a matching attenuation in dB. If it's below, do nothing.
4. **Apply attenuation** — either to the whole signal (Wideband) or only to the high band above the crossover (Split).

There's no knee, no ratio, no attack, no release knobs to tune. De-Sipper's detection path is simple on purpose — the decisions it makes are fast (~1 ms attack, ~50 ms release) and the attenuation law is 1:1 below threshold and ∞ :1 above, in dB. That's what "hard-knee" means here.

2. Signal Flow



Stereo runs two independent filter + detector chains (left and right), but attenuation metering always reports the left channel's values for UI consistency.

3. SideChain Filter

Type: toggle between 2nd-order Butterworth high-pass (default) and narrow band-pass (Q = 4).

Frequency: 1 kHz to 12 kHz, default 5,506 Hz (logarithmic skew), labelled in Hz.

The sidechain filter **never touches the audio path** — it only shapes the detection signal.

HighPass mode (default)

A 2nd-order Butterworth HP at the user's frequency. Everything below is attenuated at 12 dB/oct; everything above passes through to the detector. Use this when you want broad sibilance detection — the detector reacts to all high-frequency energy above your cut point.

```
scFilter(s) = makeHighPass(sampleRate, frequency, Q = 0.707)
```

BandPass mode

A 2nd-order narrow band-pass at the user's frequency with $Q = 4$. Roughly a 375 Hz wide window at 1.5 kHz, 2,750 Hz wide window at 11 kHz — tight enough to isolate a specific problem frequency like a harsh "s" or "sh" band.

```
scFilter(s) = makeBandPass(sampleRate, frequency, Q = 4.0)
```

Use BandPass when the singer has sibilance that sits in a narrow range (e.g. 6.5 kHz on a male voice, 8 kHz on a brighter female voice) and you want the de-esser to react only to that range — not to cymbals, or to the high end of the voice generally.

4. Energy Detector

A simple one-pole peak envelope follower with asymmetric attack/release:

```
if |input| > envelope:
    envelope += attackCoeff * (|input| - envelope)
else:
    envelope += releaseCoeff * (|input| - envelope)

level_dB = 20 * log10(envelope)
```

Attack: ~ 1 ms time constant $(1 - \exp(-1/(sr \times 0.001)))$. **Release:** ~ 50 ms time constant $(1 - \exp(-1/(sr \times 0.050)))$.

Fast attack catches transients before the signal gets loud; moderate release keeps the detector from pumping on sustained speech. The output is a single dB level per block — the detector doesn't need to run sample-by-sample through the attenuation stage.

When the envelope drops below $1e-10$ (effectively digital silence), the reported level floors at **-100 dB**.

5. Threshold and Hard-Knee Compression

Threshold range: -80 dB to 0 dB, default -16 dB.

De-Sipper's compressor isn't a compressor in the knee/ratio sense — it's a **hard-knee** gain cell:

```
if energy_dB ≤ threshold_dB:
    attenuation_dB = 0
else:
    attenuation_dB = -(energy_dB - threshold_dB)
```

Translation: - If the sidechain energy is at or below threshold, there's zero gain reduction. - If the sidechain energy is above threshold, the gain cell attenuates the signal by **exactly the amount by which the energy exceeds threshold**, in dB.

This is ∞:1 ratio, zero-knee compression — everything above threshold gets flattened to threshold level, everything below passes untouched. It's the most honest de-esser behaviour: dial threshold just below the sibilant peaks, and only the peaks get reduced. No character, no colour, no surprise pumping.

The attenuation is then converted to a linear gain factor:

```
gain_linear = 10^(attenuation_dB / 20)
```

6. Audio Mode — Wideband vs Split

Wideband (Split toggle OFF)

The gain factor is applied to the entire signal, full-band:

```
output[i] = input[i] × gain_linear
```

When sibilance hits, **the whole signal ducks** — bass, mids, and highs all attenuate together by the same amount. Wideband is the classic broadcast/recording-engineer de-esser sound.

Split (default — Split toggle ON)

The signal is split through a **Linkwitz-Riley 4th-order crossover** (two cascaded 2nd-order Butterworth filters — one LP, one HP, both with $Q = 0.707$) at the sidechain filter frequency, and the gain factor is applied **only to the high band**:

```
low = lpf(lpf(input, freq, Q=0.707), freq, Q=0.707)
high = hpf(hpf(input, freq, Q=0.707), freq, Q=0.707) × gain_linear
output = low + high
```

The low band passes through untouched; only the high band (where the sibilance lives) is attenuated. Split mode is what you want on a vocal where Wideband would also duck the body of the voice every time an "s" happens — it's surgical.

Crossover math: LR4 = two cascaded Butterworth LP (or HP) stages at $Q = 0.707$. The LP and HP outputs sum to a flat response at unity gain when recombined, so there's no frequency-domain artefact from the split itself.

7. Monitor SideChain

Boolean, default OFF.

A diagnostic listening mode. When active, the audio output is replaced with the filtered sidechain signal — the exact signal the detector is measuring. You hear whatever's above the HP cut or inside the BP window, with no de-essing applied.

This is the fastest way to: - Hear the sibilance in isolation so you can tune Frequency. - Confirm that HighPass vs BandPass is isolating the right material. - Set Threshold by ear — watch the meter and listen for where the "s" energy peaks through.

Turn this off before rendering. It's a monitoring tool, not a processing mode.

8. External Sidechain

De-Sipper declares a stereo main bus plus an **optional mono or stereo sidechain input bus**. Supported hosts:

- **Pro Tools (AAX)** — standard sidechain key input via the plug-in's Key popup.
- **Logic / Ableton / Cubase / any VST3 host supporting aux inputs.**

When the sidechain bus is active, the SideChain Filter and Energy Detector operate on the **external signal**, not the main input. The attenuation computed from the external key is then applied to both main audio channels identically.

Why use it? Common cases: - Feed a dry vocal as key into a de-esser on the same vocal that's been heavily processed (reverb/delay) — detection stays clean. - Drive De-Sipper from a parallel vocal bus to chase a specific performer's sibilance on a group of vocals. - Side-chain from another track entirely to duck high frequencies rhythmically (creative use, not strictly de-essing).

The Frequency and HP/BP mode controls still apply — the sidechain signal is filtered before detection just like the internal path.

9. Meters

De-Sipper reports three meter values to the UI:

- **SideChain Energy (dB)** — current level of the filtered sidechain signal, measured by the energy detector.
 - **Current Attenuation (dB)** — instantaneous gain reduction computed by the hard-knee compressor. Always ≤ 0 .
 - **Peak Attenuation (dB)** — largest attenuation seen since the last peak reset. Useful for setting threshold: if peak never drops below 0, nothing is triggering; if peak reads -20 dB on every phrase, you're over-de-essing.
 - **Output Level (dB, per channel L and R)** — envelope-followed output loudness, with 1 ms attack / 300 ms release. This is what the L/R meters on the plug-in display.
-

10. Latency and CPU

De-Sipper is **zero-latency**. All filters are IIR (not linear-phase), no lookahead, no oversampling.

CPU is dominated by: - One 2nd-order biquad per channel for the sidechain filter - One envelope follower per channel - Four 2nd-order biquads per channel in Split mode (the LR4 crossover: $2 \times LP + 2 \times HP$)

Stereo in Split mode totals 10 biquads per instance. Light; safe to stack on every vocal in a session.

11. Channel Configuration

- **Main I/O:** stereo in, stereo out.
- **Sidechain input (bus index 1):** accepts disabled, mono, or stereo. Stereo external SC currently processes the left channel only through detection but applies the resulting attenuation to both output channels (this matches the behaviour of most pro de-essers — stereo SC detection doesn't usefully differ).
- **Sample rates:** all sample rates supported. Coefficients recompute when sample rate changes.
- **Block sizes:** arbitrary. Detector runs per block, attenuation is a scalar per block, audio path applies that gain per sample.

12. Credits and References

- **De-essing architecture:** classic broadcast de-essers (Orban 622, DBX 902, Waves DeEsser). De-Sipper's "filtered detection + hard-knee compressor + optional band-split" lineage traces back to these.
- **Linkwitz-Riley crossover:** S. Linkwitz, *Active Crossover Networks for Noncoincident Drivers*, JAES 1976. Chosen because LR4 sums flat — critical for the Split mode.
- **Robert Bristow-Johnson**, *Cookbook formulae for audio EQ biquad filter coefficients* — source for the HP/BP/LP biquad math.

Questions, bug reports, or sound-design notes: mixedbysoda@gmail.com.