Hopki User Guide

 Hopki banner{ .hopki-banner }

Hopki User Guide

Hopki analyzes split Hopkinson, also called Kolsky bar, experiments. It takes incident and transmitted gauge signals, applies the same processing stages as the legacy MATLAB `twobar_g` workflow, and produces corrected pulses, forces, velocities, displacements, strain rate, and engineering and true stress-strain curves.

Use this guide when you want to operate the desktop app rather than inspect the source code. It explains what each tab is for, what each button does, and what each editable parameter means.

The App At A Glance

Tab	Role
Analysis	Load an experiment, edit experiment configuration, choose `tim_cut`, adjust dispersion/alignment settings, inspect the analysis plots, and export results.
Explore	Plot any derived quantity against time or true strain, optionally negate axes, and send the displayed curve onward.
Curve cleanup	Trim, zero, smooth, scale, shift, toe-correct, or convert a stress-strain curve before saving or plotting.
Figures	Combine one or more curves into a publication figure, style curves and axes, and export PNG/PDF plus a reloadable `.npz` figure file.

Typical Workflow

1. Prepare an experiment folder with two signal files and either `hopki.toml` configuration or the legacy `incid.exp`, `incid.spec`, and `DAMP_F` files.
2. Open Hopki and use **Load experiment...** on the Analysis tab.
3. Drag the vertical line in **Signals** to set the pulse start, `tim_cut`.
4. Verify the editable parameters, especially signal polarity, gauge distances, specimen dimensions, and reflected-pulse alignment.
5. Use **Export results...** if you need MATLAB-style numeric output files.
6. Send the true stress-strain curve to Figures, or clean a curve first in Curve cleanup.

Important Concepts

`tim_cut`
: The operator-selected pulse start time. In the GUI, it is the vertical draggable line on the **Signals** plot.

`x1` and `x2`
: Gauge-to-specimen distances. `x1` is gauge 1 to specimen, and `x2` is specimen to gauge 2. If omitted from `hopki.toml`, Hopki estimates them from the signals and flags them for verification.

`reflect shift`
: A sample shift applied to the reflected pulse after dispersion correction. It is used to improve force equilibrium.

Signal polarity
: The incident pulse must be compressive, meaning negative in Hopki's sign convention. If curves look inverted or striker velocity has the wrong sign, use **invert signal polarity**.

More Detail

The legacy single-page overview is still included for historical context. The MkDocs pages are the maintained user-facing guide.

Getting Started

Install Or Run

The easiest path is to download a prebuilt binary from the project releases page:

OS	Release asset	First launch
macOS	`Hopki-macos-*.zip`	Unzip, then right-click `Hopki.app` and choose **Open** if macOS warns about an unsigned app.
Windows	`Hopki-windows-*.zip`	Unzip and run `Hopki.exe`. If SmartScreen appears, choose **More info** then **Run anyway**.
Linux	`Hopki-linux-*.tar.gz`	Extract and run `./Hopki`.

To run from source:

```
uv sync --extra gui
uv run hopki-gui
```

To open a specific experiment at launch:

```
uv run hopki-gui path/to/experiment
```

Experiment Folder

Hopki expects an experiment directory with two signal channels and configuration.

Signal files are resolved in this order:

1. A local `hopki.toml` with a `[signals]` table.
2. `incid.inc` and `trans.tra`.
3. The first two `WAVE*.FLT` files by filename, incident first.

Configuration is resolved in layers:

1. Built-in defaults.
2. Any ancestor `hopki.toml` files, useful for campaign or bar constants.
3. The experiment's own `hopki.toml`.
4. Live edits made in the GUI.

If no `hopki.toml` exists in the experiment tree, Hopki falls back to the legacy files `incid.exp`, `incid.spec`, and `DAMP_F` when they are present.

Minimal `hopki.toml`

```toml
[bar]
npoint = 450
diam_bar = 0.0127
E = 1.9e11
c0 = 4800
gfact = 2.13
vbridge = 10.15
```

```
damp_f = -0.018

[specimen]
diam = 0.004
length = 0.004
tdelay_us = 0
pretr_pct = 10

[signals]
incident = "WAVE0003.FLT"
transmitted = "WAVE0004.FLT"
```

`x1` and `x2` may be included in `[bar]`. If either is omitted, Hopki estimates it from the signals and marks the value for verification in the Analysis tab.

First Analysis

1. Click **Load experiment...**.
2. Check that the **Signals** plot shows the incident/reflected channel and transmitted channel.
3. Drag the dashed vertical marker to the start of the incident pulse. This sets `tim_cut`.
4. If the pulse polarity is inverted, enable **invert signal polarity**.
5. Adjust **reflect shift** until the **Force equilibrium** and **Force difference** plots look reasonable.
6. Use **Export results...** to write output vectors, or **sigma-epsilon -> Figures** to start a publication plot.

Analysis Tab

The Analysis tab is the main working area. It loads experiment data, runs the analysis every time a relevant control changes, and shows the intermediate and final plots.

Experiment Controls

Control	What it does
Load experiment...	Opens a folder picker. Choose the experiment directory containing signal files and configuration.
Save config	Writes a per-experiment `hopki.toml` containing only fields that differ from the inherited campaign/default configuration. This is useful after editing dimensions or accepting estimated distances.
Revert config	Reloads configuration from disk and discards live edits made in the GUI.
Directory label	Shows the currently loaded experiment path.
Editable fields	The text fields for bar, specimen, and damping values. Editing a field and leaving it reruns the analysis. See Parameters for every field.
Derived readout	Shows `tpp`, which is read from the signal time column, and warns when `x1` or `x2` was auto-estimated.
Re-estimate x1, x2	Recomputes both gauge distances from the loaded signals and replaces the current values. The status message reports `x1` correlation and `x2` signal-to-noise ratio.
Pick x2 (2 clicks)	Lets you set `x2` manually from the **Signals** plot. Click the incident pulse arrival, then the transmitted pulse arrival. Hopki computes `x2` from the time difference and the acquisition delay.

Parameter Controls

Control	What it does
Theme	Switches the GUI between the available visual themes.
invert signal polarity	Negates both input channels before analysis. Use it when the incident pulse should be negative but appears positive.
Poisson nu	Poisson ratio used to build the Bancroft dispersion table. Range: `0.20` to `0.35`; default: `0.29`.
reflect shift	Integer sample shift applied to the reflected pulse after dispersion correction. Range: `-50` to `50`; default: `0`. Positive values delay the reflected pulse.
Slider below **reflect shift**	A faster way to adjust the same reflected-pulse shift. It stays synchronized with the numeric field.
`tim_cut` readout	Displays the pulse-start time set by the draggable marker on the **Signals** plot.

Graph Controls

The **Graphs** checklist toggles plot visibility. It does not change the analysis; it only hides or shows plots to make the tab easier to scan.

Plot	What to inspect
Signals	Raw incident/reflected and transmitted voltage signals. Drag the vertical line here to set `tim_cut`.
Windowed pulses	The incident, reflected, and transmitted pulse windows cut from the raw signals.
Corrected + aligned	Pulses after dispersion/damping correction, with the reflected pulse shifted by **reflect shift**.
Force equilibrium	`f_in` and `f_out`. These should be compared when tuning alignment and checking whether the analysis window is reasonable.
Force difference	`f_in - f_out`. Drag the shaded region to report mean force difference over a selected time interval.
Stress-strain	Engineering and true stress-strain curves from the current analysis.

Bottom Buttons

Button	What it does
sigma-epsilon -> Figures	Sends the current true stress-strain curve to the Figures tab.
Export results...	Opens an output-folder picker and writes MATLAB-style output files such as `inc_puls`, `ref_corr`, `f_in`, `s_e_eng`, and `s_e_true`.

Status Line

The status line reports the current striker velocity estimate and point count after a successful run. If a field is invalid or a pulse window is out of range, the same area shows the error.

Parameters

This page describes every editable Analysis-tab field and every operator-controlled parameter.

Experiment Fields

These fields are shown in the **Experiment** box on the Analysis tab.

Field	Unit	Meaning
`npoint`	samples	Number of samples in each windowed incident, reflected, and transmitted pulse.
`diam_bar` / **bar diameter**	m	Diameter of the incident/transmitted bars. Used to compute bar cross-sectional area and force.
`E`	Pa	Bar Young's modulus. Used in force calculation.
`x1`	m	Distance from gauge 1 to the specimen. Used to locate the reflected pulse and to apply dispersion correction.
`x2`	m	Distance from the specimen to gauge 2. Used to locate the transmitted pulse and to apply dispersion correction.
`c0`	m/s	Bar wave speed. Used for pulse timing, velocity, strain rate, and dispersion correction.
`gfact`	dimensionless	Strain-gauge factor. Used to convert bridge voltage to strain.
`vbridge`	V	Bridge excitation voltage. Used with `gfact` in the voltage-to-strain conversion.
`diam_spec` / **specimen diameter**	m	Specimen diameter. Used to compute specimen area and stress.
`long_spec` / **specimen length**	m	Specimen length. Used to compute engineering strain rate.
`tdelay_us`	microseconds	Acquisition delay for the transmitted signal. Hopki pads the transmitted record by this delay before windowing.
`pretr_pct`	percent	Pre-trigger percentage. Used to guess the initial `tim_cut` when an experiment loads.
`damp_f`	coefficient	Amplitude damping coefficient used during frequency-domain correction.

Derived Values

Value	Source	Notes
`tpp`	Signal time column	Time per point. This is intentionally not editable because it belongs to the recorded signal.
`x1` estimate	Signals	Used only when `x1` is absent from all config layers. Usually more reliable than `x2`.
`x2` estimate	Signals	Used only when `x2` is absent from all config layers. Treat as best-effort and verify with **Pick x2 (2 clicks)** when needed.

Operator Parameters

Parameter	Control	Default	Meaning
`tim_cut`	Vertical line on **Signals**	Initial guess from `pretr_pct`	Pulse-start time. The incident pulse window starts here; reflected and transmitted windows are positioned from `x1`, `x2`, `c0`, and `tdelay_us`.

Parameter	Control	Default	Meaning
`invert_signals`	**invert signal polarity**	off	Negates both input channels before analysis.
`nu`	**Poisson nu**	`0.29`	Poisson ratio used by the Bancroft phase-velocity table for dispersion correction.
`reflect_shift`	**reflect shift** spinbox/slider	`0`	Sample shift applied to the reflected pulse after correction. Positive values delay it; negative values advance it.

How Hopki Uses The Values

The core pipeline runs in four stages:

1. **Window pulses** using `tim_cut`, `x1`, `x2`, `c0`, `tdelay_us`, `nlong`, and `npoint`.
2. **Correct dispersion and damping** using `nu`, `diam_bar`, `c0`, `x1`, `x2`, `tpp`, `npoint`, and `damp_f`.
3. **Align the reflected pulse** using `reflect_shift`.
4. **Compute mechanics** using `diam_bar`, `E`, `gfact`, `vbridge`, `diam_spec`, `long_spec`, `c0`, and `tpp`.

Configuration Files

Modern configuration uses `hopki.toml`. Ancestor files are applied before the experiment-local file, so a campaign-level file can hold shared bar constants and each experiment can hold only specimen-specific values.

```toml
[bar]
nlong = 1000
npoint = 450
diam_bar = 1.27e-2
E = 1.9e11
x1 = 0.558
x2 = 0.230
c0 = 4800
gfact = 2.13
vbridge = 10.15
damp_f = -1.8e-2

[specimen]
diam = 4e-3
length = 4e-3
tdelay_us = 0.0
pretr_pct = 10.0

[signals]
incident = "WAVE0003.FLT"
transmitted = "WAVE0004.FLT"
```

Legacy folders with no `hopki.toml` can still use `incid.exp`, `incid.spec`, and `DAMP_F`.

Explore Tab

The Explore tab is a free-form viewer for derived analysis quantities. It does not change the analysis result; it only changes what is plotted and what curve is sent to cleanup or figures.

Controls

Control	What it does
y dropdown	Chooses the derived quantity plotted on the vertical axis.
x: time -> strain	Switches the horizontal axis from time to true strain. The button text changes to **x: strain -> time** when strain mode is active.
Negate x	Multiplies the displayed x values by `-1`.
Negate y	Multiplies the displayed y values by `-1`.
Send to cleanup ->	Sends the currently displayed `(x, y)` curve to the Curve cleanup tab.
Send to figures ->	Adds the currently displayed curve to the Figures tab.

Available Y Quantities

Quantity	Backend value	Meaning
`strain rate e_dot`	`eps_rate_eng`	Engineering strain rate.
`v_in`	`v_in`	Velocity at the input/specimen side.
`v_out`	`v_out`	Velocity at the output/specimen side.
`striker velocity`	`v_striker`	Striker velocity estimate from the incident pulse.
`u_in`	`u_in`	Input-side displacement.
`u_out`	`u_out`	Output-side displacement.
`f_in`	`f_in`	Input-side force.
`f_out`	`f_out`	Output-side force.
`eng. stress`	`str_eng`	Engineering stress.
`true stress`	`str_true`	True stress.

When To Use It

Use Explore when you need to inspect a signal that is not one of the fixed Analysis plots, compare a derived quantity against true strain, or pass a non-standard curve into cleanup or figure assembly.

Curve Cleanup Tab

The Curve cleanup tab wraps Hopki's `figcorr` operations in an interactive UI. It is intended for stress-strain cleanup after analysis, but it can also load any compatible two-column curve file.

Source

Button	What it does
Pull true sigma-epsilon	Loads the current true stress-strain curve from the Analysis result.
Pull engineering sigma-epsilon	Loads the current engineering stress-strain curve from the Analysis result.
Load curve file...	Opens a two-column curve file from disk.

Button	What it does
-> True sigma-epsilon	Converts the current engineering curve to true stress and true strain. Disabled when the current curve is already true stress-strain.

Trim / Crop

The vertical line on the plot is the cut/crop pointer.

Button	What it does
Zero	Shifts the curve so the first point is at the origin.
Cut < line	Keeps points up to the vertical line and removes points after it.
Crop line >	Removes points before the vertical line and keeps the rest.
Straighten (2 clicks)	Click two points on the linear toe region. Hopki subtracts the fitted line offset to toe-straighten the curve.
Slope (2 clicks)	Click two points and Hopki reports the slope. The curve is not modified.

Transform

Control	What it does
Negate sigma	Multiplies stress values by `-1`.
Negate epsilon	Multiplies strain values by `-1`.
Scale factor field	Numeric factor used by the scale buttons. A value of `0` is rejected.
Scale sigma	Multiplies stress by the scale factor.
Scale epsilon	Multiplies strain by the scale factor.
Shift amount field	Numeric constant used by the shift buttons.
Shift sigma	Adds the shift amount to stress.
Shift epsilon	Adds the shift amount to strain.

Filter

Control	What it does
Smooth...	Opens a power-spectrum preview and overlays a filtered curve.
cutoff	Butterworth low-pass cutoff as normalized frequency. `1` means Nyquist; default is `0.20`; range is `0.01` to `0.99`.
Cutoff line	Draggable line in the spectrum plot. It stays synchronized with the cutoff field.
Apply filter	Commits the previewed smoothed curve.
Cancel	Leaves smoothing preview mode without changing the curve.

Output

Button	What it does
Back	Undoes the last committed cleanup operation.
-> Figures	Sends the current cleaned curve to the Figures tab.
Save...	Saves the current curve. The default filename is based on the source, usually `<source>_corr`.

Status Messages

The status line tells you what operation was applied, reports slopes, and shows input errors such as invalid scale factors or failed file loads.

Figures Tab

The Figures tab builds publication figures from one or more curves. It keeps figure data and styling together in a reloadable `.npz` file, and it exports PNG or PDF through matplotlib.

File Buttons

Button	What it does
Add curve(s)...	Loads one or more two-column curve files and appends them to the current figure.
Load figure...	Loads a saved `.npz` figure. If the current figure is empty, Hopki adopts its axes and settings; otherwise loaded curves are appended.
Save (.npz)...	Saves all curves plus figure metadata and styling to a reloadable `.npz`.
Export PNG/PDF...	Exports the figure as an image and writes a `.npz` sidecar next to it.

Curve List

Control	What it does
Curve list	Selects which curve is edited by the **Selected curve** controls.
Delete curve	Removes the selected curve from the figure.
-> Cleanup	Sends the selected curve, as currently shown after trim and scaling, to the Curve cleanup tab.

Selected Curve

Control	Meaning
legend label	Name shown in the figure legend and curve list.
colour	Opens a color picker. The button shows the selected color.
line	Line style: `solid`, `dash`, `dot`, `dashdot`, or `none`.
line width	Line width. Range: `0.5` to `12.0`; default is `2.0`.
symbol	Marker symbol: `none`, `o`, `s`, `t`, `d`, `+`, or `x`.
symbol size	Marker size. Range: `2.0` to `30.0`; default is `8.0`.
x scale	Multiplies x values when drawing. Use it for unit conversions, such as strain to percent.
y scale	Multiplies y values when drawing. Use it for unit conversions, such as Pa to MPa.
show in legend	Controls whether the selected curve appears in the legend.

The shaded region on the plot trims the selected curve's displayed x range. The trim is stored in raw x units, so changing **x scale** does not change which data points are included.

Axes

Control	Meaning
x title	Horizontal axis title.
y title	Vertical axis title.
x format	printf-style numeric tick format for the x axis, for example `%g` or `%.2f`.
y format	printf-style numeric tick format for the y axis.
auto x limits	Lets Hopki autoscale the x axis.
x min / max	Manual x-axis limits when auto x limits is off. Invalid or inverted limits are ignored.
auto y limits	Lets Hopki autoscale the y axis.
y min / max	Manual y-axis limits when auto y limits is off. Invalid or inverted limits are ignored.

Publication Export

These settings affect exported PNG/PDF files. The live preview remains a pyqtgraph preview.

Control	Meaning
style	Matplotlib/SciencePlots style used for export.
LaTeX titles	Uses LaTeX-rendered text when a TeX installation is available. If TeX is missing, the option is disabled and mathtext is used.
size w x h [in]	Export figure width and height in inches. Defaults: `6.4 x 4.8`.
tick font	Tick-label font size in points. Default: `12`.
axis-title font	Axis-title font size in points. Default: `14`.
legend font	Legend font size in points. Default: `12`.

Command Line

Hopki also provides command-line tools. Use them when you need repeatable batch processing or want to run the same operation outside the GUI.

`hopki-gui`

Launches the desktop app.

```
uv run hopki-gui
uv run hopki-gui path/to/experiment
```

If a directory is supplied, Hopki opens it as the initial experiment.

`wft-to-csv`

Converts Nicolet `.WFT` oscilloscope captures into legacy `.FLT` text files.

```
uv run wft-to-csv input.WFT output.FLT
```

Use `--help` for batch options such as output directories.

`twobar-analyze`

Runs the core analysis pipeline on an experiment directory and writes MATLAB-style output vectors.

```
uv run twobar-analyze EXP_DIR \
  --tim-cut 2.83e-5 \
  --reflect-shift 3 \
  --nu 0.29 \
  --invert \
  --out OUT_DIR
```

Option	Meaning
`EXP_DIR`	Experiment directory.
`--tim-cut`	Required pulse-start time in seconds.
`--reflect-shift`	Reflected-pulse shift in samples. Default: `0`.
`--nu`	Poisson ratio for dispersion correction. Default: `0.29`.
`--invert`	Negates input signals before analysis.
`--out`	Output directory. Defaults to the experiment directory.

`figcorr`

Runs curve cleanup operations from the command line.

```
uv run figcorr CURVE --cut 256 --zero
```

Use `uv run figcorr --help` for the full operation list.