SpeAcouPy Documentation

User guide and concepts

None

None

Table of contents

1. SpeAcouPy Documentation

Welcome! This site is for **users who want to simulate loudspeaker systems** with the **SpeAcouPy CLI**.
You don't need Python knowledge to *use* the tool; copy the YAML examples and run the CLI.

- Start with **Getting started**.
- Learn the ideas behind the tool in **Concepts**.
- Explore a hands-on tutorial: **Sealed box**.
- Prefer offline reading? **Download the full PDF**.

If you're a power user interested in Python internals, see the (future) **Advanced** appendix.

2. Getting started

2.1 Install SpeAcouPy

- Install SpeAcouPy following the instructions in the project README (release binaries or pip).
- Verify installation:

```
speacoupy --version
```

2.2 Your first run (no files created)

```
speacoupy --help
```

This prints the available commands and options to your **terminal**. No files are created yet.

2.3 Run a simple simulation (sealed box)

1) **Download the example config**: sealed.yaml
2) **Run the simulation**:

```
speacoupy simulate examples/sealed.yaml
```

What happens: - SpeAcouPy reads parameters from `examples/sealed.yaml`. - It runs the sealed-box simulation. - It writes results to an output folder (plots as PNG, data as CSV). The default location and filenames depend on the CLI options; see **CLI → Overview** for details.

> Tip: Copy `examples/sealed.yaml` to your working directory and tweak values to explore different box sizes and drivers.

3. Concepts

3.1 Concepts overview

SpeAcouPy models loudspeaker systems using **lumped acoustic elements**. Typical elements include: - **Driver**: moving-coil loudspeaker modeled by its Thiele/Small parameters. - **Enclosure**: sealed or vented volumes represented by compliance and losses. - **Ports / ducts**: masses with end corrections and viscous losses. - **Crossovers / filters**: electrical networks affecting input voltage/current. - **Radiation**: baffle/port radiation models to predict on-axis SPL.

Assumptions (high level): - Small-signal, linear operation. - Frequency-domain steady-state response. - Quasi-1D lumped parameters (valid below modal region of enclosures/rooms).

See the tutorials for concrete examples.

4. CLI

4.1 CLI overview

The CLI organizes actions into subcommands. The exact list may evolve; see:

```
speacoupy --help
```

and

```
speacoupy simulate --help
```

Common patterns: - `speacoupy simulate <config.yaml>` : run a simulation using a YAML configuration. - `--output <folder>` : choose an output directory (plots, CSV). - `--format <png|svg|csv>` : choose output formats if available. - `--plots` : request generation of standard plots (SPL, impedance, phase).

The **Commands** page can be autogenerated in CI from `--help` output so it always matches the current release.

5. Configuration

5.1 Configuration files (YAML)

Simulations are driven by **YAML** configuration files. A minimal sealed-box example:

```yaml
system:
  driver:
    fs: 50       # Hz
    qts: 0.38
    vas: 20      # liters
enclosure:
  type: sealed
  volume: 15     # liters
```

Guidelines: - Use **SI units** unless otherwise stated. - Keep numbers realistic; see your driver's datasheet for T/S parameters. - Start simple; add details (losses, ducts, filters) incrementally.

6. Tutorials

6.1 Tutorial: Sealed box

Goal: Predict on-axis SPL and impedance of a simple sealed enclosure.

6.1.1 1) Prepare the config

Download: examples/sealed.yaml

```
```

6.1.2 2) Run the simulation

```
speacoupy simulate examples/sealed.yaml
```

6.1.3 3) Inspect results

- **SPL plot**: expected smooth roll-off with system Q near `Qtc`.
- **Impedance plot**: single broad resonance peak near box-loaded `Fc`.

If results look odd, double-check units (liters vs m^3, Hz vs rad/s) and typos.

7. Results

7.1 Interpreting results

Typical outputs (depends on CLI options): - **SPL vs frequency** (on-axis): dB referenced to 20 µPa at 1 m. - **Electrical impedance**: real and imaginary parts or magnitude/phase. - **Cone excursion**: useful to assess limits at low frequencies (if available). - **Port velocity** (vented): to check chuffing risk (if available).

Rules of thumb (sealed): - Box **volume ↓** → **Fc ↑**, **Qtc ↑** → more peaking and earlier roll-off. - Driver with **lower Qts** in a small box → tighter low-end, less extension.