Python in the workbench

This page explains how to run the AD101 explorers and scripts inside the IIC-OSIC-TOOLS noVNC desktop you set up in IC101, and what to do when something goes wrong. The first time you load the course, Lab 01, Part 1 walks you through it.

One-time setup

Start noVNC the way you did in IC101 (`./scripts/start_vnc.sh` from your `workspace` folder), open a terminal in the desktop, and run:

```
. /foss/designs/common/.designinit
mod add ad101                # first time only: clones into modules/ad101
mod ad101                    # → /foss/designs/modules/ad101
python3 scripts/check_env.py
```

If you'd rather not use `mod add`, you can clone the repo on the host instead:

```
# on the host, inside your workspace clone
cd workspace/modules
git clone https://github.com/uoftasic/ad101.git
```

`check_env.py` should print `OK` for numpy, for matplotlib and for the `adsig` smoke line, and finish with `=== All checks passed ===`.

Running an explorer

```
cd labs/lab-01-signal-explorer
python3 src/explore.py              # interactive windows
python3 src/explore.py --figure f1  # one figure
python3 src/explore.py --headless   # PNGs into results/ (no GUI)
```

The terminal stays busy until you close the plot windows, so you won't get a prompt back while they're open. You can close a window with its × button, or by pressing `q` while the figure has focus.

Running the snippets

The short Python snippets in Labs 01–04 import only `numpy` and `matplotlib`. Unlike the explorers, they don't need `_path.py` or `mod ad101`, so they run in any Python 3 that has those two packages, whether that's the noVNC desktop, your host machine or your own laptop. Save each snippet as its own `.py` file and run it with `python3 <file>.py`. In the desktop, anything you save under `/foss/designs` is stored in your `workspace` folder on the host, so it survives a container restart.

GUI backend

The interactive explorers need a matplotlib GUI backend (Tk or Qt) and a display to draw on, and the noVNC desktop provides both. On a host machine without a display, `check_env.py` reports the `Agg` backend instead. That's fine for `--headless` and for the maintainers' `build_figures.py` (described further down), because they only write PNG files and never open a window. If you get a backend warning inside IIC-OSIC-TOOLS and noVNC, run `check_env.py` again, and if it still warns, use one of the fixes below or run with `--headless`.

If the GUI won't open

`check_env.py` may report `WARN no interactive GUI backend`, which means the explorers have nothing to open their windows with. You have three ways round it:

1. Use headless mode. You lose the interactive windows, but the exported PNGs are the same figures you see on the guide pages, so you can still learn from them:

```
python3 src/explore.py --headless
```

2. Install a GUI backend inside the container:

```
pip install --user PyQt5
# or make sure Tk is available; then re-run check_env.py
```

3. Check that you're typing in a terminal inside the noVNC desktop. A terminal on the host has no display for the windows to open on.

Saving a figure

- To save the figure in an explorer or snippet window, click the save icon (the floppy disk) in the window's toolbar.
- To save every figure from an explorer at once, run `python3 src/explore.py --headless`, which writes PNGs into the lab's `results/` folder.
- To save from your own snippet, add `plt.savefig("my-capture.png", dpi=140)` just before `plt.show()`.

Listening to a filtered signal (Lab 05)

Lab 05 exports and plays audio so that you can hear what a filter does to a chord. Live playback needs a sound device, and the noVNC desktop may not pass one through from the host, so the lab has you write a fallback, in the same spirit as `--headless` when there's no GUI backend:

1. The `play_audio` function you write tries live playback with `sounddevice`. Neither `sounddevice` nor `soundfile` is installed in the course image; both are listed as optional in `labs/lab-05-signal-workshop/requirements.txt`, and you can install them with `pip install --user sounddevice soundfile`.
2. If playback fails, your fallback prints the path of each `.wav` file, which the script has already written into `results/`, and you can open them from your host machine's file manager or media player.

Either way, you can compare the filtered chord with the unfiltered one.

Regenerating the docs figures (maintainers)

From the AD101 repo root:

```
python3 scripts/build_figures.py
# → docs/assets/img/f01-….png … f15-….png
```

The noise figures use fixed random seeds (`RNG_SEED = 42` in `labs/common/adsig.py`, and `HIDDEN_SEED = 7` for F8 in Lab 03's `src/explore.py`), so a rebuild produces the same PNGs and doesn't leave a pile of changed images in git.

Common errors

Symptom	Fix
`ModuleNotFoundError: common`	Run it from the lab folder as `python3 src/explore.py`, which uses `_path.py` to find `common`, or run `mod ad101` first

Symptom	Fix
`ModuleNotFoundError: numpy`	Install the packages with `pip install --user numpy matplotlib`
Blank or frozen window over noVNC	Open fewer windows (`--figure f1` opens just one), close any extras, and check your VNC bandwidth
`check.py` says `Not quite`	Read the spectrum peak again; your answer has to be within ±5 Hz
`NotImplementedError: TODO: …` in Lab 05	This is expected. Fill in the function the message names, then re-run

Related

- IC101: Launch noVNC
- Getting started