OpenSD2025: Summer School program

OpenSD2025 website

Four individual tracks are planned for the OpenSD2025 Summer School:

1. Vibration fatigue
2. High-speed camera identification
3. Substructuring
4. Collaboration on open-source projects

On Wednesday, June 18th, all tracks will attend shared introductory classes.
On Thursday, June 19th, participants will split into four groups based on their chosen tracks.

Pre-event Preparation:
Registered students will have access to summer school materials before the start of the event (beginning of June), allowing them to review key concepts and prepare in advance.

Post event assignments (ECTS Credits):
Students interested in earning ECTS credits will be required to complete assignments related to their chosen track after the event. Further details on these assignments, including deadlines and evaluation criteria, will be provided during the program.

Detailed Summer School tracks breakdown:

Wednesday June 18th - Summer school common track

Intro to Python, numerical methods, signal processing (*3h of lectures, 3h of hands-on work).*

- The Python Ecosystem
  - Basic data types and operators
  - Error handling in Python
  - Functions and Control Flow (If, For, While)
  - Objects and Modules in Python
- Essential numerical tools in Python
  - Basic numerical packages: Numpy, Scipy, Matplotlib
  - Interpolation and Approximation
  - Numerical Differentiation and Integration
  - Systems of linear equations
- Signal processing basics for vibration engineers
  - The Fast Fourier Transform
  - Windowing, the Welch's method and Convolution
  - Experimental Modal Analysis (FRFs, Curve fitting and Reconstruction)

Thursday June 19th - Summer School individual tracks:

Track 1: Vibration fatigue

3h of lectures, 4h of hands-on work.

- Stress response of dynamic structures under base excitation
  - Excitation signal types (deterministic, random, non-gaussian, non-stationary)
  - Stress response (time waveform, power-spectral density)
  - Methods for fatigue life estimation (spectral methods, rainflow)
- Signal preparation and fatigue life estimation (open-source packages pyExSi, FLife)
- Vibration fatigue test (Part 1):
  - Electro-dynamic shaker setup
  - Signal generation
  - Acceleration acquisition, evaluation of damage rate
- Vibration fatigue test (Part 2):
  - Dynamic specimen setup, mounting, strain-gage
  - Setup of numerical model
  - Stress-response acquisition in the fatigue zone
  - Damage rate estimation (from experiment and numerical model)

Track 2: High-speed camera identification

3h of lectures, 4h of hands-on work.

- Identification of motion from image data
  - Simplified Optical Flow
  - Lucas-Kanade
- Vibration test case (pyIDI)
- Image-based EMA
  - Basic experimental skills (lighting, surface preparation, focus,...)
- Experimental work
  - Experiment preparation (speckle pattern, lighting, ...)
  - Experiment execution (high-speed camera, modal hammer, accelerometer)
  - Analysis (SDyPy, hybrid method)

Track 3: Substructuring

3h of lectures, 4h of hands-on work.

- Theoretical background:
  - Frequency based substructuring (interface conditions, weakening).
  - Transfer path analysis (on the use of blocked forces to characterize the source).
- Experimental work (for the purpose of in-situ TPA):
  - FRF test on pre-prepared setup.
  - Operational test.
- In-situ TPA:
  - Source characterization, transferability to modified assemblies and identification of critical transfer paths.

Track 4: Collaboration on open-source projects

3h of lectures, 4h of hands-on work.

- Git and GitHub basics
  - Setting up and working with code repositories
  - Git workflow: clone, add, commit, push, Forks and Pull requests
  - Collaborating on open-source projects: hands-on example
- Code Documentation, Testing and Continuous Integration
  - Docstrings, project documentation using sphinx and autodoc
  - Automatic code testing using pytest and GitHub Actions
- Preparing and distributing Python packages
  - The pyproject.toml file
  - Generating project distribution archives (wheel)
  - Uploading to PyPi