SCIENTIFIC COMPUTING – FIRST LAB

Prof. Sebastián Roldán Vasco

MASTER IN AUTOMATION AND INDUSTRIAL CONTROL

Deadline: March 28, 2025

Dear student, read the following instructions carefully. Your grade will depend on their fulfillment.

GRADED WORK:

- Written report in a Jupyter Notebook[1] uploaded to a GIT-related platform, i.e. GitLab or GitHub.
- Data, scripts and functions would be self-contained in the notebook, i.e. without requiring external dependencies or paths, additional configurations, or missing components
- The work must be written entirely in English

GOAL:

In this assignment, you are required to use Python and Git to implement a data-driven scientific analysis. You will use your own data (e.g., images, signals, electrical variables) and apply key programming concepts such as Python fundamentals, object-oriented programming (OOP), numerical derivation and integration, vectorization, and handling of matrices, vectors, and tensors. In case you have not yet collected data, you are allowed to use synthetic or publicly available data, but it must be related to your research topic.

ACTIVITY:

1. Context. Provide a short theoretical background about your research problem with references at the beginning of the Notebook. At the end of the

[1] If you decide to work in Matlab against the course's recommendation, you can use the live script option in MLX format.

notebook, provide a discussion section and conclusions about the results obtained in this assignment.

Grading Criteria:

- Theoretical background (0.2 points)
- Discussion of results (0.2 points)
- Conclusions (0.1 points)

2. Data loading and initial visualization. Choose a dataset and describe the data characteristics. Ensure data are available in a format suitable for matrix/tensor manipulation (e.g., .csv, .wav, .jpeg, etc.). Explain how and why you chose the dataset. Visualize a representative sample of your data. If it is an image, display an example. Plot the data points if it is a signal or a time series.

   Grading Criteria:

   - Proper data loading and handling (0.5 points)
   - Clear and informative visualizations (0.3 points)

3. OOP for data processing. Implement an OOP-based design for your data analysis pipeline. Define at least two classes related to your dataset (e.g., ImageProcessor, SignalAnalyzer, ControlSystemAnalyzer). Each class should encapsulate relevant methods for data manipulation: for example, if working with images, use methods for contrast or resizing; if working with signals or electrical variables, ensure filtering, noise removal, or feature extraction. Use Python functions, loops, conditionals, and list comprehensions where applicable.

   Grading Criteria:

   - Correct application of Python fundamentals (0.2 points)
   - Effective use of object-oriented principles (0.7 points)
   - Relevant preprocessing steps (0.4 points)

4. Numerical derivation or integration of data. Implement a numerical derivative calculation or integration on your dataset. For images, the derivative could involve computing gradients or edge detection using finite difference methods. The integration computes cumulative intensity values or integrates over pixel values in specific regions.

For signals or control systems data, the derivative calculates the rate of change over time (e.g., current or voltage over time), whilst the integration can compute the total energy consumption or signal area under the curve.

Visualize the results of the derivative.

Grading Criteria:

- Correct implementation of numerical calculations (0.6 points).
- Clear visualization and explanation of results (0.6 points).

5. Matrix operations. Perform matrix or tensor operations on your data.

For images, treat them as matrices and apply operations such as matrix multiplication, tensor slicing, or eigenvalue computation.

For signals or electrical data, represent the data as matrices or higher-dimensional tensors and apply similar operations (e.g., matrix multiplication, decomposition, or transformation).

Grading Criteria:

- Correct implementation of matrix or tensor operations (0.8 points).
- Clear explanation of results and relevance to the data (0.4 points).

Deliverables:

GitHub repository link (or equivalent).