SCIENTIFIC COMPUTING – SECOND LAB

Prof. Sebastián Roldán Vasco

MASTER IN AUTOMATION AND INDUSTRIAL CONTROL

Deadline: May 02, 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 dataset (e.g., images, signals, electrical variables, or any data relevant to control systems) and submit a Python program that demonstrates your understanding of data handling, statistical analysis, data separability, and performance evaluation.

ACTIVITY:

1. Provide a shorter version of the context shown in the first assignment, in order to explain the data you will use in this work
   Grading criterion:
   - Brief contextualization (0.2 points)

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

2. Advanced data handling with Pandas. Use Pandas to clean, preprocess, and structure your dataset for analysis. Handle missing data, restructuring, and perform transformations needed to make the dataset ready for analysis. Load your dataset and filter it for specific features or time frames that are meaningful for analysis. Restructure the data as necessary: examples include aggregating data over time intervals, creating specific categories for variables, or adjusting formats for easier analysis (e.g., from a 3D array to 2D or vice versa if working with image data). Demonstrate handling of missing data, outliers, or inconsistencies within the dataset.

   Grading Criteria:

   - Proper data loading in a structured format and handling of inconsistencies (0.5 points)
   - A Markdown section explaining each data handling step, with an emphasis on why these transformations were applied (0.5 points)

3. Feature extraction using OOP. Extract relevant features to your dataset using the OOP paradigm. Define at least two feature domains, for instance, time/spatial (e.g. RMS, energy, intensity, edges), frequency (e.g. power spectrum), or time-frequency (e.g. wavelet-based coefficients). Ask your advisor for possible candidate features to be analyzed, focusing on identifying key properties and patterns within the data that are relevant to your dataset's context

   Grading Criteria:

   - Correct calculation and extraction of features or mathematical descriptors (0.7 points)
   - Brief explanation about the meaning of the extracted features and their importance for your research problem (0.3 points)

4. Statistical analysis of custom data and feature extraction. Conduct a statistical analysis of your feature space. Calculate and interpret relevant summary statistics (e.g., mean, median, variance) and distributions for the chosen features, and display the results visually where applicable. Make that the dataset includes multiple categories or classes, apply statistical tests (e.g., t-

tests, ANOVA, Fisher's ratio) to determine differences across groups. Perform an analysis of how distinct or separable different groups or categories are within your dataset, both visually and numerically.

Grading Criteria:

- Correct calculation and application of statistical measures/tests (0.5 points)
- Clear visualizations and meaningful interpretations relevant to the dataset (0.5 points)
- Clear interpretation of separability measurements in the context of the dataset (0.4 points)

5. Data separability. Compute performance metrics relevant to your data analysis goals. For datasets with labels, calculate classification performance metrics such as accuracy, precision, recall, F1-score, or ROC-AUC for your model. For continuous or time-series datasets, define performance metrics relevant to your dataset (e.g., root-mean-square error, correlation coefficient, or spectral match) and evaluate how well data transformations or models meet these metrics.

   Grading Criteria:

   - Correct implementation and calculation of appropriate performance metrics (0.6 points)
   - Insightful interpretation and comparison of performance results (0.8 points)

Deliverables:

1. GitHub repository link (or equivalent).