IM²AG - Licence 3 Informatique :
Mathématiques et Informatique

Application internship report

Characterization of Simple Electronic Filters Using PINNs (Physics-Informed Neural Networks) and Study of the Data/Equations Trade-Off

Students :
Sahiya Aboubacar Moussa,
Abdel Kader El Houssami,
Meryeme Ghannam,
Kimia Khademlou,
Maram Marsaoui,
Ho Minh Khanh Nguyen,
Hugo Viana

Academic advisor :
Marek Bucki

Client :
Nathan Chanez,
STMicroelectronics

2024

Contents

Introduction

This preliminary report aims to lay the groundwork and organize the planning of a project focused on exploring the use of Physics-Informed Neural Networks (PINNs) for the characterization of simple electronic filters. The project emphasizes reducing data dependency by leveraging physical laws to enhance prediction efficiency. Through an agile methodology and a clear roadmap, this document outlines the planned steps, tools to be used, and objectives to be achieved, ensuring a structured and efficient project management approach before its launch.

1 Context

1.1 STMicroelectronics

STMicroelectronics is a Franco-Italian company with its executive and operational headquarters located near Geneva, Switzerland. Founded in 1987 following the merger of the Italian company SGS (Società Generale Semiconduttori) and the French company Thomson Semiconductors, it has become one of the global leaders in the semiconductor industry. Specializing in the design and manufacture of electronic components, the company is a major player in the automotive, industrial, communications, and connected objects sectors. Leveraging its technological expertise, STMicroelectronics develops innovative solutions to address the growing demands for energy efficiency, connectivity, and artificial intelligence.

1.1.1 Products

STMicroelectronics offers a wide range of products, from microcontrollers and microprocessors (notably the popular STM32 and STM8) to MEMS sensors, image sensors, and analog solutions for energy management. The company also stands out in power modules for electric vehicles as well as IoT and connectivity systems (Bluetooth, RF, NFC). It excels in the design of application-specific integrated circuits (ASICs) for industrial and automotive applications. These products address the needs of key sectors such as automotive, industry, connected objects, and renewable energy, thus consolidating STMicroelectronics' position as a leader in innovative and sustainable semiconductors.

Figure 1: ST electronic chip

1.1.2 Activity and clients

STMicroelectronics carries out diverse activities, ranging from research and development to the design, manufacturing, and commercialization of semiconductors. The company excels in developing innovative solutions such as application-specific integrated circuits

(ASICs), MEMS sensors, and power management systems, while ensuring sustainable and high-quality production. It collaborates with a wide range of clients, including automotive manufacturers (Tesla, Bosch), consumer electronics companies (Apple, Samsung), industrial enterprises (Schneider Electric), and telecommunications players (Huawei). Thanks to its expertise and partnerships with startups, universities, and major corporations, STMicroelectronics addresses the needs of the most demanding technology sectors, consolidating its leadership in the global semiconductor market.

1.2 Work environment

1.2.1 PINNs

Our project focuses on the experimentation and performance characterization of neural network learning by leveraging Physics-Informed Neural Networks (PINNs). These networks incorporate physical laws, such as differential equations and the fundamental principles of electrical circuits, to improve their convergence and efficiency with limited data.

Neural networks are computational models inspired by the functioning of the human brain. They are composed of layers of interconnected neurons, where each neuron receives an input, performs a transformation, and transmits the output to subsequent neurons. A classical neural network is divided into three parts:

- `The input layer:` Receives data such as images, text, or sounds.
- `One or more hidden layers:` Processes the data to perform specific tasks, such as classification or prediction.
- `The output layer:` Produces the final result.

During training, the network adjusts its weights and biases using optimization algorithms like backpropagation, minimizing a loss function that measures the discrepancy between the obtained results and the expected results. For example, ChatGPT is a large neural network with billions of parameters.

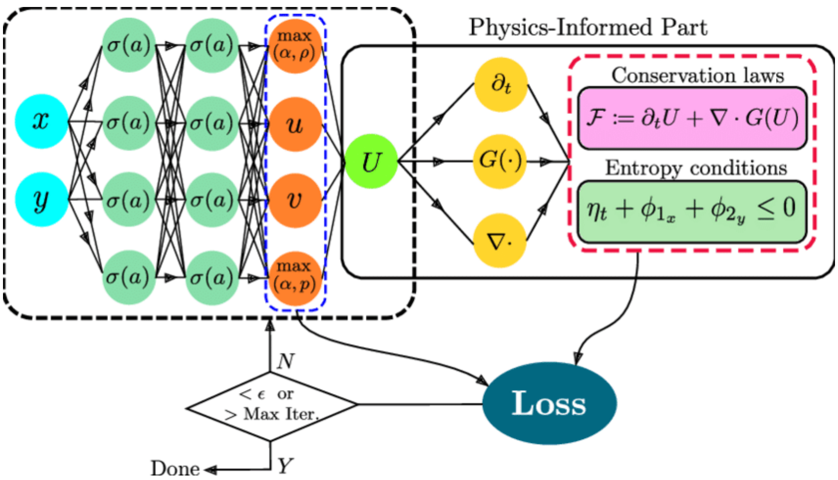

Figure 2: Schematic representation of PINNs for Euler equations

Physics-Informed Neural Networks (PINNs) are an extension of traditional neural networks. They incorporate physical laws in the form of differential equations directly into the learning process. Unlike traditional Deep Learning approaches, which rely primarily on empirical data, PINNs impose physical constraints using regularization terms in the loss function. This improves model convergence and enables the resolution of complex problems, such as fluid simulation, thermal modeling, and physical system dynamics, even when the available data is limited.

Integrating physical equations into PINNs creates a bridge between two modeling paradigms: data-driven modeling and physics-driven modeling. By leveraging the flexibility of neural networks to approximate solutions while adhering to physical laws, PINNs offer a powerful alternative to traditional numerical methods, such as the finite element method. This hybrid approach opens new opportunities in fields such as meteorology, engineering, and biology, enabling the resolution of problems where data is scarce or difficult to obtain.

1.2.2 Electronic filters

Electrical filters consist of components such as resistors, inductors, and capacitors. Among these filters, examples include RC (Resistor-Capacitor), RL (Resistor-Inductor), and RLC (Resistor-Inductor-Capacitor) filters.

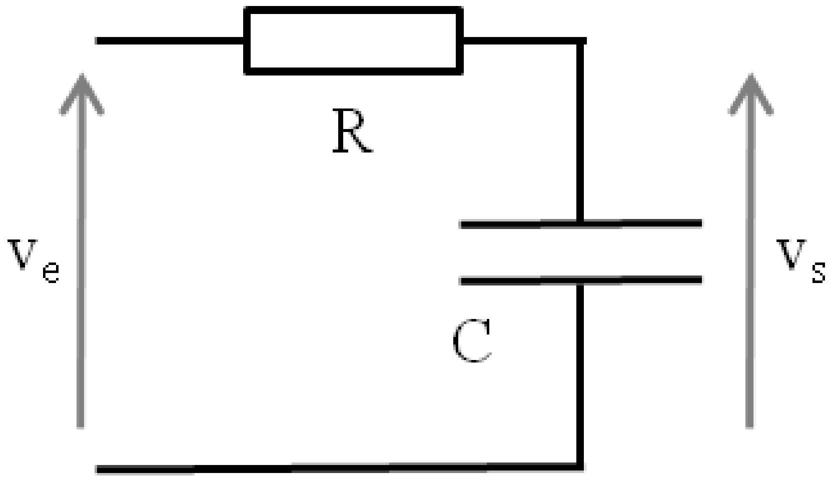

Figure 3: Diagram of an RC Circuit

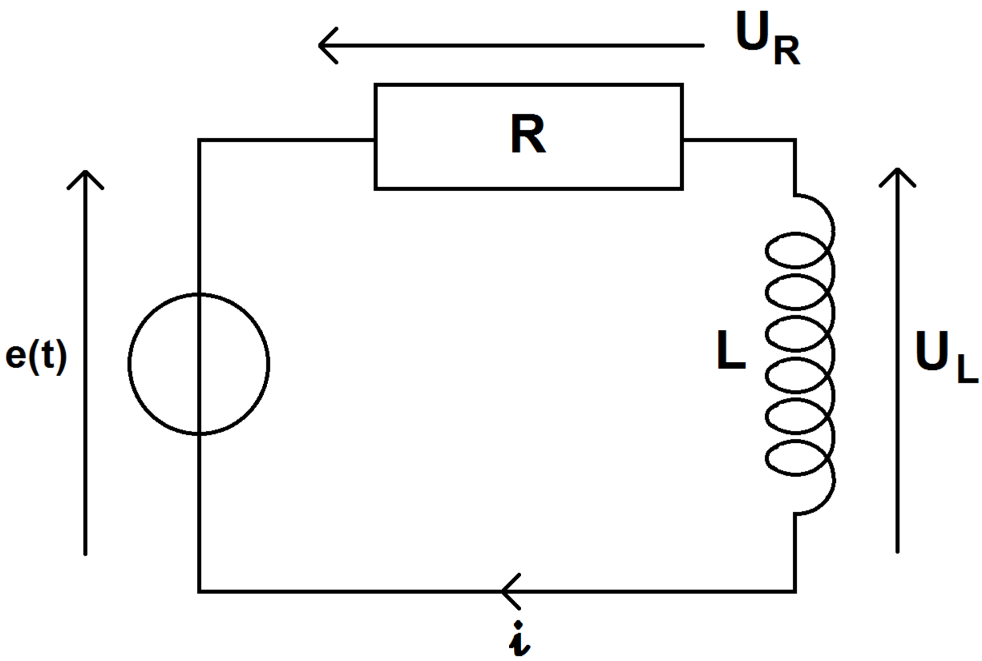

Figure 4: Diagram of an RL Circuit

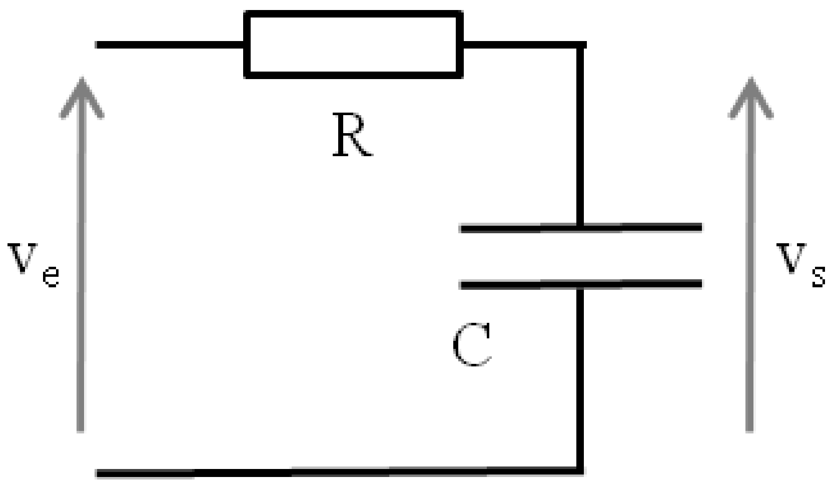

Figure 5: Diagram of an RLC Circuit

Each electrical circuit can be modeled using fundamental physical laws:

- `Kirchhoff's Voltage Law (KVL):` States that the sum of voltages around a closed loop is zero.
- `Kirchhoff's Current Law (KCL):` States that the sum of currents entering and leaving a node is zero.
- `Constitutive Equations:` These relate the variables of components (resistors, capacitors, inductors) through the following relationships:
  - Ohm's Law: $V = R \cdot I$
  - Capacitor: $I = C \cdot \frac{dV}{dt}$
  - Inductor: $V = L \cdot \frac{dI}{dt}$

These relationships give rise to differential equations that describe the dynamic behavior of circuits over time.

2 Goal

2.1 Main goal

The main objective of this project is to replace traditional simulators such as SPICE with neural networks capable of efficiently predicting the performance of electrical circuits (e.g., consumption, gain margin, phase margin, etc.). Current simulators, while accurate, suffer from very long computation times and increasing inefficiency as the number of conductors in modern circuits grows.

2.2 Constraints and Limitations of Traditional Methods

Traditional methods such as SPICE present several major limitations:

1. `High Computational Cost:` Simulations require a significant amount of time and resources, especially for complex circuits containing a large number of components and conductors. As the size and complexity of circuits increase, the simulation time grows exponentially, making this approach less viable for large systems.

2. `Limited Scalability:` Each parameter variation requires a new simulation, which can lead to a multiplication of tasks and increased computational complexity.

3. `Solution divergence:` SPICE can encounter convergence issues in complex non-linear circuits, particularly when initial conditions are poorly defined or the differential equations exhibit discontinuities.

4. `Dependence on Hardware Resources:` Running complex simulations requires powerful machines, which increases infrastructure costs.

These limitations make traditional simulators inefficient for the modern needs of fast and precise circuit optimization.

2.3 Why PINNs ?

The proposed approach combines machine learning methods with Physics-Informed Neural Networks (PINNs) to address these limitations. PINNs allow for the direct integration of physical laws (such as differential equations or transfer functions) into learning models.

This combination is particularly suited for:

- Reducing the number of simulations needed to optimize circuits.
- Improving data efficiency by decreasing the need for large amounts of training examples.
- Accelerating performance predictions while maintaining high accuracy.
- Leveraging physical knowledge to improve model convergence and efficiency.
- Optimizing electrical circuits faster to assist engineers in working on complex modern circuits.
- Reducing simulation tasks through predictive models trained on existing data.

By replacing traditional simulators with neural networks, this project aims to provide engineers with a faster, more efficient, and better-suited tool for complex modern circuits.

2.4 Optimizing SPICE Functionality with PINNs

Physics-Informed Neural Networks (PINNs) can improve SPICE by addressing its limitations:

1. `Faster Simulations:` PINNs reduce computation time for complex circuits by generalizing solutions after training.
2. `Better Convergence:` They more easily solve nonlinear circuits, overcoming divergence issues faced by classical methods.
3. `Advanced Modeling:` PINNs better capture the complex behavior of modern components (e.g., FinFETs, advanced materials).
4. `Efficient Parametric Analysis:` Once trained, they can quickly predict circuit behavior for different parameter values.
5. `Multi-Physics Circuits:` PINNs simultaneously handle electrical, thermal, and mechanical effects.

2.5 Deliverables

1. `Database Creation:` Generation of training and validation data via SPICE for various circuit scenarios.
2. `Data Efficiency Improvement:` Reduction of the need for large datasets by integrating physical laws into the models.
3. `Extrapolation Capability:` Development of networks capable of accurately predicting beyond the provided data.
4. `Loss Minimization:` Achieving minimal loss to ensure accurate predictions and stable convergence.
5. `Partial Replacement of SPICE Simulations:` Networks capable of predicting circuit performance faster than SPICE while maintaining comparable accuracy.

3 Project ressources

3.1 Supervisors

- `Nathan Chanez` : Doctoral Engineer at STMicroelectronics.
- `Marek Bucki` : CSO - Co-founder of TwInsight & Part-time Professor at UGA

3.2 Scientific literature

Physics-informed Neural Networks: a simple tutorial with PyTorch, Theo Wolf[1]

The article presents a straightforward approach to leveraging neural networks by introducing physical constraints to guide the network's learning using PyTorch.

It also includes an example of the cooling rate of a cup of coffee. The results are represented in the form of a curve, which clearly visualizes the data provided to the neural network as well as the curve representing predictions generated by AI.

3.3 SPICE

SPICE, or Simulation Program with Integrated Circuit Emphasis, is an open-source software for simulating electronic circuits. It facilitates the characterization of electronic filters, mainly used to analyze temporal and frequency behaviors by solving the differential equations governing the studied circuit.

Example of SPICE Simulator: NgSpice

Advantages :

- Using a simulator allows for comparison between the theoretical results of the filter and the version generated by the neural network.
- SPICE generates specific data such as voltage, frequency, etc., which can be used to train the neural network.

Limitations : SPICE remains relatively time- and resource-intensive.

3.4 Programming environment: Python

3.4.1 Specific libraries

- `PySpice` :

  PySpice is a Python library that interfaces with the SPICE tool (Simulation Program with Integrated Circuit Emphasis) to model, analyze, and visualize electronic circuits. PySpice allows users to define electronic components such as resistors, inductors, transistors, and generators, enabling the definition of entire circuits. Simulations of these circuits can be conducted using various types of simulations, such as DC (direct current) or AC (alternating current) circuits. After simulating the circuit, PySpice provides tools to visualize the results, including integration with

[1] Link to the article

matplotlib for plotting graphs and curves, which are instrumental in circuit analysis.

PySpice also serves as an interface to SPICE simulators and is particularly useful for analyzing electronic filters.

- `PyTorch :`

  PyTorch is an open-source Python library for machine learning, developed by Meta. It enables complex and precise computations essential for deep learning.

3.5 Data

One of the most critical resources for a neural network is the dataset used for training. In the context of this project, the dataset is generated by simulating circuits using the SPICE tool. The data collected from these simulations will be fed into the neural networks.

3.6 Machines

- `Physical mesurements:` Machines for testing filters (e.g., oscilloscope, VNA, etc.): Since the ultimate goal is to calculate loss precisely, access to exact voltage and current values within the circuit is essential. This will allow for an estimation of the error rate between the approximation generated by the neural network and the theoretical value to be calculated.

- `Processing and Computation:` A high-performance computer and analysis/simulation software: Calculations will be executed by neural networks, performing intermediate computations in hidden neurons. The output will be an approximation of the solution to the differential equations governing the circuit.

The obtained data will then be interpreted using additional libraries like matplotlib, which will assist in visualizing these data through graphs.

4 Project management

As part of our project, we chose to apply the Scrum methodology for agile and efficient project management. The two-month project will be organized around weekly sprints, supported by structured daily and weekly meetings with Trello as the tool to manage tasks.

The Scrum method is based on sprints, which are short, time-boxed periods during which selected tasks that add value to the project are completed. At the end of each sprint, a review of the work is conducted, and the next sprint is planned to increment on the previous one while adjusting for any mistakes made.

For our project, we will do one-week sprints because the project is only two months and needs frequent tracking.

We will hold one-hour weekly meetings, during these meetings, each team member will give a 5 minute presentation of their previous sprint, explaining the actions they completed or didn't complete and issues they encountered, so we can make adjustments for the upcoming sprints. Then, we will plan the next sprint and assign the various actions to each team member.

An action consists of three elements:

- A description formulated with an action verb
- A deadline
- One or more team members responsible for the action

In order to manage the actions, we will use Trello, an application on which we have a dashboard containing five lists :

- One for the backlog, by default all actions start in the backlog
- One for actions to be completed during the sprint
- One for actions in progress
- One for blocked actions
- One for completed actions

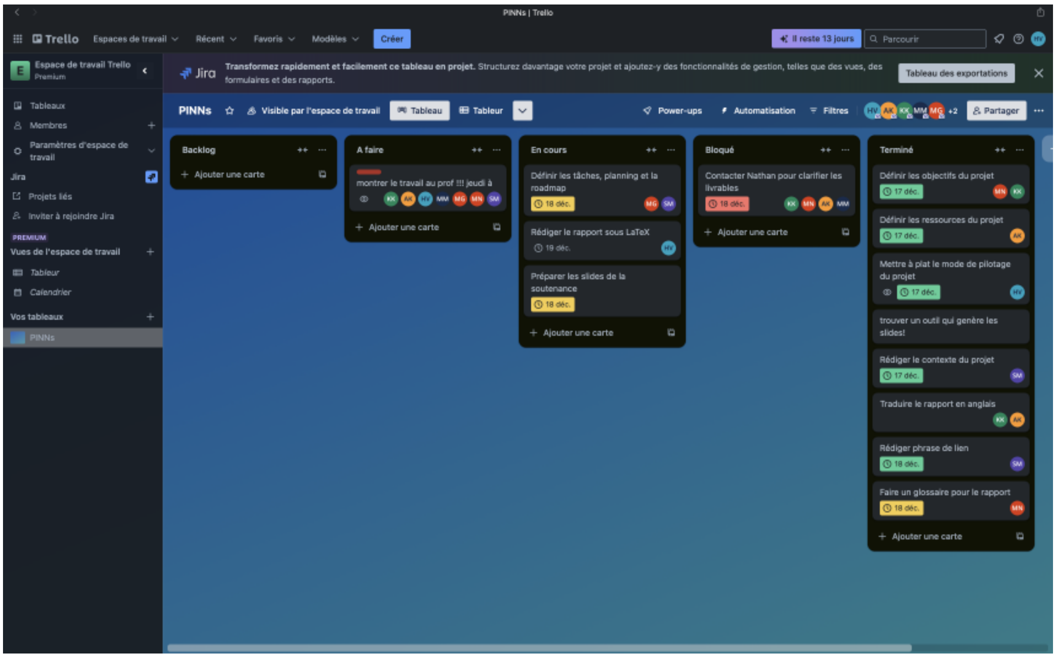

Figure 6: Action list on Trello

Trello allows us to assign actions to one or more team members and to add a deadline. For an efficient project tracking, it is essential that each action is on Trello and includes the three attributes defined earlier. Each team member manages their actions and moves them between columns as the project progresses.

In addition to the weekly meetings, we will hold 10-minute daily meetings every morning at 9 AM. The purpose of these meetings is to review what each team member did the previous day and what they plan to do that day. This way, each team member has a clear overview of the sprint's progress. Each day, a different team member will lead the daily meeting. Their role is to keep the meeting on track, ensuring it stays brief and does not turn into a problem-solving session.

This Scrum organization, centered on short sprints and daily communication, will enable smooth and responsive project management while fostering collaboration and continuous improvement.

5 Project roadmap

To ensure a structured and efficient progression of the project, a detailed plan has been developed. We divided the work into two main phases, each with clear objectives and well-defined deadlines. A Gantt chart was used to visualize and organize the major tasks and key milestones.

Phase 1

- `When:` From December 16^{th} to 20^{th}, 2024.
- `Main goal:` Right a preliminary report and presentation.

This first phase aims to establish the bases of the project, clarify expectations and prepare a preliminary presentation. The following gantt diagram shows this organisation.

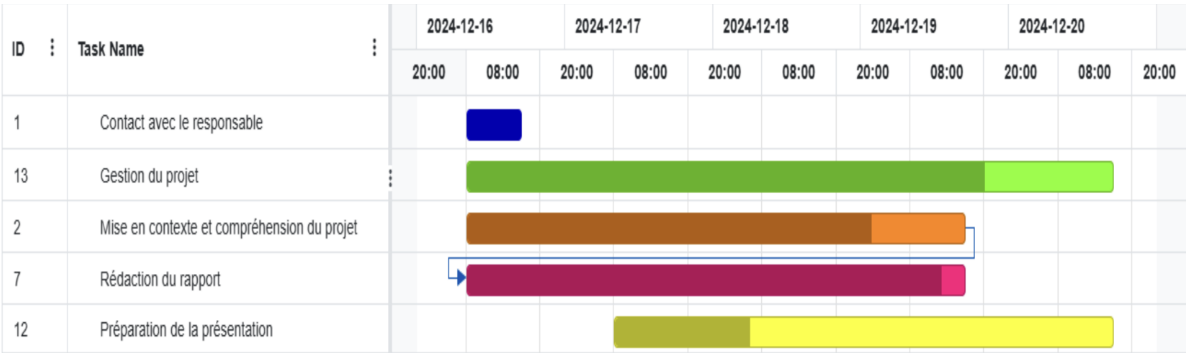

Figure 7: Gantt diagram for phase 1

Phase 2

- `When:` From April 14^{th} to June 9^{th}, 2025.
- `Main goal:` Finalize the project and achieve the set objectives.

La seconde phase, d'une durée de deux mois, nous permettra de se concentrer sur les tâches techniques et de finaliser le projet. Les tâches ont été planifiées de manière à équilibrer les efforts entre développement, tests et documentation.

The second phase of two month will allow us to focus on technical tasks and finalize the project. Tasks were planned in order to balance effort between development, tests and documentation

Key actions:

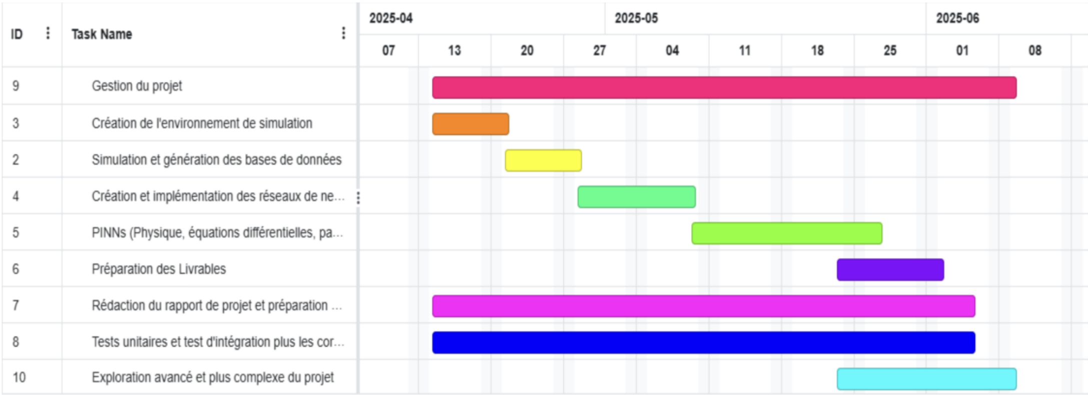

Figure 8: Gantt diagram for phase 2

Alignment of Actions

- `Project management:` It spans the entire duration of the project as this task encompasses overall project monitoring and progress, adjustments, and validations.
- `Creation of the simulation environment:` It concludes fairly early in the project. However, this task may require an update or extension.

Glossary

`SGS` : Società Generale Semiconduttori
`MEMS` : Micro Electronic Mechanical Systems
`IoT` : Internet of Things
`RF` : Radio Frequency
`NFC` : Near-Field Communication
`ASIC` : Application-Specific Integrated Circuit
`PINNs` : Physics-Informed Neural Networks
`SPICE` : Simulation Program with Integrated Circuit Emphasis
`VNA` : Vector Network Analyzer

Appendix

Sources

`Figure 1` : st.com
`Figure 2` : researchgate.com
`Figure 3` : courselectronique.wordpress.com
`Figure 4` : methodephysique.fr
`Figure 5` : methodephysique.fr