RotateAI Simulator Outline

Maxence Morel Dierckx

February 14, 2026

1 Problem Statement

RotateAI inference pipelines must run on STM32U5 microcontrollers for multi-month whale tag deployments at ultra low power. To design efficient programs, we need the ability to validate that a pipeline:

1. Produces acceptable error in outputs.
2. Meets resource constraints (space, time, power).

The proposed simulator is a desktop tool that tests pipelines in the form of C binaries against real sensor data without requiring embedded hardware.

To enable rapid testing and iteration, the simulator must be decoupled from the pipelines, allowing any conforming binary to be tested with any input data.

2 Functional Requirements

2.1 Inputs

The simulator will accept any quantity of MAT files with columns labelled:

- a_x, a_y, a_z (accelerometer).
- m_x, m_y, m_z (magnetometer).
- p (pressure).
- aw_x, aw_y, aw_z (ground truth accelerometer).
- mw_x, mw_y, mw_z (ground truth magnetometer).

The simulator will accept one reference to any compiled C binary.

The simulator will accept a configuration file in the TOML format providing:

- Data sample rate in Hertz.
- Microcontroller voltage.
- Microcontroller Dhrystone Million Instructions Per Second (DMIPS) per MHz.
- Minimum frequency safety margin.

2.2 Execution

The simulator will run a separate simulation for every MAT file and output a separate report for each.

A simulation will:

- Run the binary.
- Feed data to the process sample by sample.
- Read data of the same shape from the process.

2.3 Benchmarking

The simulator will measure the following metrics:

- **File size** of the program in bytes.
- **Peak memory usage** in bytes.
- **Average instructions** per inference.
- **Floating-point Operations** per inference.

The simulator will derive (estimate) the following metrics using configurable constants:

- **Minimum operating frequency** in Hertz.
- **Energy per inference** estimate in Joules.
- **Duty cycle** as a percentage.
- **Power consumption** in Watts.

The simulator will validate the following metrics:

- **Absolute error** from the ground truth in sensor units (A_w in g, M_w in μT).

2.4 Outputs

The simulator will produce one JSON file for every simulation.

Each JSON file will contain all of the stated metrics in human-readable format, with units labelled.

3 Non-Functional Requirements

The platform should be Linux only with Python 3.10+.

The simulation should take less than 100 times the real dataset duration to complete.

Identical inputs should produce identical outputs.

4 Deliverable

Version 1.0.0 of the proposed simulator should be delivered as a repository on github.com/mtmd1 by 26/02/2026 at the latest.

It should include clear instructions in the form of a README documenting the entire installation process and intended workflow.

It should also contain a PDF document detailing how the derived metrics are estimated.

5 Success Criteria

The simulator will be considered successful when:

- It runs without error on at least one provided test dataset.
- It produces a valid JSON file containing all specified metrics.
- Measured metrics (file size, instruction count) match independent verification (e.g., `stat`, `perf stat`).
- Derived metrics match hand calculations using the same input constants.
- A known-correct pipeline produces near-zero error; a known-incorrect pipeline produces significant error.