Potato: A Data-Oriented Programming 3D Simulator for Large-Scale Heterogeneous Swarm Robotics

Jinjie Li[1], Liang Han[2], Haoyang Yu[2], Zhaotian Wang[2], Pengzhi Yang[3], Ziwei Yan[2], Zhang Ren[1]

Homepage: https://jinjie.li/

[1]School of Automation Science and Electrical Engineering, Beihang University
[2]Sino-French Engineer School, Beihang University
[3]Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology

Abstract

- Existing multi-agent simulators → Object-Oriented Programming (OOP) → CPU
- This simulator:
  - homogeneous agents → Data-Oriented Programming (DOP) → GPU
  - heterogeneous agents → multi-process
- The simulator is developed using PyTorch, and is further accelerated using TorchScript, a tool from AI community for model deployment
- Simulating 5,000 nonlinear quadrotors for 1 round → less than 2ms

Methodology

- OOP: Agents are computed through for-loops, multi-threads, or multi-processes. However, since a desktop CPU has 10-20 threads, each CPU thread computes multiple agents serially in a loop. → computational speed increases almost linearly with the number of agents
- DOP: grouping the computations of homogeneous agents together and parallelizing them in batches using tensors, which can be computed directly on GPUs. → computational speed remains almost the same with the number of agents

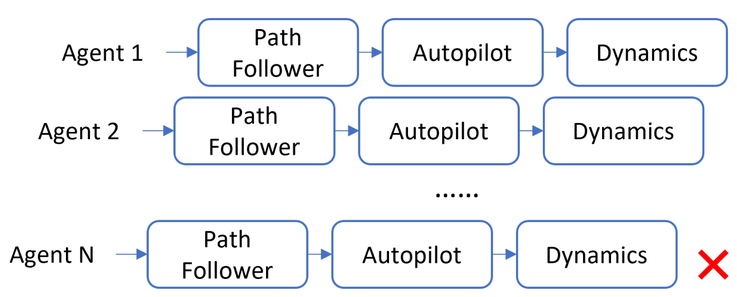

Fig. 1. Object-Oriented Programming (OOP)

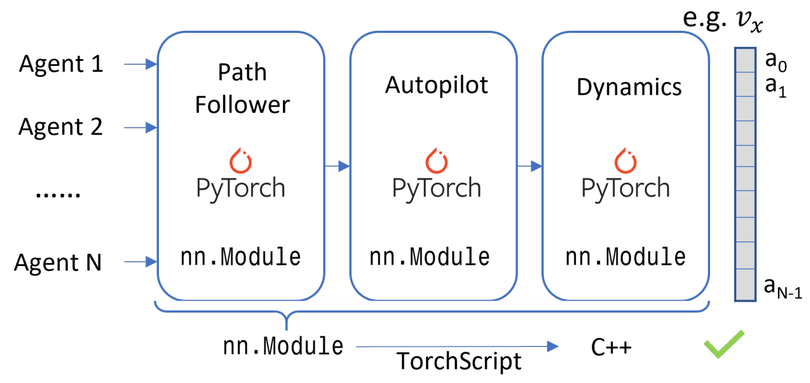

Fig. 2. Data-Oriented Programming (DOP)

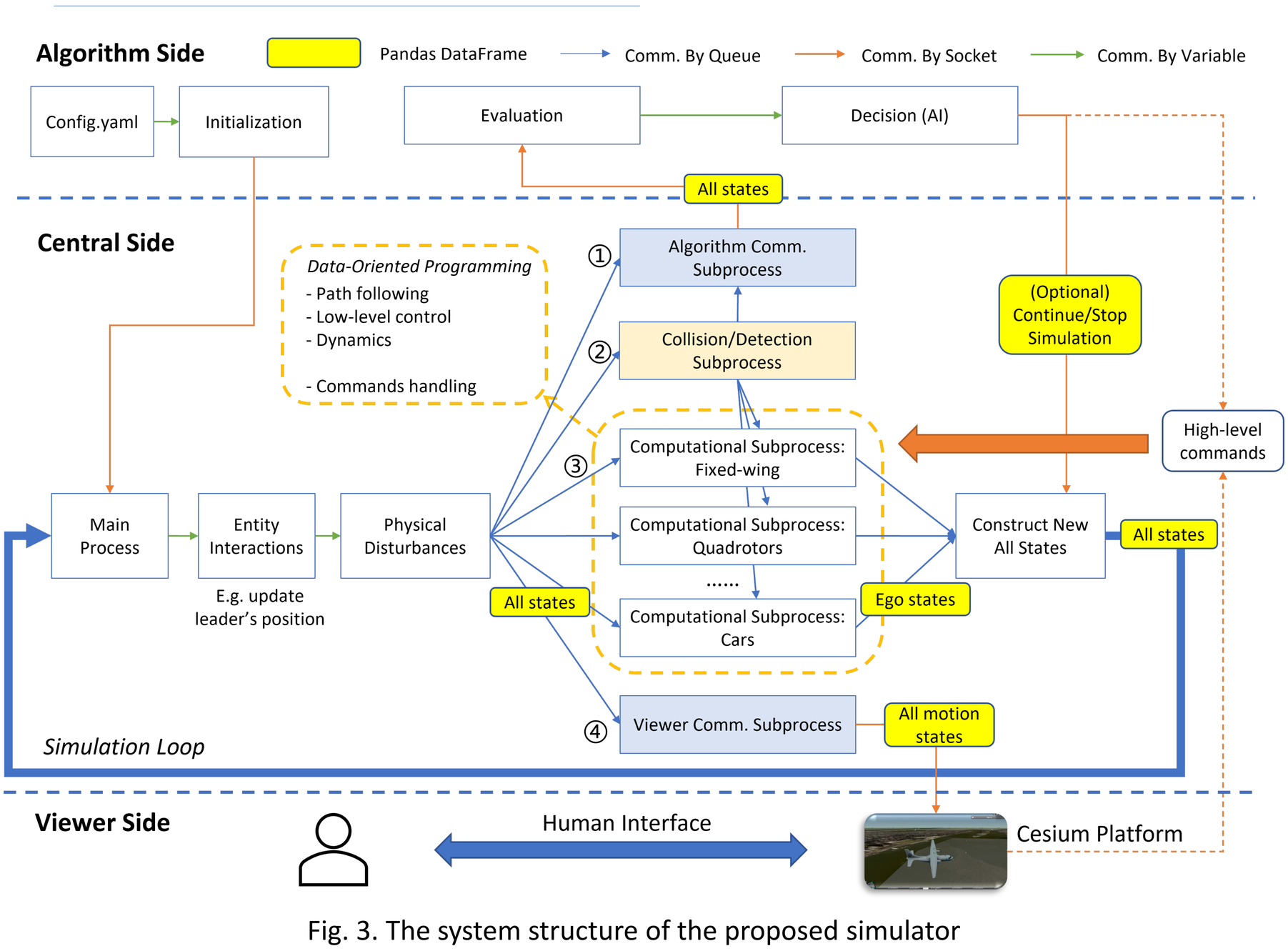

Fig. 3. The system structure of the proposed simulator

The proposed simulator consists of a *Simulation Loop* where all agents' states are transmitted to 4 directions.

- Direction ① sends the states to the *Algorithm Side* via an algorithm communication subprocess, which uses this information for evaluation and decision making.
- Direction ② calculates collision and detection results, which are sent to computational subprocesses for handling.
- Direction ③ computes low-level algorithms and dynamics for heterogeneous agents, and sends the updated states back to the main process to refresh the all-states data.
- Direction ④ uses a viewer communication subprocess to visualize all agents' motions, and users can manipulate the mouse to influence the agents' behaviors.

Performance and Example

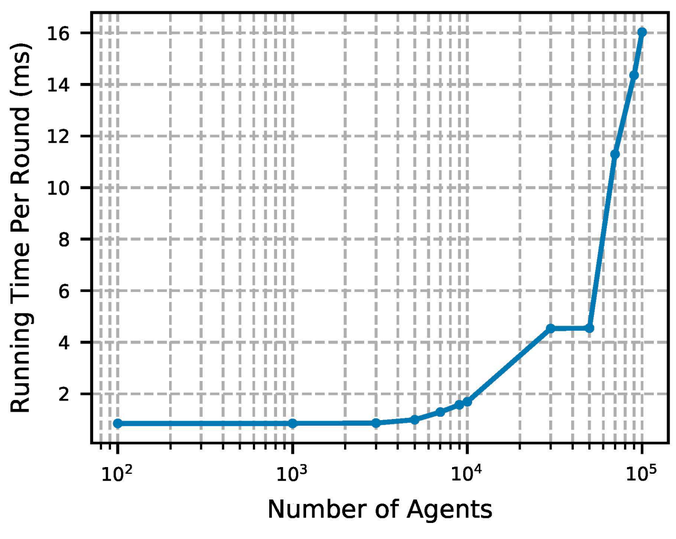

Fig. 4. Running time as a function of number of agents based on the log coordinates. The time remains stable under 5,000 quadrotors.

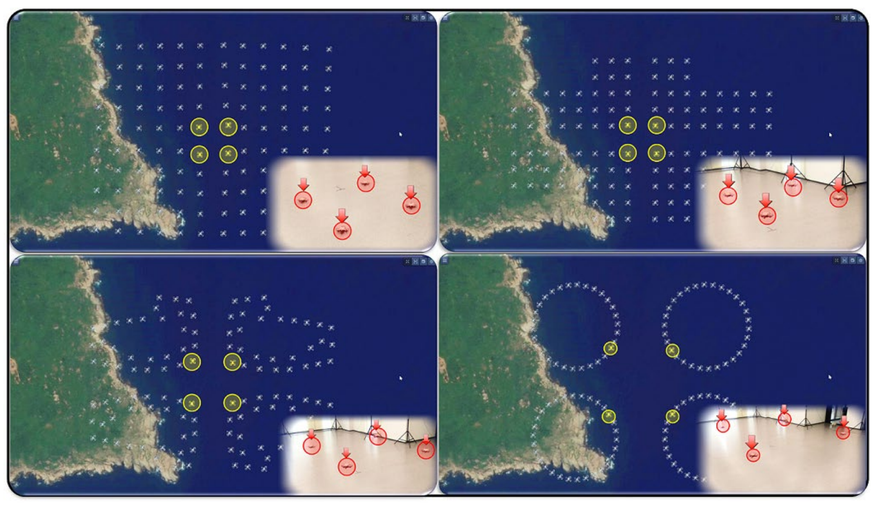

Fig. 5. Demo: one paper accepted by ICRA 2023 is supported by this simulator. Paper link: →

Conclusion

- Data-Oriented Programming is suitable for multi-agent simulation
- Written in PyTorch and accelerated by TorchScript is a great balance of flexibility and efficiency

Get the paper!

- Authors cannot attend the conference because of the visa problem
- Please email us at lijinjie@buaa.edu.cn if you are interested!
- Zoom: 784 0450 7759 PSW: 4yHsGy