General context

The goal of this project is to develop a reinforcement learning environment for a multirotor drone using Isaac Lab, with the objective of training control policies for basic tasks such as stabilization and position control.

Initial approach

The original approach was to build a custom drone environment from scratch in Isaac Lab. A USD model of a quadrotor was used, and a custom environment was defined, including observations, actions, and reward functions, integrated with SKRL for training.

First attempt

The first attempt focused on controlling the drone as a rigid body by directly applying forces or velocities. This approach failed because the required action configurations are not available in the installed version of Isaac Lab, leading to import errors and API mismatches. This approach was therefore abandoned.

Second attempt

The second attempt involved converting the drone into an articulated system by adding joints to the propellers and controlling them through joint velocities. Here, the main issue was severe API instability across Isaac Lab versions. Many classes and arguments referenced in examples or documentation no longer exist in the current API, resulting in continuous compatibility issues.

Third attempt

In the third phase, the project was executed using different setups: Isaac Lab via pip, the Python environment bundled with Isaac Sim, and mixed configurations. This led to persistent errors related to missing `omni` modules and version conflicts between Isaac Sim, Isaac Lab, and SKRL. As a result, the training process could not be initialized reliably.

Key issues identified

Across all attempts, several structural issues were identified:

- Significant API instability in Isaac Lab across versions.
- Misalignment between available examples and the actual installed environment.
- Strong dependency on internal Isaac Sim modules not accessible via pip.
- Excessive complexity when starting directly with a full multirotor system.

Proposed next steps

The recommended direction is to simplify and restructure the approach. The main options are:

- Start from a stable, official Isaac Lab environment and progressively adapt it.
- Use an existing drone implementation already available in Isaac Sim.
- Decouple the problem by validating classical control first, then introducing reinforcement learning.

Conclusion

In conclusion, the work completed so far has been valuable in defining the practical limits of the current technical stack. Moving forward, the priority should be stability and reproducibility of the environment before increasing system complexity.