

1 correction

The project's initial goal was to achieve walking for the bipedal robots of the Bold Hearts team. To achieve this, the problem has been split into three major stages. The first stage solved the cartpole environment, a classic reinforcement learning problem. This stage allowed to test, in practice, different learning algorithms and implementations on a simple problem. This stage also allowed to develop a base structure for the rest of the development, including a logging system.

The second stage consists of developing a 2D environment with a simplified humanoid, allowing to increase the complexity and approximate the development's target. This task helped to better understand the Keras API and neural networks as it required research to understand how to output eight simultaneous actions. This stage required the adaptation of the learning algorithm as it now needs to control eight joints simultaneously. To successfully develop this stage required the research of physics engines and visualizers, understanding how this works and future usage possibilities. The decision to use pygame and Pymunk was based on how well they integrate, as pygame can display the objects created directly on pymunk without duplication. Also, pymunk was chosen over Box2d as it has python bindings that are no longer maintained and will lose compatibility with new python versions.

The third stage consisted of developing a 3D environment of the Bold Hearts robot using Mujoco. This stage has been modified as using Mujoco would require developing a new control interface for the robot, which would fall outside the scope given the time constraints. To achieve this stage, train the robot in a 3D environment, the development shifted to use gazebo and ROS as the team is currently using them. Using the team's current environment allowed for a better understanding of the current implementation of the software and a more in deep knowledge of ROS and custom environment creation on openAI.

At this moment, the first stage has been fully completed. A development structure has been created, including logging using Wandb, where all the metrics, code, and hyperparameters for each training session are logged and can be compared. During the development, two types of learning implementations have been tested. One is a high-level layer implementation of Keras agent-based learning, Keras-rl and the second implementation uses the Keras API directly, providing more control over the implementation. This has been chosen over the more straightforward implementation. While it was harder to comprehend and get good results, it helped develop the knowledge on learning algorithms and how they work and provide better control over

the experiment. The second stage achieved a fully working environment and learning algorithm, although a walking pattern has yet to be achieved. This is the result of such a complex problem and can originate from multiple variables such as the reward function, training time or other hyperparameters. Continuous iteration to tune this variables continues along with the last stage of development.

The third stage is currently in development, and as mentioned, it will be using the current implementation of ROS and Gazebo. The link between ROS and openAI will be based on the package *openai_ros*, developed for ROS1 and Bold Hearts uses ROS2. This stage also helped document and understand how to implement the team's environment in the architecture of the computer used in this development (ARM64). It required some workarounds due to compatibility issues and the need for a custom docker image. This will be helpful if other members switch platforms as the problems and solutions have been researched and documented.

While the initial target was to achieve full walking capabilities, the time and complexity constraints might limit how far this can be achieved. This project will nonetheless document the entire development, including the findings during training, an important achievement of this project is related to the last stage, the link between our robot environment and openAI Gym, allowing for easier and faster implementation of future attempts of reinforcement learning using the platform.