

DRLGyro

DRLGyro is the project folder for my semester project *Deep Reinforcement Learning to Control a Gyroscope* conducted during the 2020 spring semester at EPFL.

For more information, please contact me at: lecauchoismatthieu@gmail.com

Installation

The project highly depends on two frameworks, namely OpenAI's *Gym* and *Spinning Up* frameworks, which can be installed as such:

- For Gym:

```
pip install gym
```

- For Spinning Up:

```
git clone https://github.com/openai/spinningup.git cd spinningup pip install -e .
```

Files

The project folder consists of two main directories:

```
├─ code 'All code files and model files of the project'  
└─ report 'Report and pictures of the report'
```

The *code* directory contains the utilities, experiments and results of the project. Not all directories and sub-directories are shown here, such as the many model folders that were generated during the experimentation process, but the most relevant ones are. To understand what the many model experimentation folders correspond to, please refer to the notebooks where the experiments are performed.

```
├─ custom_functions 'Custom python library implementing useful functions for the project'  
├─ environment  
│   ├── Gyroscope 'Environment folder ready for import as a library'  
│   │   └─ gym_GyroscopeEnv  
│   │       └─ envs  
│   │           ├── gyroscopediscontinuous_env.py 'Environment with discontinuity'  
│   │           ├── gyroscope_env.py 'Simple environment'  
│   │           ├── gyroscopeintegral_env.py 'Environment with additionam integral term'  
│   │           └─ gyroscoperobust_env.py 'Robust environment'  
│   └─ gyroscope_environment_testing.ipynb 'Simulation code (python class) transferred to the Gym API'  
├─ simulation 'Initial simulation experiments'  
│   ├── AgramFrame.png  
│   └─ gyroscope_simulation.ipynb 'Simulation coding and experiments'  
├─ training_spinuplib 'Experiments using the spinning up library'  
│   ├── activations_visualization.ipynb 'Visualization notebook to cluster activations using t-SNE'  
│   ├── baseline_norm_ddpg_td3_sac.ipynb 'Testing notebook between DRL algorithms for norm. reward'  
│   ├── baseline_quadreward_ddpg_td3_sac.ipynb 'Testing notebook between DRL algorithms for quad. reward'  
│   ├── discontinuity_ddpg.ipynb 'Testing notebook for discontinuous environment'  
│   ├── envvariants_quadreward.ipynb 'Testing notebook with additional integral term'  
│   ├── paramsearch_iter0_norm_td3.ipynb 'Testing notebook for hyperparamater search iteration 0'  
│   ├── paramsearch_iter1_norm_td3.ipynb 'Testing notebook for hyperparamater search iteration 1'  
│   ├── paramsearch_iter2_norm_td3.ipynb 'Testing notebook for hyperparamater search iteration 2 and 3'  
│   ├── reward_testing_ddpg.ipynb 'Testing notebook for reward function analysis'  
│   ├── robust_agent_testing.ipynb 'Testing notebook for the robust agent'  
│   ├── simple_agent_testing.ipynb 'Testing notebook for the simple agent'  
│   ├── td3_n_it2_an_var1_extralay_extralay_s0 'Model folder (load w/ pytorch) of the final simple agent'  
│   └─ best_agent_robustified 'Model folder (load w/ pytorch) of the final robust agent'  
└─ training_udalib 'Preliminary experiments using the udalib library before moving to spinning up'
```