Deep Reinforcement Learning algorithms for autonomous systems

Prof. Pietro Michiardi - Eurecom
Prof. Elena Baralis - Politecnico di Torino

Report settimana 09/09/2019

Piero Macaluso

Contents

1 Problems and solutions

1.1 Stationary policy

During the training attempts made with the implemented algorithm, a marked difference emerged between the training and the test phases. The actions of the robot in the testing phase appeared to be slow, repetitive and, even after numerous training periods, the robot continued to behave in the same way.

The first attempt was to increase the learning epochs and manipulate the size of the batch extracted at each time. This choice did not allow to obtain the desired results: the intervals between one episode and the next one raised due to the increase in the previously indicated parameters, without leading to noticeable improvements. This change would have made even future experiments impossible due to the very long time spent in training.

For this reason, it was useful to proceed with a revision of the previous code used to solve positively the problem provided by the OpenAI Gym *Pendulum-v0* environment. In this context, the size of the system state image presented a resolution that was much smaller than the one used in the experiment with Cozmo.

The decision was, therefore, to reduce the size of the Cozmo state image from 320x240 to 64x64 pixels while maintaining a batch size of 256 samples and a proportion of 100 epochs/episodes. These changes have allowed not only to solve the problem related to the stationary policy but have drastically reduced the waiting times. This result enables us to perform experiments and to obtain plots in reduced time compared to the previous situation.

1.2 Save'n'Restore phase

The previous implementation offered a state saving phase (e.g. networks, episodes, memory) that could be started by the user who deliberately chose to suspend the experiment. This type of approach reveals a critical drawback. Although the connection between Cozmo, Tablet and PC is stable, a random disconnection sometimes happens: it leads to a sudden interruption of the program with consequent deterioration of the data collected so far. This event risked making hours of experiments vain.

In order to avoid further problems of this type, we implemented a checkpoint mechanism: the system saves the state of the experiment by creating a copy that can be useful to restart the procedure. This event triggers every X episodes. In the case of undesired interruption of

the program, the lost learning episodes will be paltry this time, and the experiment can restart from the last checkpoint.

1.3 First 500 episode experiment

Thanks to the improvements applied to the algorithm, it was possible to carry out the first experiment using the track. During the experiment, promising signals have finally appeared: Cozmo seems to learn, albeit in a prolonged way. The experiments were carried out on 500 episodes for a total of about 50000 learning epochs. The experiment also exploited alpha autotuning. The plot in fig. 1 shows that the algorithm improves and reaches a peak in episode 400: in that episode, Cozmo manages to cover about half of the circuit without crashing. A GIF image of this episode is available at: `https://drive.google.com/file/d/17wCT0SlqVdHrVQ4HTu3GU-Cz8Q7dFeqA/view?usp=sharing`.

The trend of the averages presented in figs. 3 and 4 on the next page and on page 4 shows an increasing average of the rewards with the passing of episodes. During the following week, we will carry out further experiments starting from this basis.

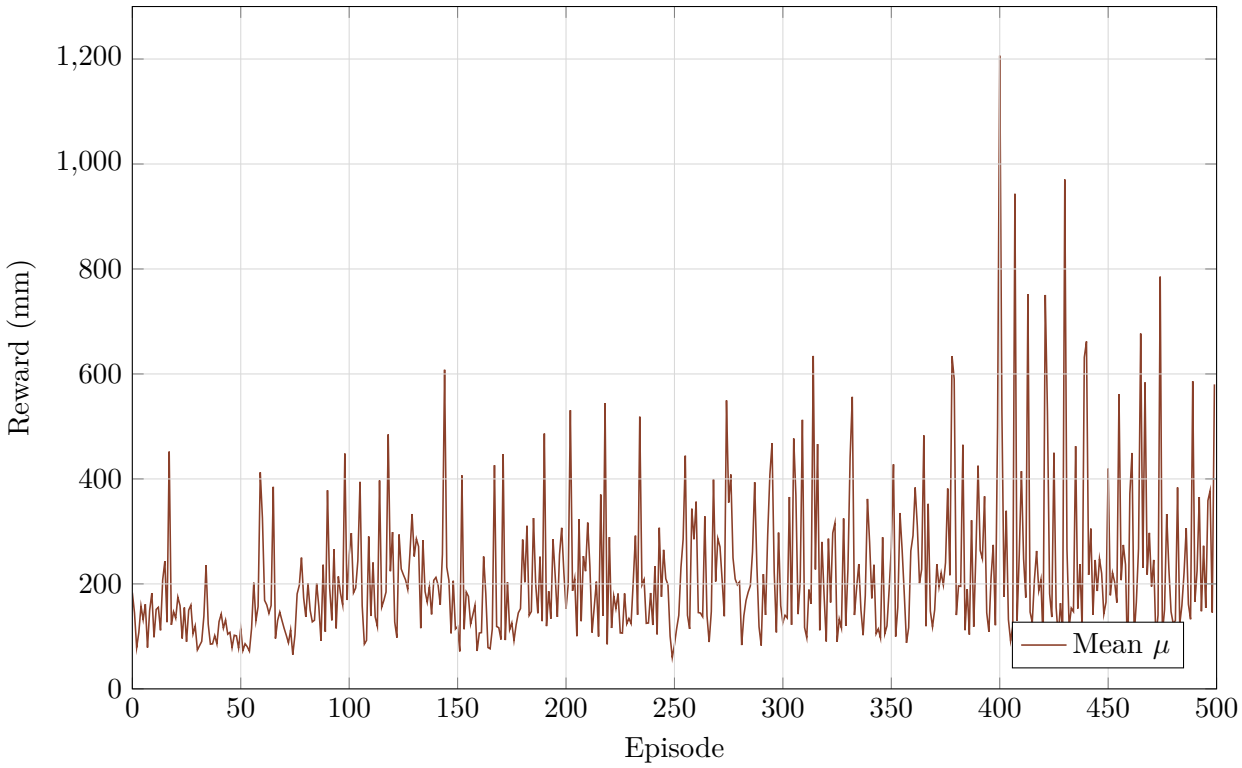

Figure 1: Total reward for each episode. The maximum value of about 1200mm is at iteration 400.

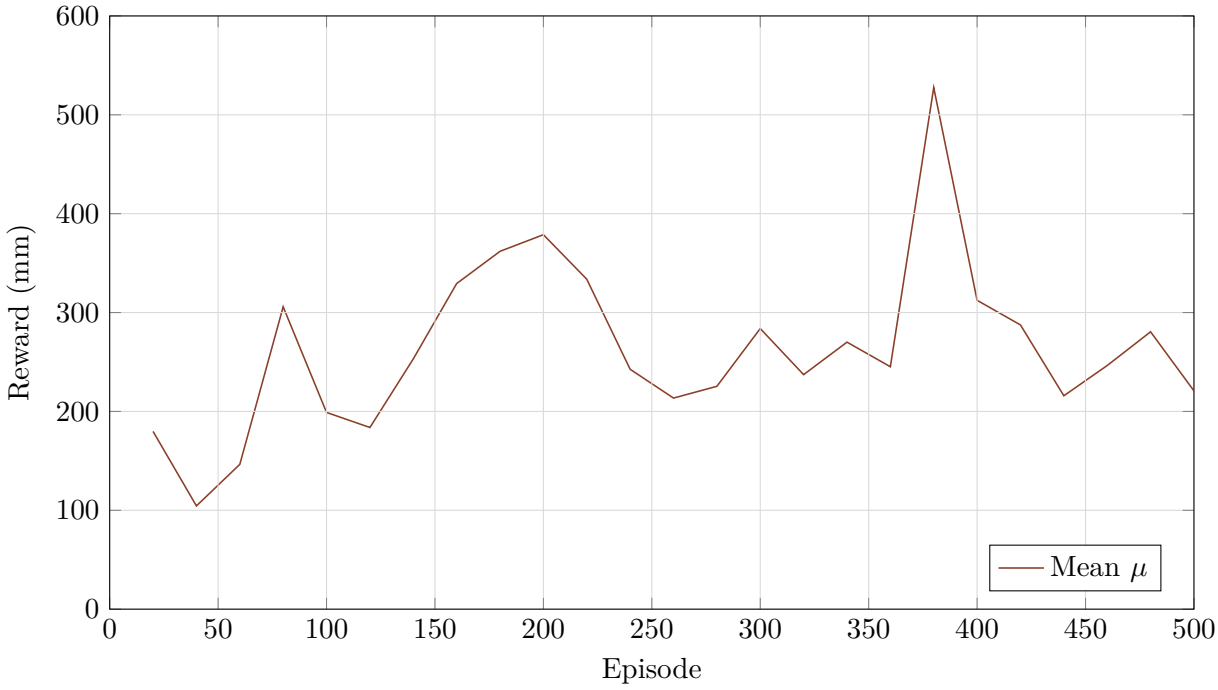

Figure 2: Mean reward of 5 episode of test. The maximum value is at iteration 380.

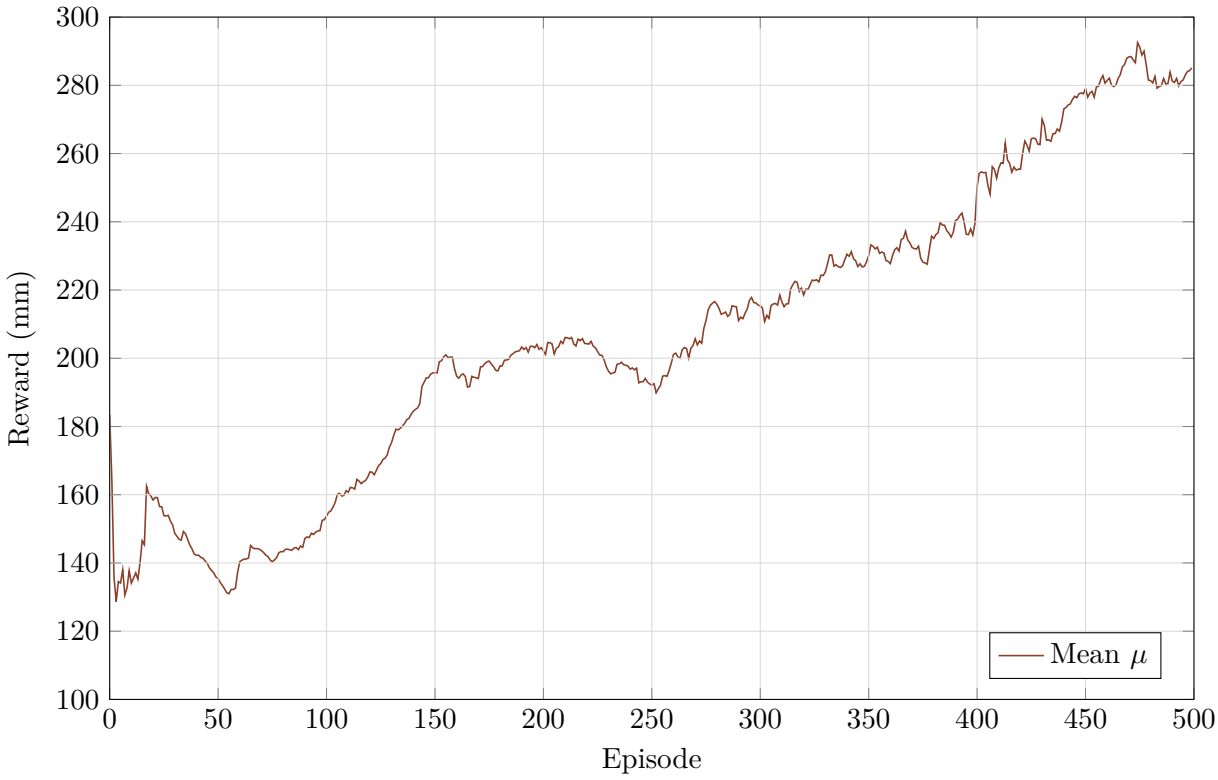

Figure 3: Mean reward of the last 100 episodes.

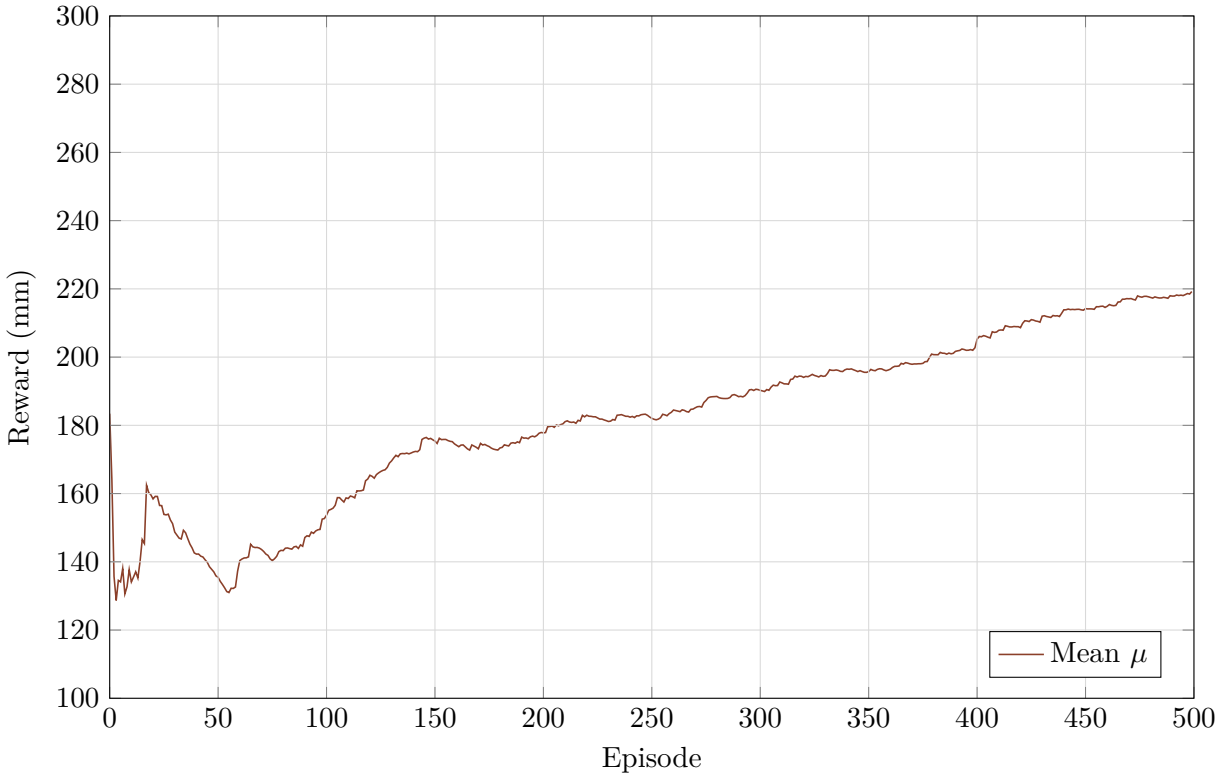

Figure 4: Mean reward of all episodes.

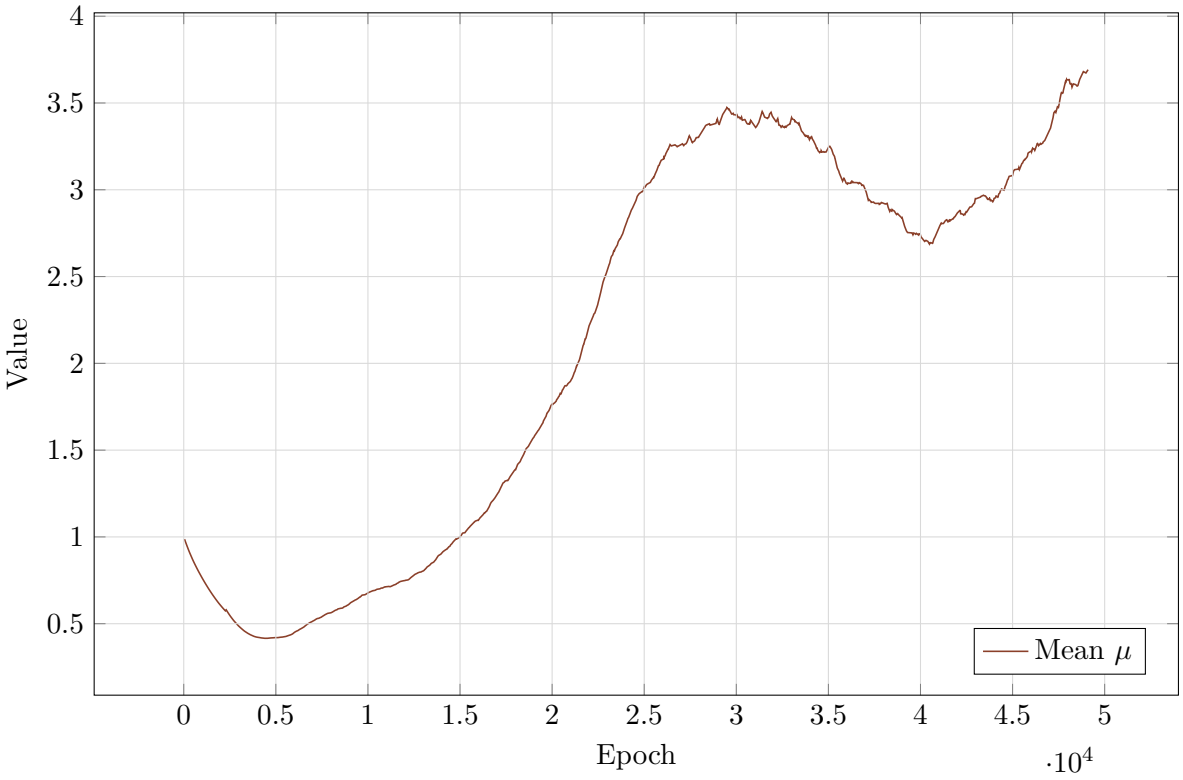

Figure 5: Value of alpha during the training

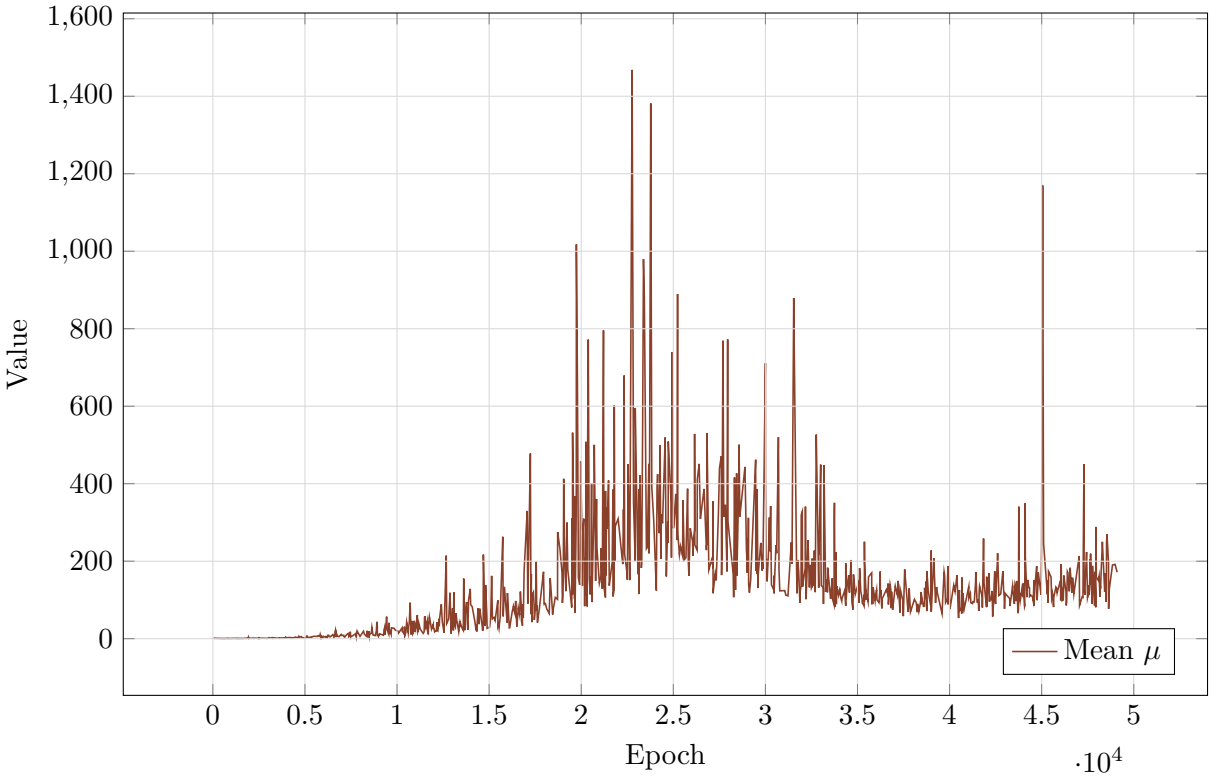

Figure 6: Critic 1 Loss

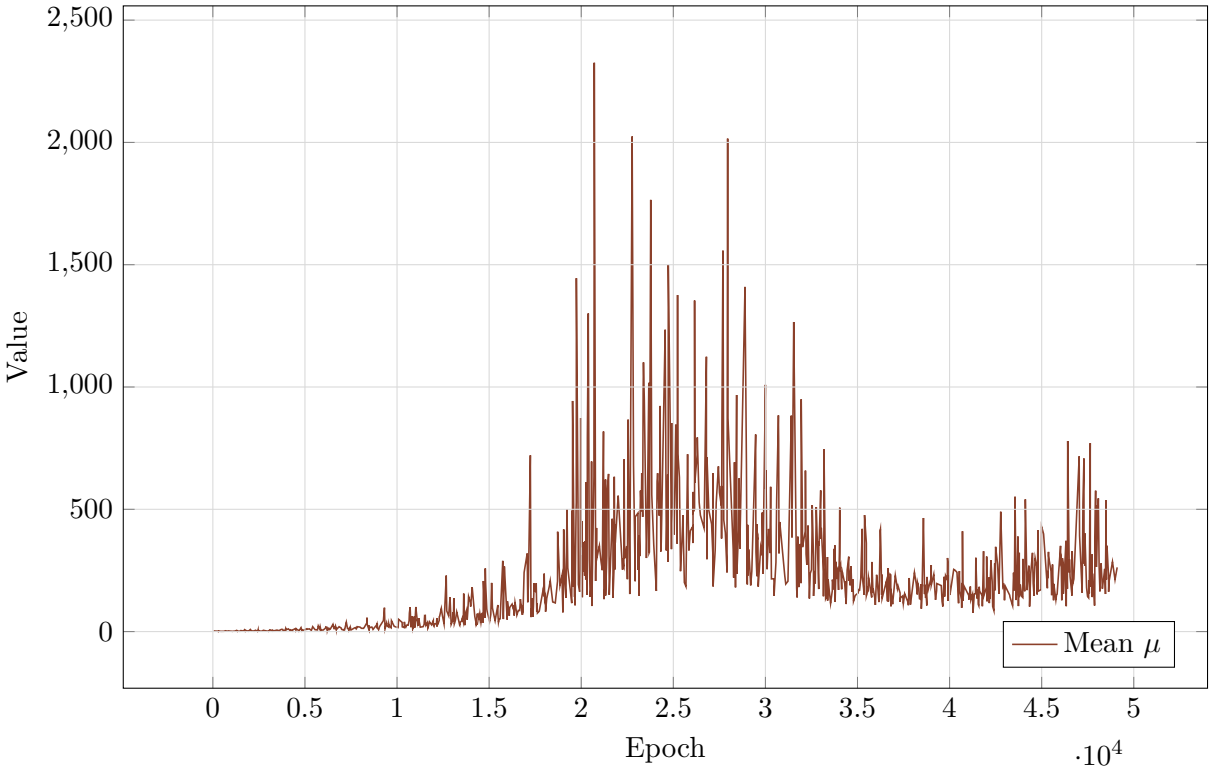

Figure 7: Critic 2 Loss

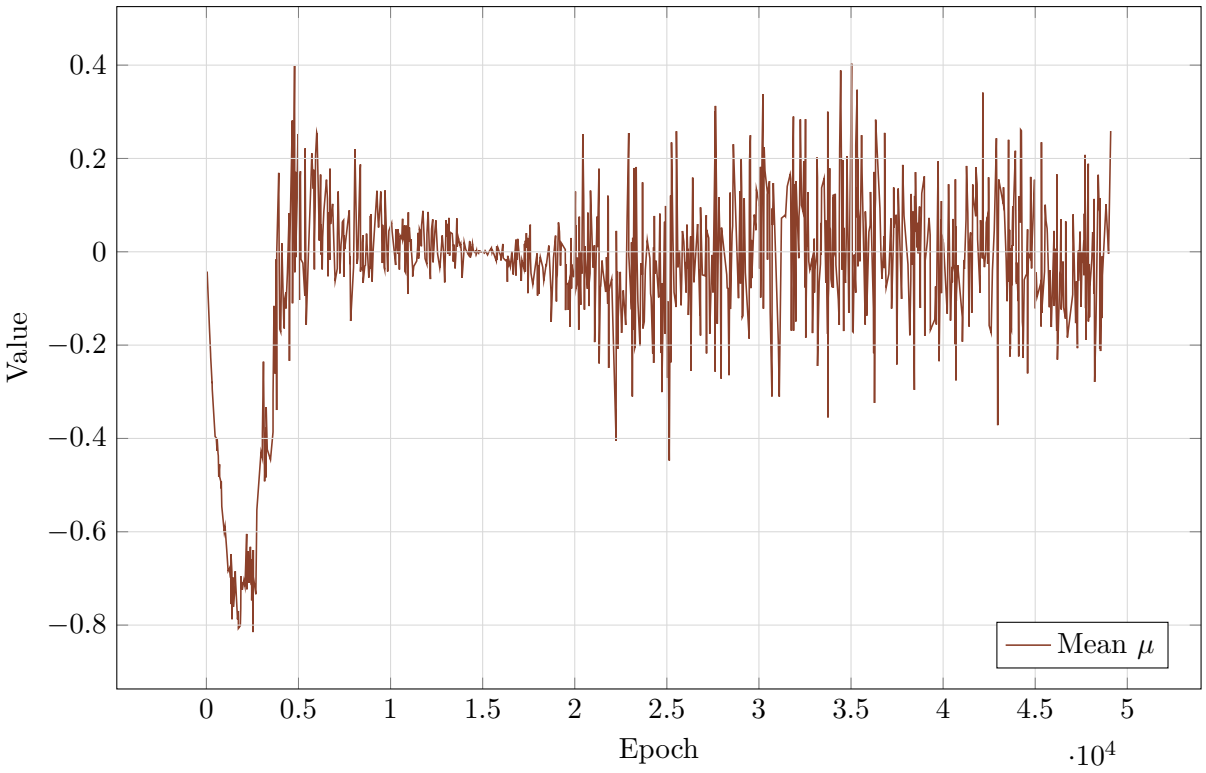

Figure 8: Entropy target Loss

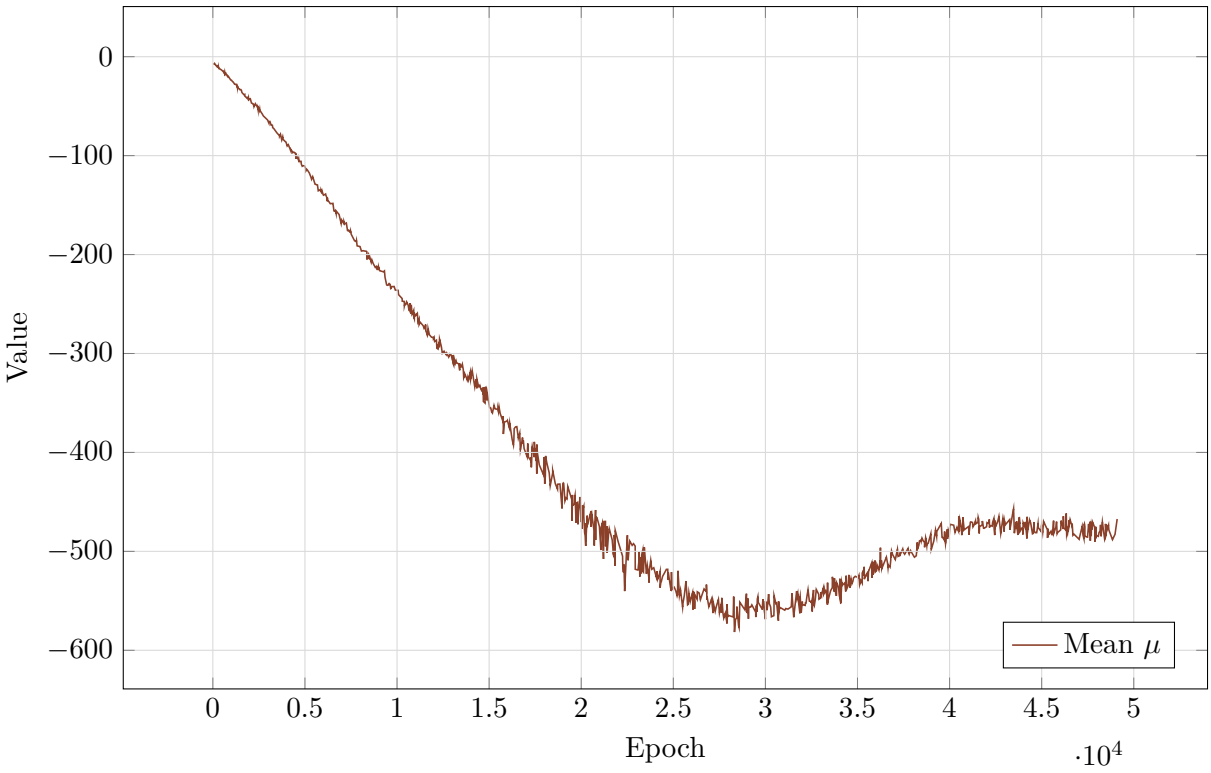

Figure 9: Policy Loss

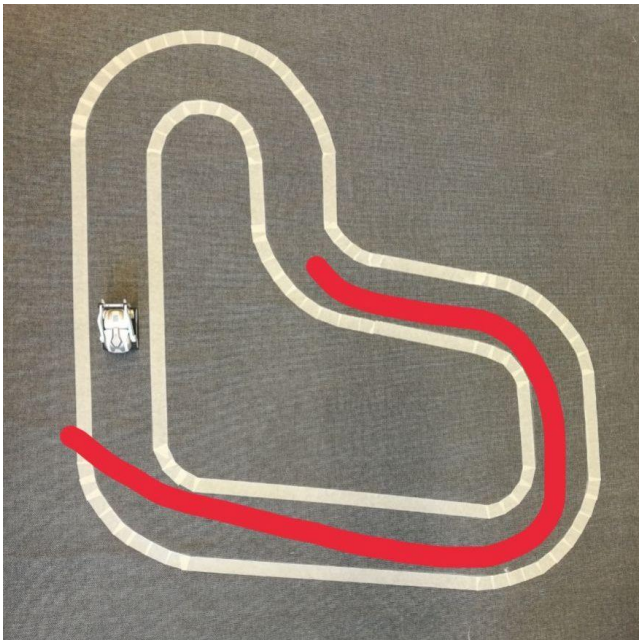

Figure 10: Episode 400 of Cozmo

2 Next Steps

2.1 Cozmo-DDPG

One crucial step for the thesis is the implementation of the DDPG algorithm for Cozmo in order to carry out a comparison between the two approaches.

2.2 Experiments

Now that the implementation allows more suitable learning timing, the next step is starting a batch of experiments using different environments (e.g. single-lane, single-line) on at least 300-500 episodes. This step is crucial to see how learning evolves and then check if the robot is learning or not.