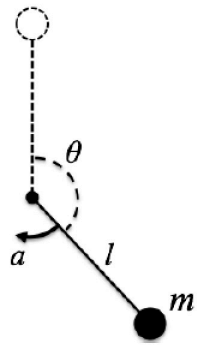
Programming Session: Exercise

Exercise 1: Inverted Pendulum



$s = \{\theta, \dot{\theta}\}, a = \tau$

$\theta = [-\pi, \pi]$

Re-scale for values out of range for the angular position.

$\dot{\theta} = [-2\pi, 2\pi]$

Angular velocity limits.

Dynamics equation:

$$ml^2\ddot{\theta} = -\mu\dot{\theta} + mgl\sin\theta + \tau$$

Parameters' values:

$m = 1$

$l = 1$

$g = 9{,}8$

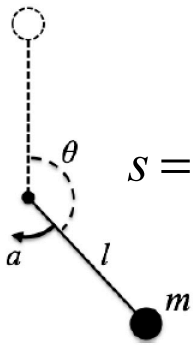
$\mu = 0.01$

$\tau = [-5,5]$ Input torque (underactuated)

Implement a simulator of the inverted pendulum using the Euler method with a differential time dt=0.001 seconds.

Implemet a function to visualize the behaviour of the pendulum for actions drawn from a uniform distribution in [-5,5] and using an action interval of 0.1 seconds (i.e. select a new action every 0.1 seconds).

Exercise 2: Variable Resolution Q-Learning



$$s = \{\theta, \dot{\theta}\}, a = \tau$$

$$r(\theta, \dot{\theta}) = -|\theta|$$

Reward r(s')

Implement Q-Learning with variable resolution function approximation. Use the simulation of the inverted pendulum of previous exercise for action execution with an action interval of 0.1 sec.

Perform training episodes of 500 iterations starting with the pendulum in the downwards position. After each training episode, perform a test episode of 500 iterations only using the policy learned so far (no learning) and memorize the accumulated reward. Execute a total of 100 training episodes (or more if needed).

Plot the evolution of the accumulated reward of the test episodes after completing the training episodes. Visualize the behaviour of the pendulum during 10 seconds using the final policy starting from the downwards position.

Programming Session

Implement the code (preferably) in Matlab. Save all the implemented files in a folder `L3_surnames`. Implement a script called `test.m` that executes the implemented functions and presents the requested results.

Send the folder compressed (.zip) by email to alejandro.agostini@tum.de with the subject RLRWS20_L3_surnames.