

Mountain Car

Environment:

A car is on a one-dimensional track, positioned between two "mountains". The goal is to drive up the mountain on the right; however, the car's engine is not strong enough to scale the mountain in a single pass. Therefore, the only way to succeed is to drive back and forth to build up momentum. Here, the reward is greater if you spend less energy to reach the goal

State space:

The agent (a car) is started at the bottom of a valley. For any given state the agent may choose to accelerate to the left, right or cease any acceleration.

Observation:

Type: Box(2)

Num	Observation	Min	Max
0	Car Position	-1.2	0.6
1	Car Velocity	-0.07	0.07

Action Space:

Type: Box(1)

Num	Observation	Min	Max
0	the power coef	-1.0	1.0

Note: actual driving force is calculated by multipling the power coef by power (0.0015)

Reward:

Reward of 100 is awarded if the agent reached the flag (position = 0.45) on top of the mountain.

Reward is decrease based on amount of energy consumed each step.

Starting State:

The position of the car is assigned a uniform random value in $[-0.6, -0.4]$.

The starting velocity of the car is always assigned to 0.

Episode Termination:

The car position is more than 0.45

Episode length is greater than 200

Citation:

This problem was first described by Andrew Moore in his PhD thesis [Moore90]. Here, this is the continuous version.

[Moore90]A Moore, Efficient Memory-Based Learning for Robot Control, PhD thesis, University of Cambridge, 1990.

Source code for environment:

https://github.com/openai/gym/blob/master/gym/envs/classic_control/continuous_mountain_car.py