Markov Decision Process
in
Bipedal Locomotion

Markov Process

A markov process is a stochastic process where the probability distribtion of fututre states only depends on the current state of the system and not any previous states.

$$\mathcal{P}_{ss'} = \mathbb{P}\left[S_{t+1} = s' \mid S_t = s\right]$$

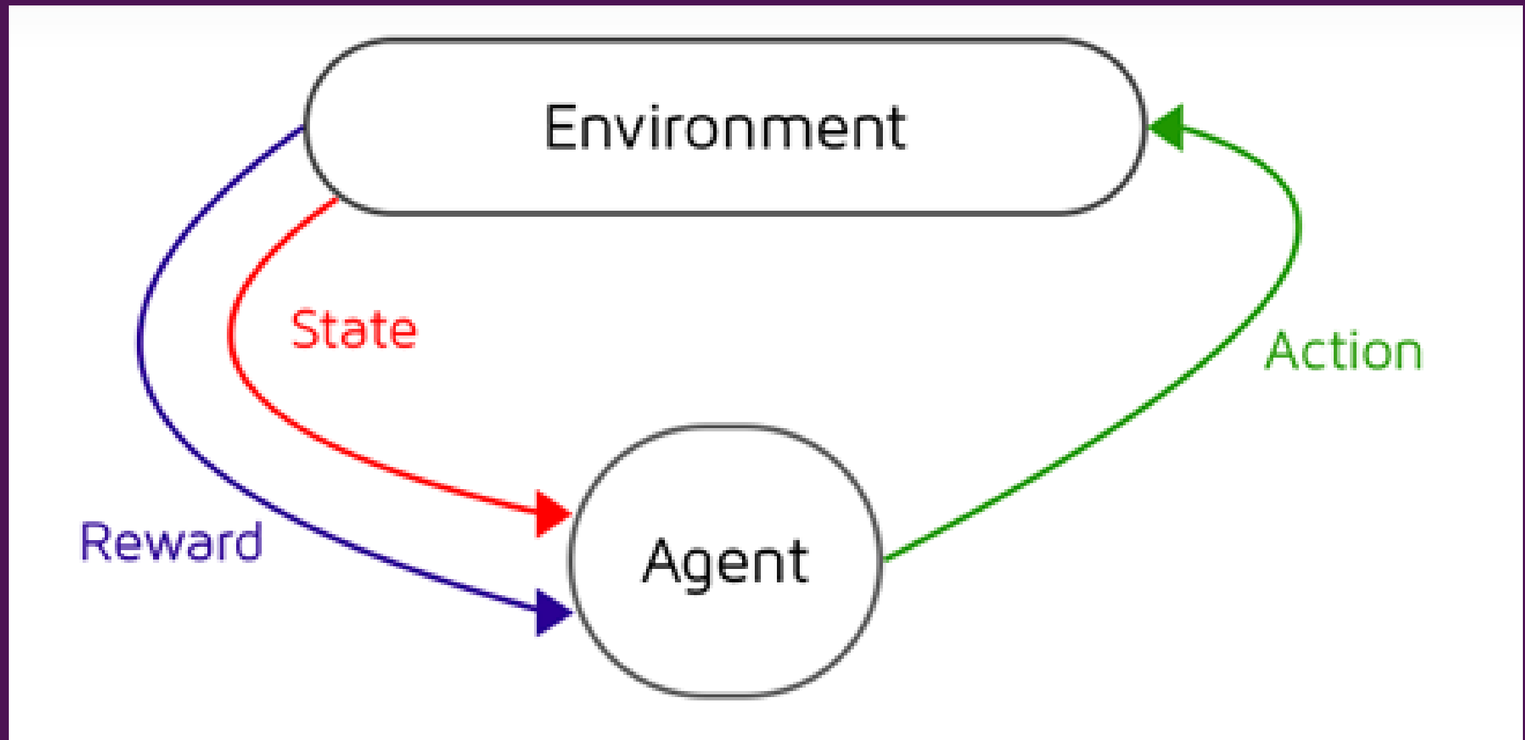
$$\mathcal{P} = \begin{matrix} \\ \text{from} \end{matrix} \begin{matrix} \text{to} \\ \begin{bmatrix} \mathcal{P}_{11} & \cdots & \mathcal{P}_{1n} \\ \vdots & & \\ \mathcal{P}_{n1} & \cdots & \mathcal{P}_{nn} \end{bmatrix} \end{matrix}$$

What is Markov Reward process ?

We can extend the idea of Markov Processes to include a “reward” term.It can be thought of as a Markov Process, but with a “reward” associated with every single possible transition among states.

What is Markov Decision process ?

This is a further extension of the Markov Reward Process, and includes an “action” component to it.



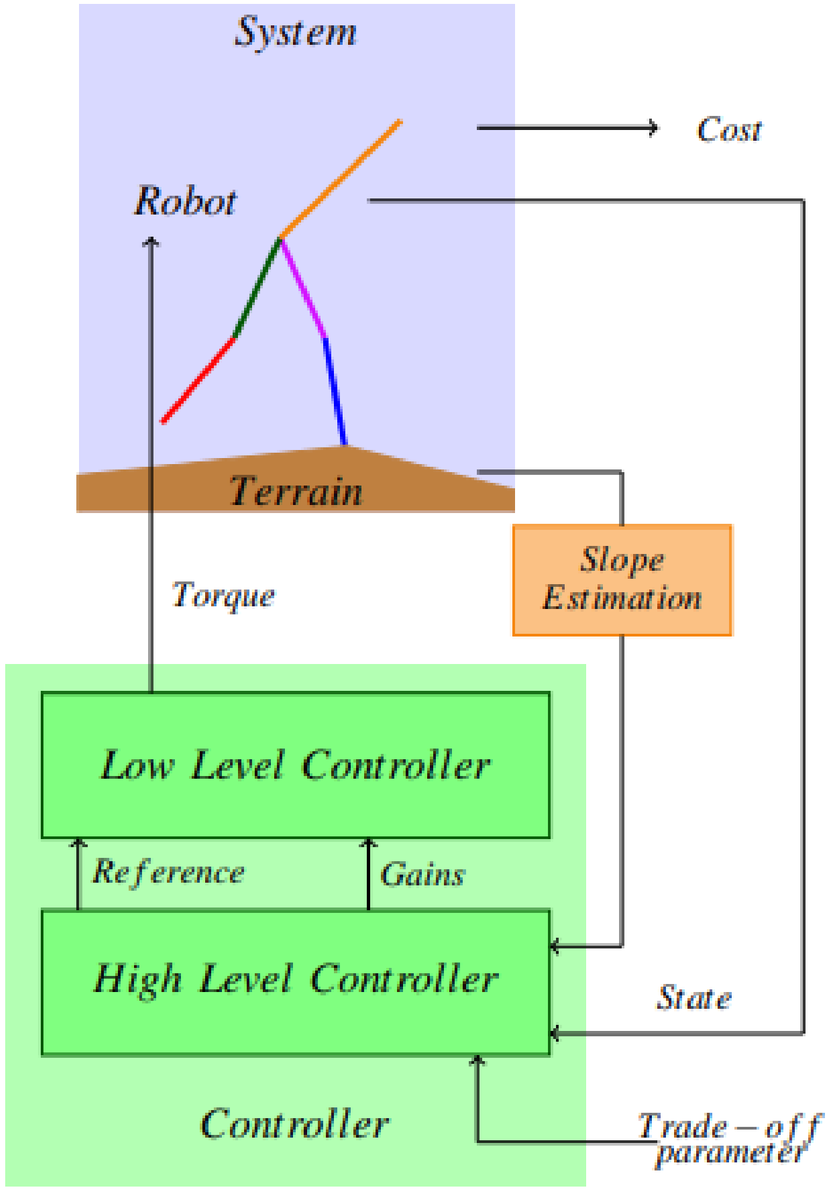
Spot by Boston Dynamics

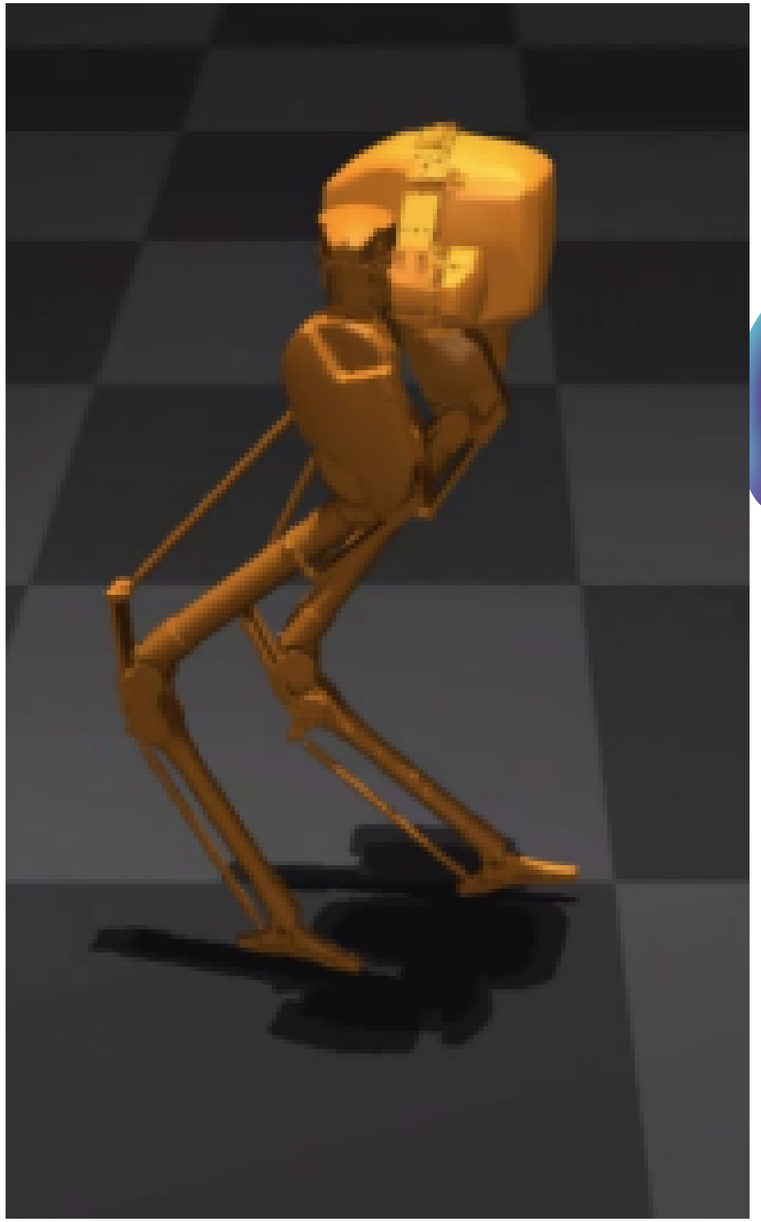
MDP → Robotics

SETTING UP THE ENVIRONMENTAL INTERACTION

MDPs are commonly used in robotics to model the control of robotic systems. In the case of walking, an MDP can be used to model the decision-making process involved in determining the next step to take.

To model the walking process with an MDP, we first need to define the state space, action space, transition probabilities, and rewards.
The state space would consist of the current state of the robot, which could include its position, velocity, and orientation.
The action space would consist of the possible steps the robot can take, such as moving one leg forward, backward, or to the side.


System
Cost
Robot
Terrain
Slope
Estimation
Torque
Low Level Controller
Reference
Gains
High Level Controller
State
Controller
Trade – off
parameter



Bellman Expectation Equations

$$
\begin{aligned}
V(s) &= \mathbb{E}[G_t|S_t = s] \\
&= \mathbb{E}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots | S_t = s] \\
&= \mathbb{E}[R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} + \ldots) | S_t = s] \\
&= \mathbb{E}[R_{t+1} + \gamma G_{t+1} | S_t = s] \\
&= \mathbb{E}[R_{t+1} + \gamma V(S_{t+1}) | S_t = s]
\end{aligned}
$$

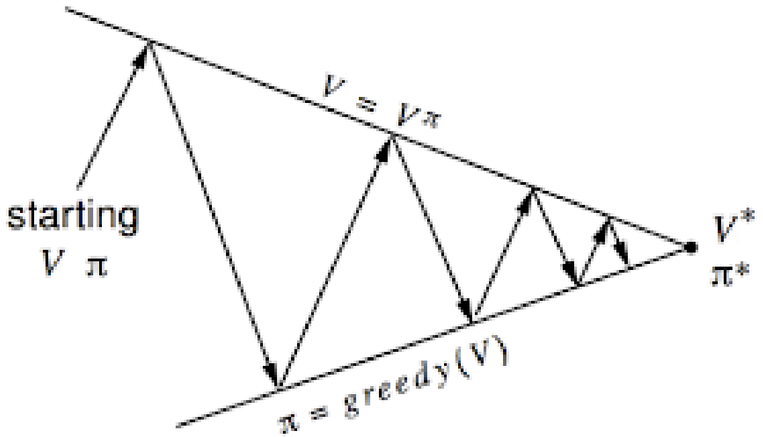
The Bellman expectation equations describe the relationship between an action in a Markov decision process (MDP) and the values of its successor states or actions.

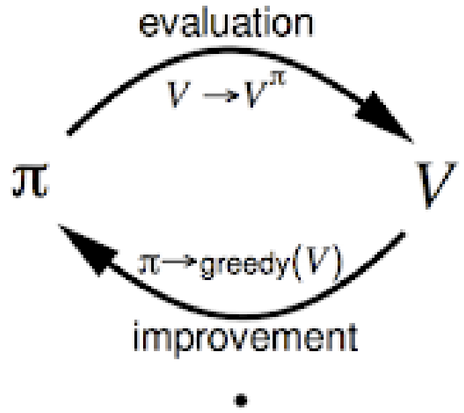
Applicable for both the state value function equation and the action value function equation

$$
\begin{aligned}
V_*(s) &= \max_{a \in \mathcal{A}} Q_*(s,a) \\
Q_*(s,a) &= R(s,a) + \gamma \sum_{s' \in \mathcal{S}} P^a_{ss'} V_*(s') \\
V_*(s) &= \max_{a \in \mathcal{A}} \left(R(s,a) + \gamma \sum_{s' \in \mathcal{S}} P^a_{ss'} V_*(s') \right) \\
Q_*(s,a) &= R(s,a) + \gamma \sum_{s' \in \mathcal{S}} P^a_{ss'} \max_{a' \in \mathcal{A}} Q_*(s',a')
\end{aligned}
$$

Solving the MDP

The final solution hence comprises of putting the two previous methods , Successive Policy evaluation and Policy Improvement over several iterations.





Solving known MDPs: Dynamic Programming

Settings where Dynamic Programming applies:

- Optimal substructure
- Overlapping subproblems

MDPs satisfy both the above conditions

We assume full knowledge of the MDP (ie. , we know the Probability Transition Matrices and the Rewards associated with each transition in the state space) to solve it with Dynamic Programming

What are Monte Carlo methods?

Monte Carlo methods are computational techniques that use random sampling to approximate the solution to complex problems.

$$V(s) = \frac{\sum_{t=1}^{T} \mathbb{1}[S_t = s] G_t}{\sum_{t=1}^{T} \mathbb{1}[S_t = s]}$$

THANK YOU