Review:

Bellman equation:

$$V(x_t, t) = \min_{u_t} \{ L(x_t, u_t) + V(\underbrace{x_{t+1}}_{f_D(x_t, u_t)}, t+1) \}$$

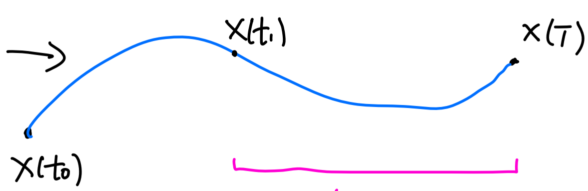
$$V(x, T+1) = \ell(x)$$

Continuous-time OC problem via DP

$$\min_{u(\cdot)} J(x(t), u(\cdot), t_0) = \int_{s=t_0}^{T} L(x(s), u(s))\,ds + \ell(x(T))$$

$$\text{s.t.} \quad \dot{x}(s) = f_c(x(s), u(s)) \qquad s \in [t_0, T]$$

optimal trajectory →
(by applying $u(\cdot)$)



optimal trajectory from $[t_1, T]$

$$V(x, t) = \min_{u(\cdot)} J(x, u(\cdot), t)$$

$$= \min_{u(\cdot)} \left\{ \int_{s=t}^{t+\delta} L(x(s), u(s))\,ds + J(x(t+\delta), u(\cdot), t+\delta) \right\}$$

$$= \min_{u(\cdot)} \left\{ \int_{s=t}^{t+\delta} L(x(s), u(s))\,ds + V(x(t+\delta), t+\delta) \right\}$$

Now, suppose δ is very small.

$$V(x,t) = \min_{u(\cdot)} \left\{ L(x(t),u(t))\delta + V(x(t+\delta), t+\delta) \right\}$$

$$= \min_{u(t)} \left\{ L(x(t),u(t))\delta + V(x(t+\delta), t+\delta) \right\}$$

Taylor Expansion around $V(x(t),t)$

$$V(x(t+\delta), t+\delta) = V(x(t),t) + \frac{\partial V}{\partial x} \cdot (x(t+\delta) - x(t))$$

$$+ \frac{\partial V}{\partial t} \cdot (t+\delta-t) + h.o.t.$$

$$= \boxed{V(x(t),t) + \frac{\partial V}{\partial x} f_c(x(t),u(t))\delta + \frac{\partial V}{\partial t}\cdot\delta + h.o.t.} \quad A$$

$$\boxed{\begin{aligned} \dot{x}(t) &= f_c(x(t),u(t)) \\ x(t+\delta) &\approx x(t) + \delta\,\dot{x}(t) \end{aligned}}$$

Ignoring h.o.t. and plug in A

$$V(x(t),t) = \min_{u(t)} \left\{ L(x(t),u(t),t)\delta + V(x(t),t) + \frac{\partial V}{\partial x}\cdot f_c(x(t),u(t))\delta + \frac{\partial V}{\partial t}\delta \right\}$$

$$V(x(t),t) = V(x(t),t) + \frac{\partial V}{\partial t}\cdot\delta + \delta \min_{u(t)} \left\{ L(x(t),u(t)) + \frac{\partial V}{\partial x}\cdot f_c(x(t),u(t)) \right\}$$

$$\delta\left[\frac{\partial V}{\partial t} + \min_{u(t)} \left\{ L(x(t),u(t)) + \frac{\partial V}{\partial x}\cdot f_c(x(t),u(t)) \right\} \right] = 0$$

This holds for any $\delta > 0$

$$\boxed{\frac{\partial V}{\partial t} + \min_{u(t)} \left\{ L(x(t),u(t)) + \frac{\partial V}{\partial x}\cdot f_c(x(t),u(t)) \right\} = 0}$$

Hamilton - Jacoby - Bellman PDE (terminal time PDE)

$$V(x,T) = \ell(x)$$

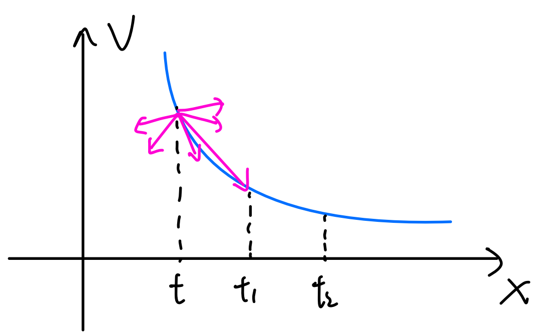
$$H(x,t,u,V) = L(x(t),u(t)) + \frac{\partial V}{\partial x}\cdot f_c(x(t),u(t))$$

hamiltonian

Suppose $L(x(s),u(s)) \equiv 0$

PDE reduces to:

$$\frac{\partial V}{\partial t} + \min_{u(t)} \left\{ \frac{\partial V}{\partial x}\cdot f_c(x(t),u(t)) \right\} = 0$$



$$V(x_t,t) = \min_{u_t} \{ V(x_{t+1},t+1) \}$$

$$= \min_{u_t} \{ V(f_D(x_t,u_t),t+1) \}$$

$$f_D(x_t,u_t) = x_t^2 + u_t$$

$$f_c(x(t),u(t)) = x^2(t) + u(t)$$

$$V(x_t,t) = \min_{u_t} \{ V(x_{t+1},t+1) \}$$

$$= \min_{u_t} \{ V(x_t^2 + u_t, t+1) \}$$

$$= \min_{u_t} g(u_t) \leftarrow \text{nonlinear}$$

$$\min_{u(t)} \frac{\partial V}{\partial x}\cdot(x^2(t)+u(t))$$

$$= \frac{\partial V}{\partial x}\cdot x^2(t) + \min_{u(t)} \frac{\partial V}{\partial x}\cdot u(t)$$

Methods to solve

(1) closed-form computation for V

- LQR is an example
- very rare

(2) Tabular / grid-based methods to compute V

- discretize the state to create a grid in the state space as well as discretize the control
- the value function is computed over the grid, starting from the terminal time

P_y

0	0	0	0	0	0
0	0	0	0	-5	0
0	0	0	0	0	0
0	0	0	10	0	0
0	0	0	0	0	0

P_x

$V(x, T+1)$

move $\rightarrow$ or $\uparrow$

0	0	0	0	0	0
0	0	0	-5	-5	0
0	0	0	0	-5	0
0	0	0	10	0	0
0	0	0	0	0	0

$V(x, T)$

- the number of grid points $= (N)^{n_x}$ ← # of states
  - exponential in # of states
  - curse of dimensionality
  - 5D - 6D MAX

(3) Function-approximation-based methods

- approximate value function as a parametrized function

$$V(x,t) \equiv V(x,t;\theta) \equiv V_\theta(x,t)$$

- These parameters are computed to satisfy the Bellman eqn as closely as possible.

$$\boxed{\begin{aligned} \theta^* = \arg\min_{\theta} \; & \left\| V_\theta(x,t) - \min_u \{ L(x,u) + V_\theta(x_{t+1}, t+1) \} \right\| \\ & + \\ & \lambda \left\| V_\theta(x, T+1) - \ell(x) \right\| \qquad \forall (x,t) \end{aligned}}$$

- These methods are also called "Fitted Value Iteration" because they fit the value function with an approximation function.

- A variety of function approximation can be used
  - Neural network $\quad x, t \to \boxed{\begin{matrix} NN \\ \theta \end{matrix}} \to V_\theta(x,t)$
  - Quadratic $\quad$ - Linear combinations of "feature" functions.
  - Polynomials

- Variants to this algorithm
    - DQN (Deep Q-Network)
    - Actor-critic methods