I. DEFINITIONS AND PRINCIPLE OF OPTIMALITY

System characterized by:

- State $x \in \mathcal{X}$
- Control $u \in \mathcal{U}$
- "evolution operator" : $x_{k+1} = next(x_k, u_k)$

GOAL: Minimize a <u>cost</u> function $J(\{x\},\{u\})$ by choosing $\{u\}$

sequences of states/controls

<u>Def</u>. $\pi(x) = u$ is called <u>control law</u> (or <u>policy</u>) and implements <u>closed loop control</u>

Under the <u>assumption</u> that:

$$J(\{x\},\{u\}) = \sum_{k=0}^{k_f-1} cost(x_k, u_k) \qquad (1)$$

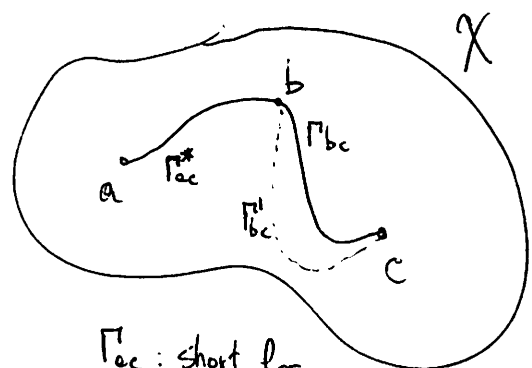
↳ cost of action using control u_k in state x_k

i.e. <u>COST</u> is separable in time

Then ⟹ it holds the <u>PRINCIPLE OF OPTIMALITY</u>:
(optimal substructure)

If Γ^*_{ac} achieves minimum cost ~~from~~ a to c

$\Longrightarrow \Gamma_{bc}$ " " " from b to c



Γ_{ac} : short for the seq. of states and controls that induces the trajectory

<u>Proof</u>: (by contradiction)

If $\exists\, \Gamma'_{bc}$ s.t. $J(\Gamma'_{bc}) < J(\Gamma_{bc})$

$$\Longrightarrow J(\Gamma_{ab}) + J(\Gamma'_{bc}) < J(\Gamma_{ab}) + J(\Gamma_{bc}) \overset{(1)}{=} J(\Gamma^*_{ac})$$

$$\overset{(1)}{\Longrightarrow} J(\Gamma'_{ac}) < J(\Gamma^*_{ac})$$

CONTRADICTION! ∎

Def INFINITE HORIZON PROBLEM :

Optimal Control problem with no given endpoint

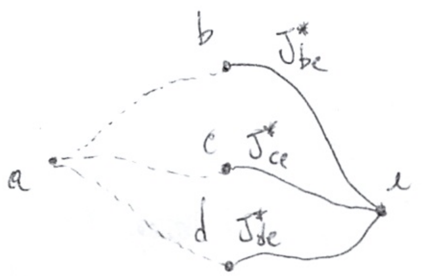
$$\Rightarrow \quad J(\{x\},\{u\}) = \sum_{k=0}^{\infty} \gamma^k \, cost(x_k, u_k) \quad , \quad \gamma < 1$$

II. DYNAMIC PROGRAMMING (DP)

DP: Set of algorithms that exploit the optimality principle to solve optimization problems

MOTIVATION : Direct enumeration of all possible trajectories to find minimum is exponential in the number of timesteps

INSTEAD, consider :



Which of ~~a~~ $\Gamma_{ab}, \Gamma_{ac}, \Gamma_{ad}$ is optimal?

If we know J^*_{be}, J^*_{ce}, J^*_{de}, it's sufficient to compute

$$J_{ae} = \min_{i=b,c,d} \left(J_{ai} + J^*_{ie} \right)$$

by starting from the endpoint, one can REUSE previously computed quantities

$\Rightarrow$ DP scales linearly with the number of timesteps

IN EQUATIONS

<u>Def</u> $V_{\pi}(x)$: <u>value</u> of a state, i.e. total cost starting from x to $x(T)$, <u>following the control law $\pi(x)$</u>

<u>Def</u> $V^*(x) := V_{\pi^*}(x)$, π^* : <u>optimal</u> control law

$\Longrightarrow$ Iterative application of the <u>optimality principle</u> lead to:

$$V^*(x) = \min_u \left[cost(x,u) + V^*(next(x,u)) \right] \qquad (2) \qquad \boxed{\text{Bellman equation}}$$

$$\pi^*(x) = \operatorname{argmin}_u \left[cost(x,u) + V^*(next(x,u)) \right]$$

Bellman eq. is a <u>RECURSIVE EQ</u> for the <u>OPTIMAL VALUE FUNCTION</u>

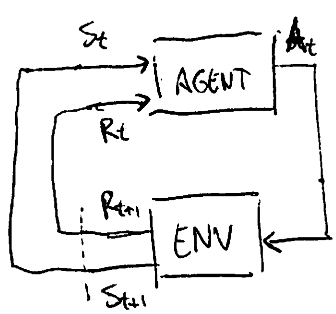
III a) WHAT is "next(x,u)" ?

<u>Optimal control</u> : Dynamical system $\dot{\vec{x}} = f(\vec{x}, \vec{a}, t)$

OR

$$d\vec{x} = f(\vec{x}, \vec{u}, t)dt + F(\vec{x}, \vec{u})\, d\vec{W}$$

<u>Reinforcement learning</u> (RL) : • Markov decision process (MDP)



S_t, R_t, A_t are <u>random variables</u>, fully characterized by

$$p(s', r \mid s, a) : P(S_{t+1} = s', R_{t+1} = r \mid S_t = s, A_t = a)$$

only previous state $\Rightarrow$ MARKOVIAN

b) WHAT is "cost(x,u)"?

- Optimal Control. Arbitrary function $J = h(x(t_f)) + \int_0^{t_f} \ell(x(t),u(t),t)\,dt$

EXAMPLES: Minimum effort; Minimum time; Endpoint control; tracking problems ...

- RL: Instead of min. cost, we want max Total return (expected)

Def $G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$, $\gamma < 1$ Total return

Def $\pi(a|s)$ Policy, prob. of action a given state s

GOAL: Maximize $E_\pi[G_t]$

Def (VALUE) $V_\pi(s) = E_\pi[G_t \mid S_t = s] = E_\pi[R_{t+1} + \gamma V(S_{t+1}) \mid S_t = s]$ (3)

- $V^*(s)$: Optimal value function (achieves max return)

$V^*(s)$ obeys

$$\boxed{\begin{aligned} V^*(s) &= \max_a E\left[R_{t+1} + \gamma \cdot V^*(S_{t+1}) \mid S_t = s, A_t = a\right] \\ &= \max_a \sum_{s',r} p(s',r \mid s,a)\left[r + \gamma V^*(s')\right] \end{aligned}} \quad (4)$$

Bellman optimality eq. in RL

IV. DP in RL

(4) is a set of $|S|$ nonlinear eqs. $\Rightarrow$ iterative methods to solve it

$\Rightarrow$ DP methods:

- Policy iteration : Alternate between:
  - Policy evaluation : $\pi(s) \longrightarrow V(s)$ by repeatedly use (3)
  - Greedy policy improvement :

$$\tilde{\pi}(s) = \underset{a}{\operatorname{argmax}}\, V(s)$$

- Value iteration : turn (4) into an iterative update :

$$V^{(k+1)}(s) = \max_a \left\{ \sum_{s',r} p(s',r|s,a)\left[r + \gamma V^{(k)}(s') \right] \right\}$$

Can be seen as a policy iteration, with policy evaluation step truncated after one update

D.P. - Pros

- Reduces complexity thanks to principle of optimality (BOOTSTRAPPING)

D.P. - Cons

- Requires knowledge of $p(s',r|s,a)$ (MODEL-BASED)

V. Continuous Control

System: $d\vec{x} = \vec{f}(\vec{x},\vec{u})\,dt + F(\vec{x},\vec{u})\,d\vec{W}$ ↳ Brownian motion

$\vec{u}, \vec{x} \in \mathbb{R}^n, \quad F(\vec{x},\vec{u}) \in \mathbb{R}^{n\times n}$

discretization $\Longrightarrow$ $\vec{x}_{k+1} = \vec{x}_k + \Delta\cdot\vec{f}(\vec{x}_k,\vec{u}_k) + \sqrt{\Delta}\cdot F(\vec{x}_k,\vec{u}_k)\cdot \varepsilon_k \quad , \quad \varepsilon_k \sim N(0, I_n)$

↳ time step

Cost:

$$J(\vec{x}(\cdot),\vec{u}(\cdot)) = h(\vec{x}(t_f)) + \int_0^{t_f} \ell(\vec{x}(t),\vec{u}(t),t)\,dt \rightarrow h(x(t_f)) + \sum_{k=0}^{K-1} \ell(\vec{x}_k,\vec{u}_k,k\cdot\Delta)\Delta$$

$$\Longrightarrow \underbrace{\vec{x}_{k+1} = \vec{x}_k + \Delta\cdot\vec{f}(\vec{x}_k,\vec{u}_k) + \xi}_{\text{Plug into Bellman eq (2)}} \quad , \quad \xi \sim N(0,\Delta S) \quad , \quad S = F(\vec{x},\vec{u})F^T(\vec{x},\vec{u})$$

$$V(\vec{x},k\cdot\Delta) = \min_{\vec{u}}\left\{\Delta\cdot\ell(\vec{x}_k,\vec{u}_k,\Delta\cdot k) + E\left[V\left(\vec{x}_k + \Delta\cdot\vec{f}(\vec{x}_k,\vec{u}_k) + \xi, (k+1)\Delta\right)\right]\right\}$$

IDEA: Expand V up to order Δ.

$$V(\vec{x}+\vec{\delta}) = V(\vec{x}) + \vec{\delta}^T\cdot\vec{\partial}_x V(\vec{x}) + \frac{1}{2}\vec{\delta}^T\cdot\partial_{xx}V(\vec{x})\cdot\vec{\delta} \qquad \text{(time index hidden)}$$

$$\vec{\delta} := \Delta\vec{f}(\vec{x}_k,\vec{u}_k) + \xi$$

$$\Rightarrow E\left[V(\vec{x}+\vec{\delta})\right] = V(\vec{x}) + \Delta\cdot\vec{f}^T(\vec{x}_k,\vec{u}_k)\cdot\vec{\partial}_x V(\vec{x}_k) + \underbrace{E\left[\frac{1}{2}\vec{\xi}^T\cdot\partial_{xx}V(\vec{x}_k)\vec{\delta}\right]}_{\frac{1}{2}E\left[\Delta\cdot\vec{\xi}\cdot\vec{\xi}^T\partial_{xx}V(\vec{x})\right] = \frac{\Delta}{2}\mathrm{Tr}\left(S\cdot\partial_{xx}V(\vec{x})\right)}$$

$$\hookrightarrow V(\vec{x},k\Delta) - V(\vec{x},(k+1)\cdot\Delta) = \min_{\vec{u}}\left\{\Delta\cdot\ell(\vec{x},\vec{u},\Delta\cdot k) + \Delta\vec{f}^T(\vec{x},\vec{u})\cdot\vec{\partial}_x V(\vec{x},(k+1)\cdot\Delta) + \frac{\Delta}{2}\mathrm{Tr}\left(S\,\partial_{xx}V\right)\right\}$$

$\lim_{\Delta \to 0} \Longrightarrow \quad -\partial_t V(\vec{x},t) = \min_{\vec{u}} \left\{ \ell(\vec{x},\vec{u},t) + \vec{f}^{\,T}(\vec{x},\vec{u}) \cdot \vec{\partial}_x V(\vec{x},t) + \frac{1}{2} \mathrm{Tr}\left(S \cdot \partial_{xx} V(\vec{x},t)\right) \right\}$ (5)

$\boxed{\text{Hamilton - Jacobi - Bellman Eq}}$ (HJB)

- HJB can be applied to any system / cost, including stochastic ones, but PDE solvers are exponential in n $\Rightarrow$ CURSE OF DIMENSIONALITY

VI. Detour: Maximum Principle from HJB (informal)

For a deterministic system

$$-\partial_t V(\vec{x},t) = \min_{\vec{u}} \left\{ \ell(\vec{x},\vec{u},t) + \vec{f}^{\,T}(\vec{x},\vec{u}) \cdot \vec{\partial}_x V(\vec{x},t) \right\}$$

Def. $\vec{p} := \vec{\partial}_x V(\vec{x},t)$ COSTATE

Def. $H(\vec{x},\vec{p},\vec{u},t) := \ell(\vec{x},\vec{u},t) + \vec{f}^{\,T}(\vec{x},\vec{u}) \cdot \vec{p}$

$$\Rightarrow -\partial_t V(\vec{x},t) = \min_{\vec{u}} \left\{ H(\vec{x},\vec{p},\vec{u},t) \right\}$$

$\Longrightarrow$ Differs from H-J eq. of classical mechanics only because of "$\min_u$"

$\Rightarrow$ One can show that, as in Classical mechanics:

$$\begin{cases} \dot{x}_i = \dfrac{\partial H}{\partial p_i} \\ \dot{p}_i = -\dfrac{\partial H}{\partial x_i} \\ \vec{u} = \arg\min_{\vec{u}} \left[H(\vec{x},\vec{p},\vec{u},t) \right] \end{cases}$$

with n init. cond. $x_i(0)$ and n boundary cond. $p_i(t_f) = \dfrac{\partial h(\vec{x}(t_f))}{\partial x_i}$

$\Longrightarrow$ System of n ODEs $\Rightarrow$ Avoids curse of dim.!

BUT:
- No noise
- init. cond. fixed

VII. Final Observations

- HJB can be solved ANALYTICALLY for Linear-Quadratic Gaussian regulator (LQG)

$\Rightarrow$ Used as an approximation technique for NONLINEAR PROBLEMS: iLQG:

Given $\vec{u}^{(k)}(t)$

- Run nonlinear dyn. forward $\xrightarrow{\text{get}}$ $\bar{x}(t)$, J
- Fit LQG approx. of $\bar{x}(t), J$.
- Solve analitycally for $\hat{u}$
- Set $\vec{u}^{(k+1)}(t) = \vec{u}^{(k)}(t) + \hat{u}(t)$

- RL – MODEL FREE METHOD

Combine Optimality principle + Montecarlo estimation of Value

$\Rightarrow$ TD learning

$$V(S_t) \leftarrow V(S_t) + \alpha \left[R_{t+1} + \gamma V(S_{t+1}) - V(S_t) \right]$$

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t) \right]$$

Requires $\{S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1}\} \Rightarrow$ SARSA