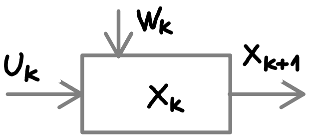
Dynamic programming



system dynamics: $x_{k+1} = f_k(x_k, u_k, w_k)$

(k: time index, N: time horizon; x_k: system state $\in S_k$; u_k: control input $\in \mathcal{U}_k(x_k)$ (dep. on x_k, u_k) → w_k: disturbances; $X = (x_1, \dots, x_N)$, $U = (u_0, \dots, u_{N-1})$, $W = (w_0, \dots, w_{N-1})$)

objective: find optimal input policies $\pi = (\mu_1, \dots, \mu_{N-1})$ to minimize cost (g_k: stage cost; g_N: terminal cost)

$$J^*(x_0) = \min_\pi J(x_0, \pi) = \min_\pi E\{g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k = \mu_k(x_k), w_k)\} \quad \text{subj. to } x_{k+1} = f_k(x_k, u_k = \mu_k(x_k), w_k)$$

[using expected value in cost is not ideal, as variance could be high, but having variance in cost makes dynamic programing no longer applicable!]

dynamic programing algorithm (DPA)

optimized tail of trajectory is also part of optimal trajectory, so: start from tail and iteratively find one input/policy to append that minimizes cost of current trajectory (= cost-to-go)

optimal cost-to-go: $J_k(x_k)$ optimal cost to go from state x_k to end of trajectory

recursion: $J_k(x_k) = \min_{\mu_k} \underset{(w_k|x_k,\mu_k)}{E} \{g_k(x_k, u_k = \mu_k(x_k), w_k) + J_{k+1}(f_k(x_k, u_k = \mu_k(x_k), w_k))\}$ initial cond: $J_N(x_N) = g_N(x_N)$

(in practice: minimize over u_k for all possible states x_k at time k → collection of $u_k \triangleq \mu_k(x_k)$)

[
- conversions to standard form:
- time lag: $x_{k+1} = f(x_k, x_{k-1}, u_k, u_{k-1}, w_k) \rightarrow \tilde{x}_k = \begin{bmatrix} x_k = x_k \\ y_k = x_{k-1} \\ s_k = u_{k-1} \end{bmatrix}$; $\tilde{x}_{k+1} = \begin{bmatrix} x_{k+1} \\ y_{k+1} \\ s_{k+1} \end{bmatrix} = \begin{bmatrix} f_k(x_k, y_k, u_k, s_k, w_k) \\ x_k \\ u_k \end{bmatrix} = \tilde{f}_k(\tilde{x}_k, u_k, w_k)$
- correlated disturbance: $w_k = C_k y_k$, $y_{k+1} = A_k y_k + \xi_k$ $\rightarrow \tilde{x}_k = \begin{bmatrix} x_k \\ y_k \end{bmatrix}$; $\tilde{x}_{k+1} = \begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} f_k(x_k, u_k, C_k(A_k y_k + \xi_k)) \\ A_k y_k + \xi_k \end{bmatrix} = \tilde{f}_k(\tilde{x}_k, u_k, \xi_k)$
- forecast: at the start of each step k we get a forecast y_k with which w_k correlates. The a priori prob of y_k is known.
  - ↳ $p(w_k|y_k) = \dots$
  - ↳ $p(\xi_k) = \dots$; $y_{k+1} = \xi_k$

  $\rightarrow \tilde{x}_k = \begin{bmatrix} x_k \\ y_k \end{bmatrix}$; $\tilde{w}_k = \begin{bmatrix} w_k \\ \xi_k \end{bmatrix}$; $\tilde{x}_{k+1} = \begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} f_k(x_k, u_k, w_k) \\ \xi_k \end{bmatrix} = \tilde{f}_k(\tilde{x}_k, u_k, \tilde{w}_k)$

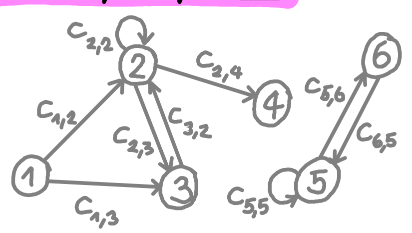
  $p(\tilde{w}_k|\tilde{x}_k, u_k) = \dots = p(w_k|y_k)p(\xi_k)$ $\quad J_k(\tilde{x}_k) = \min_{\mu_k} \underset{(w_k|y_k)}{E} \{g_k(x_k, u_k = \mu_k(x_k), w_k) + \underset{\xi_k}{E}\{J_{k+1}(\tilde{f}_k(\tilde{x}_k, u_k = \mu_k(x_k), \tilde{w}_k))\}\}$
]

infinite horizon (BE)

for a time invariant system + cost, let $N \to \infty$. Then $J_k = J_{k+1} = J_\infty$ and $\mu_k = \mu_{k+1} = \mu_\infty$ as 1 timestep is small compared to ∞

↳ Bellmann Equation: $J_\infty(x) = \min_{\mu_\infty} \underset{(w|x,\mu)}{E} \{g(x, u, w) + J_\infty(f(x, u = \mu_\infty(x), w))\} \rightarrow u = \mu(x)$: optimal input

shortest path problem (SP)



$\mathcal{V}$: vertex space ①, ...

$\mathcal{C}$: edge space

$\mathcal{C} := \{(i, j, c_{i,j}) \in \mathcal{V} \times \mathcal{V} \times \mathbb{R}\}$

$c_{i,j}$: lenght from vertex i to j

Q: path = ordered list of nodes

$Q := (i_1, \dots, i_q)$ (q: path lenght)

$\mathbb{Q}_{s,\tau}$: set of all paths starting at vertex $s \in \mathcal{V}$ and ending at $\tau \in \mathcal{V}$

objective: $Q^* = \underset{Q \in \mathbb{Q}_{s,\tau}}{\text{argmin}} J_Q$ (shortest path from vertex s to τ) ← $J_Q = \sum_{h=1}^{q} c_{i_h, i_{h+1}}$ (lenght of path Q)

assumption: no negative cycles! $\forall i \in \mathcal{V}$ and $\forall Q \in \mathbb{Q}_{i,i}$ $\nexists J_Q < 0$

stochastic shortest path problem (SSP) (discrete system defined by transition probabilities)

system: $x_{k+1} = w_k$ $\quad x_k \in S_k,\ u_k \in \mathcal{U}_k(x_k)$

$p_{x_{k+1}|x_k,u_k}(j|i,u) = p_{w_k|x_k,u_k}(j|i,u) = P_{ij}(u,k)$

(j: discr. state x_{k+1} #i ; i: discr. state x_k #j ; u: discr. input u_k #U)

(convert system of standard formulation: $P_{ij}(u) = \sum_{\{w_k | f(i,u,w_k)=j\}} p_{w_k|x_k,u_k}(\bar{w}_k|i,u)$)

objective: $J^*(x_0) = \min_{\pi} E\{g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k{=}\mu_k(x_k), w_k)\}$ (g_k: stage cost; g_N: terminal cost)

- **DPA + SSP problem**: $J_k(i) = \min_{\mu_k} q_k(i, u{=}\mu_k(i)) + \sum_{j=1}^{n} P_{ij}(u{=}\mu_k(i),k)\cdot J_{k+1}(j)$ $\quad \left(q_k(i,u) = \underset{(j|i,u)}{E}\{g_k(i,u,j)\}\right)$

- **BE + SSP problem**:

assumptions:
- time invariant SSP: $P_{ij}(u,k) \to P_{ij}(u)$; $g_k \to g$; $S_k \to S$; $\mathcal{U}_k(x_k) \to \mathcal{U}(x_k)$
- cost free termination state: state with $i=0$ is such that $P_{00}(u)=1$ and $g(0,u,0)=0$ $\forall u$
- $\exists$ a policy μ such that $i=0$ is eventually reached from all x_0 (propper policy)

↳ Bellmann equation: $J_\infty(i) = \min_{\mu_\infty}\left[q(i,u{=}\mu_\infty(i)) + \sum_{j=1}^{n} P_{ij}(u{=}\mu_\infty(i))\cdot J_\infty(j)\right]$ $\quad q(i,u) = \underset{(j|i,u)}{E}\{g(i,u,j)\} = \sum_{j=1}^{n} P_{ij}(u)\, g(i,u,j)$

value iteration: do DPA, until cost-to-go stops changing

$V_l \triangleq J_{N-l}$ cost-to-go l steps from end. → if l is large, then $V_l \approx J_\infty$

initial cond: $V_0(i)$ = some arbitrary cost

policy update: $\mu_l(i) = \underset{\mu}{\operatorname{argmin}}\left[q(i,u{=}\mu(i)) + \sum_{j=1}^{n} P_{ij}(u{=}\mu(i))\, V_l(j)\right]$

value update: $V_{l+1}(i) = q(i,u{=}\mu_l(i)) + \sum_{j=1}^{n} P_{ij}(u{=}\mu_l(i))\, V_l(j)$

break cond: $\|V_{l+1}(i) - V_l(i)\| < tol$

(not guaranteed to converge in finite iterations
less comp expensive, needs many iterations)

policy iteration: improve $\mu_\infty(i)$ guess until it converges

$J_{\mu_h}(i)$: cost to ∞ with policy $\mu_h(i)$

initial cond: $\mu_0(i)$ = some arbitrary propper policy

policy eval: solve $J_{\mu_h}(i) = q(i,u{=}\mu_h(i)) + \sum_{j=1}^{n} P_{ij}(u{=}\mu_h(i))\, J_{\mu_h}(j)$

policy update: $\mu_{h+1}(i) = \underset{\mu}{\operatorname{argmin}}\left[q(i,u{=}\mu(i)) + \sum_{j=1}^{n} P_{ij}(u{=}\mu(i))\, J_{\mu_h}(j)\right]$

break cond: $\|J_{\mu_{h+1}}(i) - J_{\mu_h}(i)\| < tol$

(guaranteed to converge in finite iterations
more comp expensive, needs few iterations)

→ variations:
- Gauss-Seidel update: in value update use $V_{l+1}(i)$ instead of $V_l(i)$ if it was already computed
- run multiple value updates with same policy before policy update ($\triangleq$ policy eval for inf.)
- update only some states in policy/value update (e.g. half in one iter., half in the following)

→ policy eval in matrix form: $\underline{J}_{\mu_h} = \underline{q} + \underline{\underline{P}}\,\underline{J}_{\mu_h}$

↳ $\underline{J}_{\mu_h} = (\underline{\underline{I}} - \underline{\underline{P}})^{-1}\underline{q}$

$\left(\underline{J}_{\mu_h} = \begin{bmatrix} J_{\mu_h}(1) \\ \vdots \\ J_{\mu_h}(n)\end{bmatrix} \quad \underline{q} = \begin{bmatrix} q(1,\mu_h(1)) \\ \vdots \\ q(n,\mu_h(n))\end{bmatrix} \quad \underline{\underline{P}} = \begin{bmatrix} P_{11}(\mu_h(1)) & \cdots & P_{1n}(\mu_h(1)) \\ \vdots & & \vdots \\ P_{n1}(\mu_h(n)) & \cdots & P_{nn}(\mu_h(n))\end{bmatrix}\right)$

→ linear program equivalent to value iteration: $J^* = \max_{V} \sum_{i=0}^{n} V(i)$ subj. to: $V(i) \le \left(q(i,u) + \sum_{j=0}^{n} P_{ij}(u) V(j)\right)$ $\forall u, \forall i$

→ discounted problem: (solve an auxiliary prob. to get solution to discounted prob.)

discounted SSP+BE: → auxiliary SSP+BE:

$\tilde{x}_{k+1} = \tilde{w}_k$ $\quad (\tilde{x}_k \in S^+ = \{1,\dots,n\}$, $\tilde{u}_k \in \tilde{\mathcal{U}}(\tilde{x}_k))$ → $x_k \in S = S^+ \cup \{0\}$, $\mathcal{U}(0) = \{stay\}$ add aux terminal state

$p_{\tilde{w}|\tilde{x},\tilde{u}}(j|i,u) = \tilde{P}_{ij}(u)$ ↖ terminal state not needed → $P_{ij}(u) = \alpha \tilde{P}_{ij}(u)$, $P_{i0}(u) = 1-\alpha$, $P_{0j}(u{=}stay) = 0$, $P_{00}(u{=}stay) = 1$

$\tilde{g}(\tilde{x}_k, \tilde{u}_k, \tilde{w}_k)$ → $g(x_k,u_k,w_k) = \alpha^{-1}\tilde{g}(x_k,u_k,w_k)$, $g(x_k,u_k,0) = 0$, $g(0,stay,0) = 0$

$\tilde{q}(\tilde{x}_k,\tilde{u}_k) = E\{\tilde{g}\} = \sum_{j=1}^{n} \tilde{P}_{ij}(u)\,\tilde{g}(i,u,j)$ → $q(x_k,u_k) = E\{g\} = \sum_{j=0}^{n} P_{ij}(u) g(i,u,j) = \dots = \tilde{q}(x_k,u_k)$, $q(0,u_k) = 0$

$\tilde{J}^*(\tilde{x}_0) = \min_{\tilde{\pi}} E\{\sum_{k=0}^{N-1} \alpha^k \tilde{g}(\tilde{x}_k, \tilde{u}_k{=}\tilde{\mu}_k(\tilde{x}_k), \tilde{w}_k)\}$ → $J^*(x_0) = \min_{\pi} E\{\sum_{k=0}^{N-1} g(x_k, u_k{=}\mu_k(x_k), w_k)\}$

$\tilde{J}_\infty(i) = \min_{\tilde{\mu}_\infty}\left[\tilde{q}(i,u{=}\tilde{\mu}_\infty(i)) + \alpha\sum_{j=1}^{n} \tilde{P}_{ij}(u{=}\tilde{\mu}_\infty(i))\cdot\tilde{J}_\infty(j)\right]$ → $J_\infty(i) = \min_{\mu_\infty}\left[q(i,u{=}\mu_\infty(i)) + \sum_{j=0}^{n} P_{ij}(u{=}\mu_\infty(i))\cdot J_\infty(j)\right]$

↳ $\mu_\infty(i) = \tilde{\mu}_\infty(i)$ $\quad \forall i \neq 0$

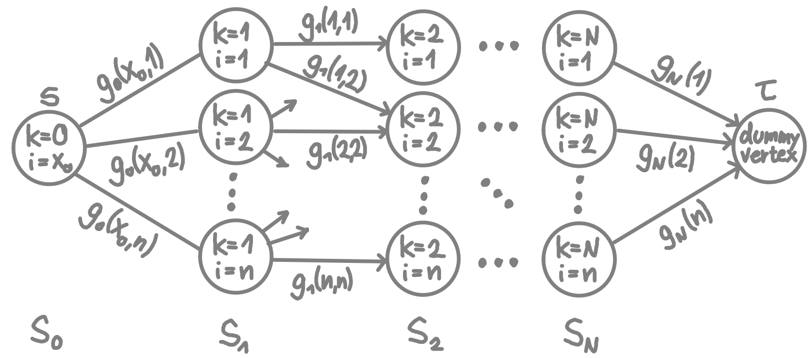
deterministic finite state problem (DFS)

system: $x_{k+1} = u_k \quad x_k \in S_k,\ u_k \in \mathcal{U}_k(x_k)$

objective: $J^*(x_0) = \min_\pi g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, u_k = \mu_k(x_k))$

j: discr. state x_{k+1} #i
i: discr. state x_k #j
u: discr. input u_k #u

- **DFS to equivalent SP:**



- **SP to equivalent DFS:**

set $c_{i,i} = 0\ \forall i \in \mathcal{V}$, search for paths of len. $N = |\mathcal{V}|$:

$S_0 = \{s\}\ ;\ S_k = \mathcal{V} \setminus \{\tau\}$ for $k = 1, .., N-1\ ;\ S_N = \{\tau\}$

$\mathcal{U}_k = \mathcal{V} \setminus \{\tau\}$ for $k = 0, .., N-2\ ;\ \mathcal{U}_{N-1} = \{\tau\}$

$$g_N(\tau) = 0\ ;\ g(x_k, u_k) = \begin{cases} 0 & \text{if } x_k = u_k \\ \infty & \text{if } \nexists\, c_{x_k,u_k} \\ c_{x_k,u_k} & \text{otherwise} \end{cases}$$

$\boxed{\text{solve with DPA} \to \text{skip degenerate moves}}$

- **DPA + DFS problem:** $J_k(x_k) = \min_{\mu_k} g_k(x_k, u_k = \mu_k(x_k)) + J_{k+1}(x_{k+1} = \mu_k(x_k))$ initial cond: $J_N(x_N) = g_N(x_N)$

- **forward DPA** for DFS derived from SP (works for all DFS)

optimal path from s to τ is also optimal path from τ to s if all edges are flipped $\to$ formulate aux. SP:

$\tilde{c}_{j,i} = c_{i,j} \quad \forall (i,j,c_{i,j}) \in \mathcal{C} \longrightarrow$

$\tilde{J}_N(s) = 0 \to \tilde{J}_{N-1}(j) = \tilde{c}_{j,s} \to \ldots \to \tilde{J}_k(j) = \min_i \tilde{c}_{j,i} + \tilde{J}_{k+1}(i) \to \ldots \to \tilde{J}_0(\tau) = \min_i \tilde{c}_{\tau,i} + \tilde{J}_1(i)$

$J^F_L = \tilde{J}_{N-L}\ ;\ L = N-k$

$\tilde{J}_N(s) = 0 \to \tilde{J}_{N-1}(j) = c_{s,j} \to \ldots \to \tilde{J}_k(j) = \min_i c_{i,j} + \tilde{J}_{k+1}(i) \to \ldots \to \tilde{J}_0(\tau) = \min_i c_{i,\tau} + \tilde{J}_1(i)$

$\boxed{J^F_L(j): \text{optimal cost-to-arrive}}$

$J^F_0(s) = 0 \to J^F_1(j) = c_{s,j} \to \ldots \to J^F_L(j) = \min_i c_{i,j} + J^F_{L-1}(i) \to \ldots \to J^F_N(\tau) = \min_i c_{i,\tau} + J^F_{N-1}(i)$

hidden Markov model + Viterbi algorithm

hidden Markov model: —

$\hookrightarrow$ given measurements $Z = (z_1, \ldots, z_N)$, find most likely state trajectory $X = (x_0, \ldots, x_N)$:

$x_{k+1} = w_k$ — $x_k \in S$ discrete states

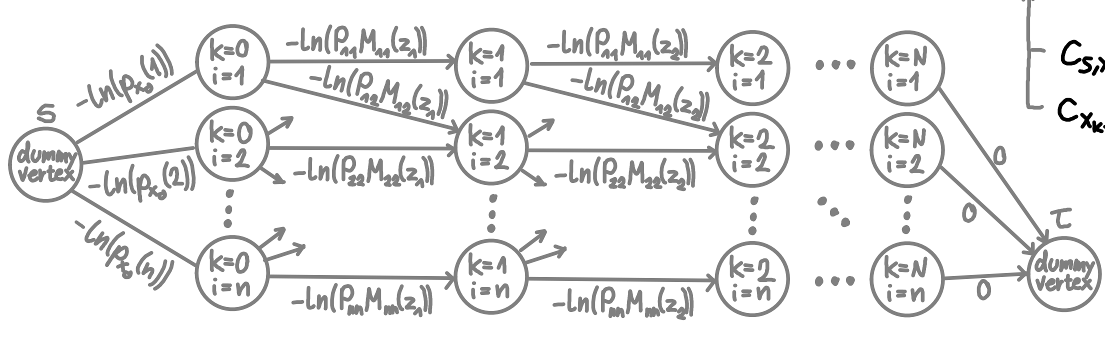
$P_{ij} = p_{w|x}(j|i)$ — prob. of moving from i to j

$M_{ij}(z) = p_{z|x,w}(z|i,j)$ — prob. of measuring z when moving from i to j

$$\hat{X} = \arg\max_X p(X|Z) = \arg\max_X p(X,Z) = \arg\max_X p(x_0) \cdot \prod_{k=1}^N P_{x_{k-1},x_k} M_{x_{k-1},x_k}(z_k) = \arg\min_X c_{s,x_0} + \sum_{k=1}^N c_{x_{k-1},x_k}$$

$c_{s,x_0} = -\ln(p(x_0))$

$c_{x_{k-1},x_k} = -\ln(P_{x_{k-1},x_k} M_{x_{k-1},x_k}(z_k))$



shortest path algorithms

- dynamic programming algorithm (DPA): convert SP to DFS and solve with DPA (see DFS chapter for details)
  ↳ inefficient because DPA finds shortest path from <u>any</u> vertex to τ, not just from s.

- **label correcting algorithm** (LCA): (same as forward DPA with DFS, but ignore brances with higher cost-to-arrive than current guess of cost-to-arrive at τ.)

$d_s = 0$; $d_j = \infty \ \forall j \in V \setminus \{s\}$; $\text{OPEN} = \{s\}$

while $\text{OPEN} \neq \{\}$
- remove a node i from OPEN
- for all children j of i $(c_{i,j} \neq \infty)$
  - if $(d_i + c_{i,j}) < d_j$ and $(d_i + c_{i,j}) < d_\tau$
    - set $d_j = d_i + c_{i,j}$
    - if $j \neq \tau$, then add j to OPEN

multiple ways to pick node i from OPEN:
- depth-first search: last in, first out
- breath-first search: first in, first out
- best-first search: pick i with smallest d_i (Dijkstras Algorithm)

result: $L_{Q^*} = d_\tau$ $\quad Q^*$ = parent of τ that last was in OPEN
→ recursion until s

- **A^*-algorithm**: if some positive lower bound h_j is known for the optimal path-lenght from j to τ, then replace $d_i + c_{i,j} < d_\tau$ with $d_i + c_{i,j} + h_j < d_\tau$ in LCA. (e.g. h_j = "air-line distance")

deterministic continuous time optimal control

system: $\dot{x}(t) = f(x(t), u(t)) \quad 0 \leq t \leq T$ $\qquad$ cost: $h(x(T)) + \int_0^T g(x(\tau), u(\tau))\, d\tau$

optimal cost to go: $J^*(t, x(t)) = \min_\mu h(x(T)) + \int_t^T g(x(\tau), u = \mu(\tau, x))$ subj. to: $\dot{x}(t) = f(x(t), u(t))$ $\qquad u \in \mathcal{U}, x \in \mathcal{X}$
↳ optimal cost: $J^*(0, x_0)$; optimal policy: $u = \mu^*(t, x)$

- **Hamilton-Jacobi-Bellman Equation** (HJB): (can be derived directly from DPA with step size $T_s \to 0$)

$$\min_\mu g(x(t), u = \mu(t, x)) + \partial J^* / \partial t \big|_{x(t)}^{t} + \partial J^* / \partial x \big|_{x(t)}^{t} \cdot f(x(t), u = \mu(t, x(t))) = 0 \quad \forall t \in [0, T], x \quad \text{and} \quad J^*(T, x(T)) = h(x(T)) \ \forall x$$

↳ hard to solve. easier: guess $\mu(t, x)$ → calculate $J_\mu(t, x)$ → if HJB eq. is fulfilled: $\mu^* = \mu$ $J^* = J_\mu$ optimal!

- **Pontryagin's minimum Principle**: (easier to solve than HJB, but only necessary cond. on μ, not sufficient)
(Hamiltonian function: $H(x, u, p) = g(x, u) + p^T f(x, u)$ with p: some auxiliary variable)

$u(t) = \arg\min_u H(x(t), u, p(t))$	ODE 1: $\dot{p}(t) = -\partial H / \partial x^T \big|_{x(t), u(t), p(t)}$	BC 1: $p(T) = \partial h / \partial x^T \big|_{x(T)}$
$H(x(t), u(t), p(t)) = \text{const.} \ \forall t \in [0, T]$	ODE 2: $\dot{x}(t) = f(x(t), u(t))$	BC 2: $x(0) = x_0$

usual approach: solve ODE1, ODE2, argmin assuming x, u, p are known → reformulate such that each eq. only depends on one of x, u, p → apply BC1, BC2 to get rid of integration constants

extensions:
- fixed terminal state: replace BC1 with $x(T) = x_T$ $\quad$ (x_T: terminal state)
- free initial state: replace BC2 with $p(0) = -\partial l / \partial x^T \big|_{x(0)}$ $\quad$ ($l(x)$: initial cost)
- free terminal time: solve for T with cond $H(x(t), u(t), p(t)) = 0$
- time varying f, g: H becomes func. of t. <u>drop</u> cond. $H(x(t), u(t), p(t), t) = \text{const.}$
- singular problem: if $\arg\min_u H$ is undefined, assume $\dot{p} = 0$ over that interval