Fall 2021 ME424 Modern Control and Estimation

Lecture Note 8
"Dynamic Programming" &
Linear Quadratic Regulator

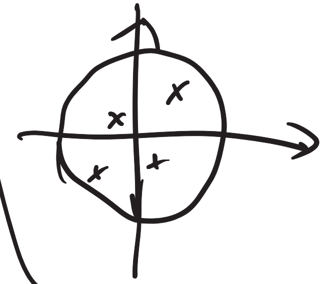
LQR

Prof. Wei Zhang
Department of Mechanical and Energy Engineering
SUSTech Institute of Robotics
Southern University of Science and Technology

zhangw3@sustech.edu.cn
https://www.wzhanglab.site/

- **Outline**
  - General Discrete-Time Optimal Control Problem
  - Short Introduction to Dynamic Programming
  - Linear Quadratic Regulator

- Closed-loop eigenvalues roughly indicate system response, but do not represent all factors: → asymptotic behavior
  - Same eigenvalues may also have different transient responses
  - We often want control input to be small, which cannot be formally addressed with eigenvalue assignment approach



optimization → key

modern engineering

Linear algebra
probability
optimization

L.S. → $\min_{\theta} \|y - g(\theta)\|$

- Metric-based controller design
  - Represent design objectives in terms a **cost function**
  - Cost functions typically **penalize**
    - state deviation from 0
    - Large control effort
  - These are conflicting goals: larger control can often drive state to zero faster

Kalman Filter:
$\min E(\|x - \phi(y)\|^2)$

General Discrete-Time Optimal Control Problem

- Dynamics: $x_{k+1} = f(x_k, u_k)$ $\Leftarrow$
- State constraints: $x_k \in X$
- Control constraints: $u_k \in U(x_k)$
- Controller (Control law): $\mu_k: X \to U$
- Control Horizon: $[0, N]$

e.g. motor control

temperature $\leq 80°$

func of state x_k

torque $\leq 35\,Nm$

$N \to \infty$

$[0, \infty) \Leftarrow$ infinite-horizon problem

$x \to$ controller $\to u$

- Control policy vs. control inputs:
  - Control policy: a sequence of control laws

$\pi = \{\mu_0(\cdot), \mu_1(\cdot) \cdots \}$

$\mu_1(x) = K_1 \cdot x$

$\pi = \{\mu(\cdot)\ \mu(\cdot) \cdots \mu(\cdot)\}$

stationary policy

  - Control inputs: a sequence of control actions

$u = \{u_0, u_1 \cdots, u_{N-1} \cdots \}$

General Discrete-Time Optimal Control Problem

- Closed-loop Dynamics under policy $\pi = \{\mu_0, \mu_1, \dots\}$

$x_{k+1} = f(x_k, u_k) \in$ open loop

↑ undetermined

CL: $x_{k+1} = f(x_k, \mu(x_k))$

only depends on state

- Quantify performance of controller through cost function
  - Running (stage) cost: $l(x_k, u_k)$: penalize undesired ~~beh~~ behavior

    desired state is $\hat{x}$, small energy

    $l(x,u) = \|x-\hat{x}\|^2 + \|u\|^2$

  - Terminal cost: $g(x_N)$

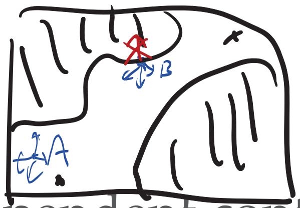
    e.g. $g(x_N) = \|x_N - \hat{x}\|^3$

  - N-horizon cost: $J_N(x_0, u) = g(x_N) + \sum_{k=0}^{N-1} l(x_k, u_k)$

  - Infinite horizon cost: $J_\infty(x_0, u) = \sum_{k=0}^{\infty} l(x_k, u_k)$

- **Finite Horizon Optimal Control** ($N < \infty$)
  - For given initial state $z \in \mathrm{R}^n$, find the control input $u_0, u_1, , \ldots, u_{N-1}$ to
    - **Minimize:** $J_N(z,u) = \sum l(x_k, u_k) + g(x_N)$
    - **subject to:** $u_k \in U(x_k)$, control constraint

      $x_{k+1} = f(x_k, u_k)$, $x_0 = z$ system dynamics constraints



    - Here: $U(x_k)$ is the set of state-dependent control action

      e.g. $U(x) = \{u \le 2x\}$

      state constraint $\overset{\text{can be}}{\Longrightarrow}$ state dependent control constraint

      $x_k \le 2$ ⊕ assume $x_k = x_{k-1} + 2u_{k-1} \Rightarrow u_{k-1} \le \frac{2 - x_{k-1}}{2}$

    - Optimizers $\{u_0^*, \ldots, u_{N-1}^*\}$ depends on the initial state z

Example I: Cessna Citation Aircraft

Linearized continuous-time model: (at altitude of 5000m and a speed of 128.2 m/sec)

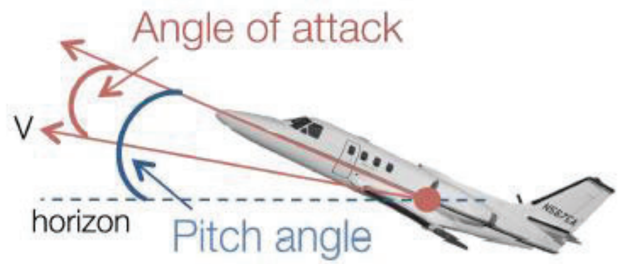
$$\dot{x} = \underbrace{\begin{bmatrix} -1.2822 & 0 & 0.98 & 0 \\ 0 & 0 & 1 & 0 \\ -5.4293 & 0 & -1.8366 & 0 \\ -128.2 & 128.2 & 0 & 0 \end{bmatrix}}_{A} x + \underbrace{\begin{bmatrix} -0.3 \\ 0 \\ -17 \\ 0 \end{bmatrix}}_{B} u$$

deviation from desired

$$y = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{C} x$$

Goal: ~~St.~~ make $x \to 0$

- Input: elevator angle
- States: x_1: angle of attack, x_2: pitch angle, x_3: pitch rate, x_4: altitude ~5000
- Outputs: pitch angle and altitude
- Constraints: elevator angle ± 0.262rad $(\pm 15^\circ)$, elevator rate ± 0.349 rad/s $(\pm 20^\circ/s)$, pitch angle ± 0.650 rad $(\pm 37^\circ)$
- Open-loop response is unstable (open-loop poles: $0,\ 0,\ -1.5594 \pm 2.29i$)



objective: to drive $x \to 0$ with small "control effort"

Example I: Cessna Citation Aircraft (A, B, C)

- Obtain DT-Model $dt = 0.25s$ $\Rightarrow$ $\begin{cases} x_{k+1} = A_d x_k + B_d u_k \\ y_k = C_d x_k \end{cases}$

- Choose cost func:

running cost: $l(x_k, u_k) = \|x_k\|^2 + \|u_k\|^2$, $g(x_N) = \|x_N\|^2$

- Choose constraint set: $l(x_k, u_k) = 100\|x_k\|^2 + \|u_k\|^2$

state constraint: $|u_k| \leq 0.262$ & $|x_{k,2}| \leq 0.65$

control

$|\dot{u}| \leq 0.349 \Rightarrow |u_k - u_{k-1}| \leq 0.349 \cdot \Delta t$

- Overall optimal control problem:

depends (z) $\leftarrow$ $\min\limits_{u_0, u_1, \cdots u_{N-1}} \left(\sum_{k=0}^{N-1} l(x_k, u_k) \right) + g(x_N)$

subject to: $x_{k+1} = A_d x_k + B_d u_k$, $x_0 = z$

$|u_k| \leq 0.262$, $|x_{k,2}| \leq 0.65$, $|u_k - u_{k-1}| \leq 0.349 \times 0.25$

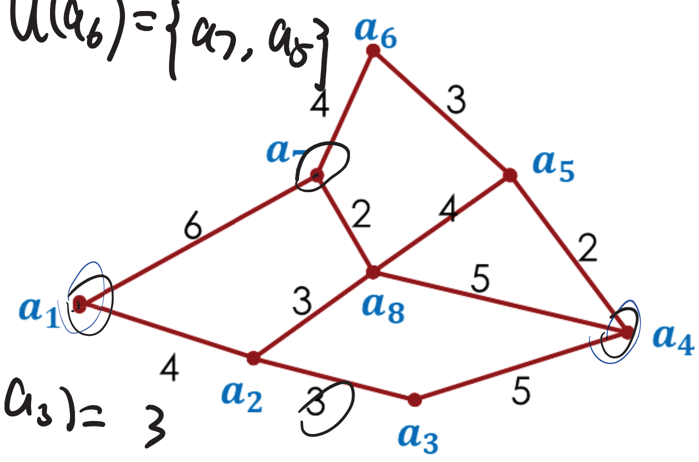
$u_k \in \mathcal{U}(x_k)$

Example II: Shortest Path Problem

goal: start from a_1, go to a_4 with minimum distance

- $X = \{a_1, \ldots, a_8\}$; $U(x)$: possible next site to visit e.g. $U(a_7) = \{a_1, a_8, a_6\}$

$U(a_6) = \{a_7, a_5\}$



- $x_{k+1} = f(x_k, u_k) = u_k$ ← next site to visit

- Running cost: $l(z, u) =$ edge "length"

e.g. $l(a_1, u = a_7) = 6$

$l(a_2, a_3) = 3$

- Terminal cost: $g(z) = \begin{cases} \infty & \text{if } z \neq a_4 \\ 0 & \text{if } z = a_4 \end{cases}$

- Optimal control problem: shortest path $a_1 \to a_4$ problem

$\boxed{N = 4}$

Parameter

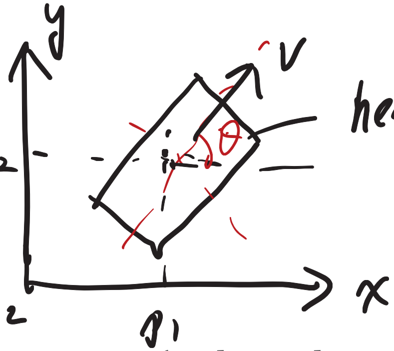
$$\begin{cases} \min \sum_{k=0}^{N-1} l(x_k, u_k) + g(x_N) \\ x_{k+1} = f(x_k, u_k) \\ u_k \in U(x_k) \end{cases}$$

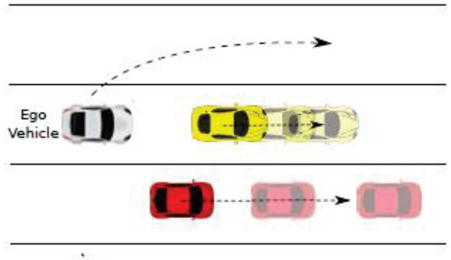
Example III: ~~Motion Planning~~ for Autonomous Vehicle

→ Tracking control

- Consider unicycle kinematic model: state $x = (p_x, p_y, \theta, v)$, control $\boxed{u = (\omega, \alpha)}$

- Dynamics: $\dot{x} = \begin{array}{cl} \begin{bmatrix} v\cos(\theta) \\ v\sin(\theta) \\ \omega \\ \alpha \end{bmatrix} & \begin{array}{l} \leftarrow \dot{p}_x \\ \leftarrow \dot{p}_y \\ \leftarrow \dot{\theta} \; u_1 \\ \leftarrow \dot{v} \; u_2 \end{array} \end{array}$





- Control Goal: Track a give reference $\underbrace{(p_k^d,\ v_k^d,\ \theta_k^d)}$ ↝ these are given by planning algo

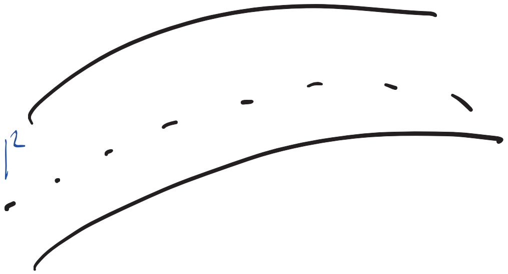
$\dot{x} = f(x,u) \Rightarrow x_{k+1} = \boxed{x_k + f(x_k,u_k)\cdot\Delta t}$

$f_d(x_k, u_k)$

$l(x_k, u_k)$

$\nearrow \alpha_k$

$$\min \sum_{k=0}^{N-1} \|p_k - p_k^d\|^2 + 0.5\,\|u_{k,2}\|^2 + 2\,\|u_{k,1}\|^2 + \|\theta_k - \theta_k^d\|^2 + 100\cdot \mathrm{dist2obstacle}(p_k)$$



subj to: $x_{k+1} = x_k + f(x_k, u_k)\Delta t$

$u_k \in U$ ← turning radius limit

Outline

- General Discrete-Time Optimal Control Problem
- **Short Introduction to Dynamic Programming**
- Linear Quadratic Regulator

Dynamic Programming (DP):

- Most important tool for solving deterministic and stochastic optimal control problems RL Q.

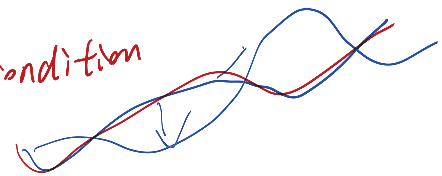
- **Divide & conquer:** The N-horizon optimal solution depends on the $N-1$ horizon optimal solution, which in turns depend on the $N-2$ horizon optimal solution …

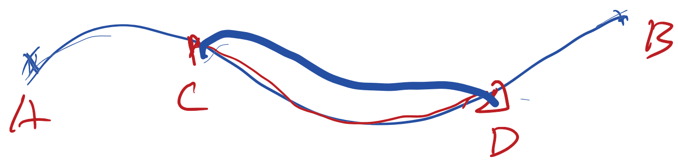
- We solve 0-horizon first, then 1-horizon, …, eventual solve the N-horizon optimal control problem.

- The divide & conquer approach is grounded by a fundamental principle: "**Bellman's principle of optimality**"

Any segment along an optimal trajectory is also optimal among all the trajectories joining the two end points of the segment

variational optimality condition





Dynamic Programming (DP)

Goal: solve N-horizon OC.

- For arbitrary integer $j \geq 0$, the j-horizon optimal control problem:

$\min_x x^2+1$

$f(x)$

value $=1$

$$V_j(z) = \min_{u_0,\dots,u_{j-1}} \left\{ g(x_j) + \sum_{k=0}^{j-1} l(x_k, u_k) \right\},$$

subject to $\quad x_{k+1} = f(x_k, u_k), \quad x_0 = z$

$\qquad\qquad u_k \in U(x_k), \quad k = 0, \dots, j-1$

cost-to-go

- $V_j^*(z)$: ***j*-horizon value function, i.e.** minimum cost if sys starts from state z when there are j steps left to reach final time

- Let $u_0^*, u_1^*, \dots, u_{j-1}^*$ is the optimal solution to the above prob. If system is at state z when there are j steps left, the first step of the optimal control is u_0^*, the second step is u_1^*, ….

$j=N$:

$$x_0 = z \xrightarrow{u_0^*} x_1^* = f(x_0, u_0^*) \xrightarrow{u_1^*} x_2^* \;\cdots\cdots \rightarrow x_N^*$$

$u_0^* = \mu_N^*(z)$

$u_1^* = \mu_{N-1}^*(x_1^*)$

$\mu_{N-1}^*(\cdot) \rightarrow$

$\min_x x^2 + e^x$

$\min_z z^2 + e^z$

Dynamic Programming: Value Iteration

- Value Iteration: Compute $V_N(z)$ iteratively from $V_0(z)$
- 0-horizon problem (degenerate case):

$J = 0 \Rightarrow V_0(z) = g(z)$

- 1-horizon problem: $V_1(z) = \min_{u_0 \in U(z)} \{ l(x_0, u_0) + g(x_1) \}$, $x_0 = z$, $x_1 = f(z, u_0)$

$\Rightarrow \boxed{V_1(z) = \min_{u_0 \in U(z)} \{ l(z, u_0) + V_0(f(z, u_0)) \}}$

$u_0^*(z) = \arg\min$ ___

- 2-horizon problem:

$z = x_0 \xrightarrow{u_0} x_1 \xrightarrow{u_1} x_2$

$$V_2(z) = \min_{\substack{u_0 \in U(z) \\ u_1 \in U(x_1)}} \left\{ \underline{l(x_0, u_0)} + \underline{l(x_1, u_1)} + g(\underline{x_2}) \right\}$$

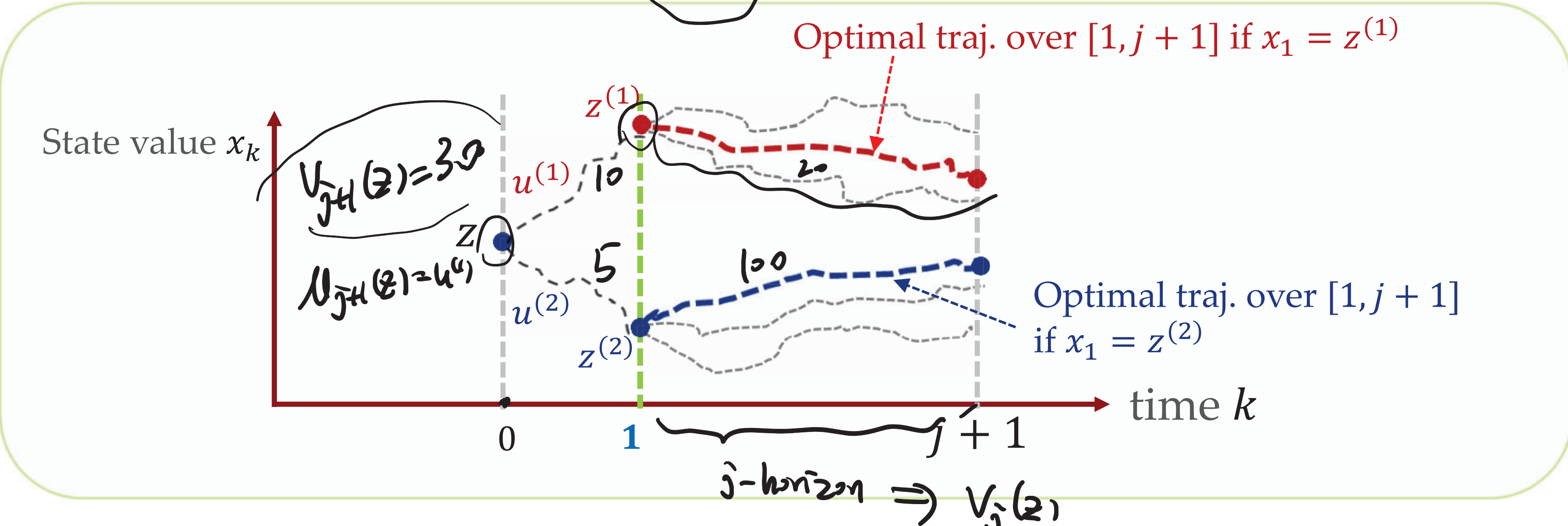
subj to: dynamics

$$= \min_{u_0 \in U(z)} \left\{ l(x_0, u_0) + \boxed{\underbrace{\min_{u_1 \in U(x_1)} \left[l(x_1, u_1) + g(f(x_1, u_1)) \right]}} \right\}$$

$$= \min_{u_0 \in U(z)} \left\{ l(z, u_0) + V_1(\underbrace{f(z, u_0)}_{x_1}) \right\}$$

$$\underline{\mu_1^*(z)} = \operatorname*{argmin}_{u \in U(z)} \left\{ l(z, u) + V_1(f(z, u)) \right\}$$

- Now suppose we are given $V_j(z)$, need to derive $V_{j+1}(z)$



- What is the optimal control for $j+1$ horizon?
  - Suppose available controls at time 0 are $U(z) = \{u^{(1)}, u^{(2)}\}$
  - Need to compare: $l(z, u^{(1)}) + V_j\left(f\left(z, u^{(1)}\right)\right)$ and $l(z, u^{(2)}) + V_j\left(f\left(z, u^{(2)}\right)\right)$
  - The optimal control: $\mu^*_{j+1}(z) = argmin_{u \in U(z)}\{l(z,u) + V_j(f(z,u))\}$
  - The minimum cost: $V_{j+1}(z) = \min_{u \in U(z)} \{l(z,u) + V_j(f(z,u))\} \Leftarrow$

$z^{(1)}$

$\min_x x^2+1$

$=1$

$0 = \arg\min x^2+1$

value iteration

function of state

- $\mu^*_{j+1}(z)$**: has the following two meanings**
  - the first optimal control action for a $j+1$ horizon problem with initial state z
  - the optimal control action when the system is at state z and there are j+1 steps to go

Value Iteration Algorithm

- System dynamics: $x_{k+1} = f(x_k, u_k)$ with $u_k \in U(x_k)$
- Determine u by solving optimization problem:

Minimize: $J_N(z,u) = g(x_N) + \sum_{k=0}^{N-1} l(x_k, u_k)$

subject to: control constraint $u_k \in U(x_k)$,

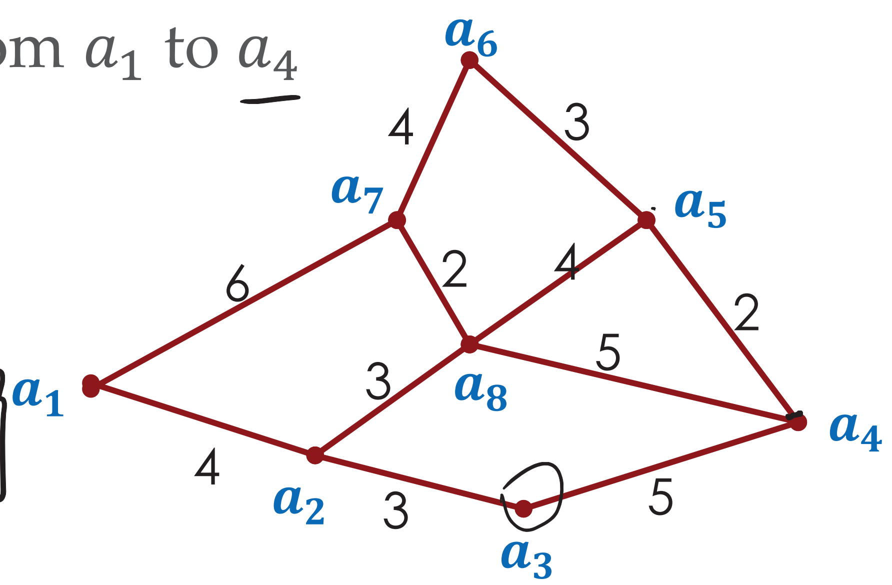
system dynamics $x_{k+1} = f(x_k, u_k)$, $x_0 = z$

- Solve problem through **value iteration**: (namely, iteratively compute the value function for 0-horizon, 1-horizon, …, N-horizon problems)

- **Step 0**: (0-horizon): $V_0(z) = g(z)$
- **Step *j***: given $V_j(z)$ and the optimal control laws $\mu_j^*(z), \mu_{j-1}^*, (z)\ \ldots, \mu_0^*(z)$ for the remaining j steps, compute:
  - $V_{j+1}(z) = \min_{u \in U(z)} \{l(z,u) + V_j(f(z,u))\}$
  - $\mu_{j+1}^*(z) = argmin_{u \in U(z)} \{l(z,u) + V_j(f(z,u))\}$
- $j \leftarrow j + 1$, until $j = N$

- Value iteration algorithm output:
  - Value functions: $V_0(z), \ldots, V_N(z)$
  - Optimal control laws: $\mu_j^*(z) = argmin_{u \in U(z)} \{l(z,u) + V_{j-1} f(z,u)\}, j = 1, \ldots, N$
    - The optimal control action if sys is at z and there are j steps to go
- How to use these control laws?
  - Optimal system trajectory
  - Time 0: $x_0 = \hat{x}$ → control action: $u_0^* = \mu_N^*(\hat{x})$
  - Time 1: $x_1^* = f(\hat{x}, u_0^*)$ → control action: $u_1^* = \mu_{N-1}^*(x_1^*)$
  - Time 2: $x_2^* = f(x_1^*, u_1^*)$ → control action: $u_2^* = \mu_{N-2}^*(x_2^*)$
  - $\vdots$
  - Time $N-1$: $x_{N-1}^* = f(x_{N-2}^*, u_{N-2}^*)$ → control action: $u_{N-1}^* = \mu_1(x_{N-1}^*)$
  - Time N: $x_N^* = f(x_{N-1}^*, u_{N-1}^*)$
- In general: at time k: **optimal control** $\boldsymbol{u_k^* = \mu_{N-k}^*(x_k^*)}$

- **Example:** Find shortest path from a_1 to a_4
  - $V_0(z) = g(z) = \begin{cases} 0 & \text{if } z = a_4 \\ \infty & \text{if } z \neq a_4 \end{cases}$
  - $V_1(z) = \min_{u \in \mathcal{U}(z)} \{ l(z,u) + V_0(f(z,u)) \}$



$$V_1(a_3) = \min \{ \underbrace{l(a_3,a_2)}_{3} + \underbrace{V_0(a_2)}_{\infty},\ \underbrace{l(a_3,a_4)}_{5} + \underbrace{V_0(a_4)}_{0} \}$$

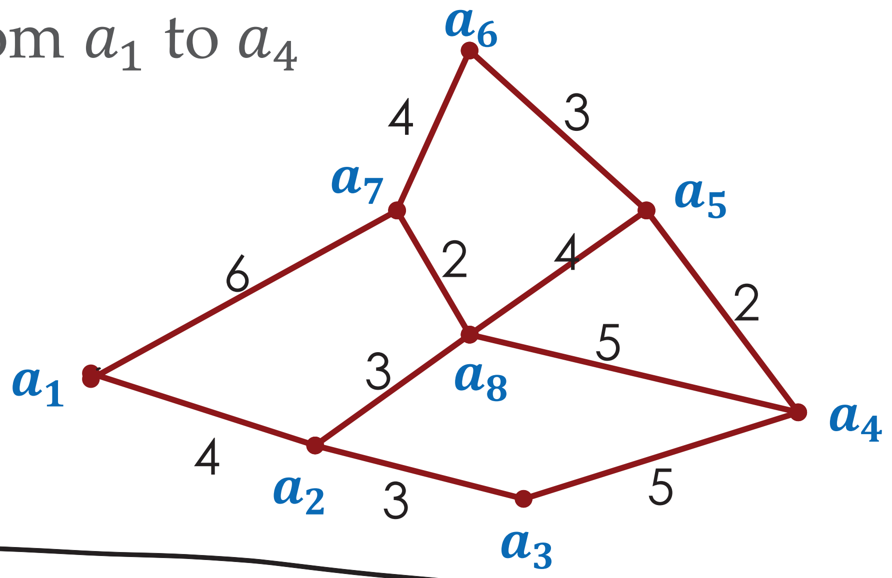
$$= 5$$

$$u_1^*(a_3) = a_4$$

$V_1(a_5) = 2$, $V_1(a_8) = 5$, $V_1(a_2) = \min\{\infty, \infty, \infty\} = \infty$

$$u_1^*(a_2) = a_2$$

- **Example:** Find shortest path from a_1 to a_4



$$V_1(z) = \begin{cases} 2, & \text{if } z = a_5 \\ 5 & \text{if } z = a_8 \\ 5, & \text{if } z = a_3 \\ 0, & \text{if } z = a_4 \\ \infty, & \text{otherwise} \end{cases}$$

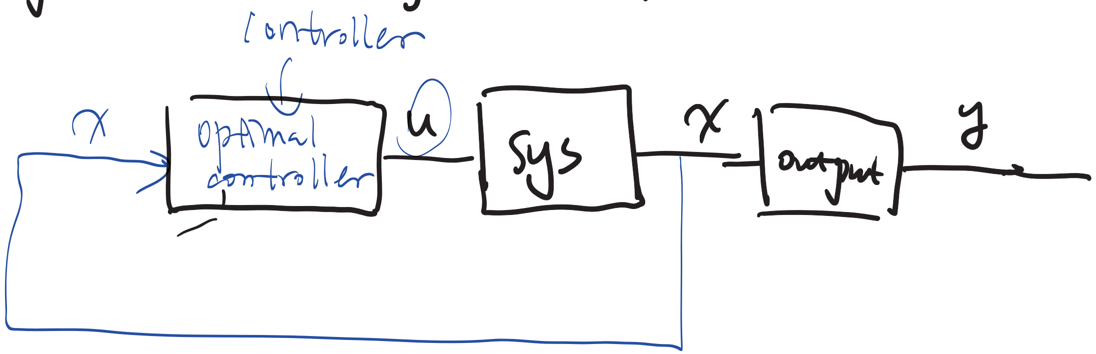
$$V_2(z) = \min_{u \in U(z)} \{ l(z,u) + V_1(f(z,u)) \}$$

e.g. $V_2(a_2) = \min \{ \underbrace{l(a_2,a_1)}_{4} + \underbrace{V_1(a_1)}_{\infty},\ \underbrace{l(a_2,a_3)}_{3} + \underbrace{V_1(a_3)}_{5},\ \overbrace{l(a_2,a_8)}^{3} + \underbrace{V_1(a_8)}_{5} \}$

$= 8$

$\mu_2^*(a_2) = a_3$ or a_8 / we need:

- Recall: optimal control formulation



· system:

$$\begin{cases} x_{k+1} = f(x_k, u_k) \\ y_k = h(x_k, u_k) \end{cases}$$

optimal control: Encode your control objective into a "cost" function then find the "optimal" control $u^* = u_0^*, u_1^* \cdots$ to minimize the cost

General cost func

$$J_N(z,u) = \sum_{k=0}^{N-1} \underbrace{l(x_k, u_k)}_{\text{running cost}} + \underbrace{g(x_N)}_{\text{terminal}}$$

$$V_N^*(z) = \min_u J_N(z,u)$$

subject to:

$$\textcircled{C}\ \begin{cases} x_{k+1} = f(x_k, u_k) \\ u_k \in U(x_k) \\ x_k \in X \end{cases}$$

j-horizon problem: $V_j(z) = \min_{u_0, \dots, u_{j-1}} J_j(z, u)$

subj to. $\textcircled{C}$

start from $j=0$, $V_0(z) = g(z)$

Value iteration: $\boxed{V_{j+1}(z) = \min_{u \in U(z)} \left\{ l(z,u) + V_j(f(z,u)) \right\}}$

$\underline{\mu^*_{j+1}(z)} = \operatorname{argmin}_{u \in U(z)} \left\{ l(z,u) + V_j(f(z,u)) \right\}$

N-horizon

$x_0 \xrightarrow{u_0^*} x_1^* \xrightarrow{u_1^*} x_2^* \longrightarrow \dots$

$u_0^* = \mu_N^*(x_0)$, $u_1^* = \mu_{N-1}^*(x_1^*)$

- $X = \{1, 2 \cdots 8\}$, for loop

eg $X = (0, 40)$

griding with grid size 100

for loop for 100 state

eg $X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, $\left.\begin{array}{l} x_1 \in [0, 40) \\ x_2 \in [0, 40) \end{array}\right\} \Rightarrow$ # grids $= 100^2 = 10000$

$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{10} \end{bmatrix}$, $\Rightarrow$ # grids points $= 100^{10}$

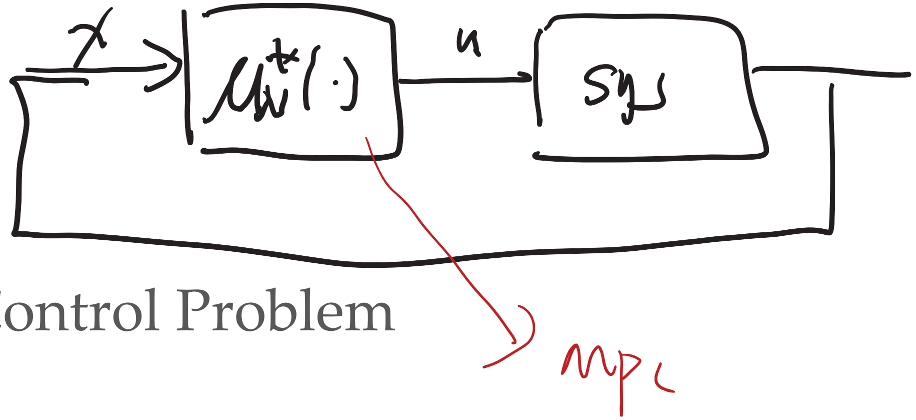
complexity grows exponentially. $\Leftarrow$ <u>curse of dimensionality</u>

- Exception: LQR $\Leftarrow$ $\begin{cases} \text{no need to do gridding} \\ \text{we have analytical form for} \end{cases}$

$V_j(z)$ and $\mu_j^{*}(z)$

- **Outline**

$u_N^*(x_0)$



  - General Discrete-Time Optimal Control Problem
  - Short Introduction to Dynamic Programming
  - **Linear Quadratic Regulator**

- **Linear Quadratic Regulator (LQR):**
  - N-horizon LQR: Find control sequence $u_0, u_1, \dots, u_{N-1}$ to minimize $J_N(z, u)$, subject to **linear dynamics constraints**:

$$x_{k+1} = Ax_k + Bu_k, \qquad x_0 = z \Leftarrow \quad x_{k+1} = f(x_k, u_k)$$

where : $J_N(x_0, u) = \underbrace{x_N^T Q_f x_N}_{g(x_N)} + \sum_{k=0}^{N-1} [\underbrace{x_k^T Q x_k \ + u_k^T R u_k}_{l(x_k, u_k) = x_k^T Q x_k + u_k^T R u_k}]$

  - Infinite-horizon LQR: Find control sequence $u_0, u_1, \dots,$ to minimize $J_\infty(x_0, u)$ subject to linear dynamics constraints: $x_{k+1} = Ax_k + Bu_k$

where $J_\infty(x_0, u) = \sum_{k=0}^{\infty} [x_k^T Q x_k^T + u_k^T R u_k^T]$

$\mathbb{R}^n \to \mathbb{R}$

- $z^T P z$: quadratic cost term, penalizing deviation from 0, e.g.:
  - if $P = I$, then $z^T P z = ||z||^2$
  - if $P = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$, then $z^T P z = z_1^2 + 2z_2^2$, penalizes z_2 more than z_1

$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \qquad z^T P z$

- Solution of LQR using Dynamic Programming (DP)
  - $V_0(z) = z^T Q_f z$ $= g(z)$
  - Suppose at j-horizon value function is: $\boxed{V_j(z) = z^T P_j z}$ holds for $j=0$

    Compute $(j+1)$-horizon value function using DP

$$\boxed{V_{j+1}(z) = \min_{u \in R^m} \{ l(z,u) + V_j(f(z,u)) \}} \Leftarrow \text{value iteration}$$
$$= \min_{u \in R^m} \{ z^T Q z + u^T R u + (Az+Bu)^T P_j (Az+Bu) \}$$
$$= \min_{u \in R^m} \{ u^T (R + B^T P_j B) u + 2 z^T A^T P_j B u + z^T (Q + A^T P_j A) z \}$$
$$\triangleq \min_{u \in R^m} h(u)$$

$h : \mathbb{R}^m \to \mathbb{R}$

- $\frac{\partial h}{\partial u}(u) = 2u^T (R + B^T P_j B) + 2 z^T A^T P_j B = 0$

Optimizer: $\mu_{j+1}^*(z) = -\overbrace{\boxed{(R + B^T P_j B)^{-1} B^T P_j A}}^{K_{j+1}} z \triangleq -K_{j+1} z$

where $K_{j+1} = (R + B^T P_j B)^{-1} B^T P_j A$

- Derivation (cont.)
  - $V_{j+1}(z) = \min_{u \in R^m} h(u) = h(u^*)$

$$= \left(-K_j z\right)^T \left(R + B^T P_j B\right)\left(-K_j z\right) + 2z^T A^T P_j B\left(-K_j z\right) + z^T\left(Q + A^T P_j A\right)z$$

$$= \boxed{z^T \left(Q + A^T P_j A - A^T P_j B\left(R + B^T P_j B\right)^{-1} B^T P_j A\right) z}$$

$$\triangleq z^T P_{j+1} z$$

where $P_{j+1} \triangleq \underline{Q + A^T P_j A - A^T P_j B\left(R + B^T P_j B\right)^{-1} B^T P_j A}$

Riccati recursion

- If at time k, the state is at x_k, then the optimal control applied at time k is

$$u_k^* = \underline{\mu_{N-k}^*(x_k)} = -K_{N-k} x_k$$

$P_0 \Leftarrow Q_N \rightarrow P_1 \rightarrow P_2 \cdots P_N$

- **Summary of LQR**
  - Value function is given by: $V_j(z) = z^T P_j z$, where P_j is given by the so-called Riccati recursion:

$$\boxed{P_{j+1} = Q + A^T P_j A - A^T P_j B\left(R + B^T P_j B\right)^{-1} B^T P_j A}$$

  - To compute the LQR controller:
    - Start from initial matrix: $P_0 = Q_f$
    - Riccati recursion: $P_j \leftarrow P_{j-1}$
    - Compute optimal feedback gain: $K_j = \left(R + B^T P_{j-1} B\right)^{-1} B^T P_{j-1} A$

$V_j(z) \leftarrow \mu_j^*($

$z^T P_j z \leftarrow$

  - Apply LQR controller:
    - Start from an IC: x_0
    - For $k = 0, \dots, N-1$
      - Compute: $u_k^* = -K_{N-k} x_k^*$,
      - $x_{k+1}^* = A x_k^* + B u_k^*$

- Infinite horizon case:
  - It can be proved that if (A, B) is controllable and (A, G) is observable, where $Q = G^T G$, then as $N \to \infty$,
  - $P_j \to P^*$, and $K_j \to K^*$, with $|\lambda(A - BK^*)| < 1$
  - P^* and K^* satisfy the algebraic equations:

'a Riccati

$$P^* = A^T \left[P^* - P^* B (R + B^T P^* B)^{-1} B^T P^*\right] A + Q$$

$$K^* = (R + B^T P^* B)^{-1} B^T P^* A$$

Coding Example