Aalto University

Dynamic programming

CHEM-E7225 (was E7195), 2023

Francesco Corona (¬_¬)

Chemical and Metallurgical Engineering
School of Chemical Engineering

Multi-stage optimisation

Dynamic programming

Optimising multi-stage functions

Consider the set of decision variables w, x, y, and z and the following objective function

$$\underbrace{f(w,x)}_{0} + \underbrace{g(x,y)}_{1} + \underbrace{h(y,z)}_{2}$$

Each stage-cost function in the sum depends only on the adjacent pairs of variables

---

Consider the case in which w is known, and we want to solve the optimisation problem

$$\min_{x,y,z|w} \quad f(x|w) + g(x,y) + h(y,z)$$

One possibility would be to jointly optimise for all the three decision variables (x, y, z)

⇝ This solution is certainly valid, but it does not exploit the problem structure

We could, alternatively, solve a sequence of single-variable optimisation problems

$$\underbrace{\min_{x|w} \left(f(x|w) + \underbrace{\min_{y} \left(g(x,y) + \underbrace{\min_{z} \quad h(y,z)}_{\text{1st}} \right)}_{\text{2nd}} \right)}_{\text{3rd}}$$

Optimising multi-stage functions (cont.)

$$\min_{x|w} \left(f(x|w) + \min_{y} \left(g(x,y) + \underbrace{\min_{z} \quad h(y,z)}_{\text{1st}} \right) \right)$$

Starting from the innermost optimisation problem, we solve with respect to variable z

$$\min_{z} \quad h(y,z)$$

We obtain the solution for z and get the optimal value-function in terms of variable y

$$h^*(y) = \min_{z} \quad h(y,z) \qquad \text{(optimal value-function)}$$

$$z^*(y) = \arg\min_{z} \quad h(y,z) \qquad \text{(minimiser)}$$

Optimising multi-stage functions (cont.)

$$\min_{x|w} \left(f(x|w) + \underbrace{\min_{y} \left(g(x,y) + \underbrace{\min_{z} \; h(y,z)}_{h^*(y)} \right)}_{\text{2nd}} \right)$$

Proceeding with the next optimisation problem, we solve it with respect to variable y

$$\min_{y} \quad g(x,y) + h^*(y)$$

We obtain the solution for y and get the optimal value-function in terms of variable x

$$g^*(x) = \min_{y} \quad g(x,y) + h^*(y) \qquad \text{(optimal value-function)}$$

$$y^*(x) = \arg\min_{y} \quad g(x,y) + h^*(y) \qquad \text{(minimiser)}$$

Optimising multi-stage functions (cont.)

$$\underbrace{\min_{x|w} \left(f(x|w) + \underbrace{\min_{y} \left(g(x,y) + \underbrace{\min_{z} \quad h(y,z)}_{h^*(y)} \right)}_{g^*(x)} \right)}_{\text{3rd}}$$

With the third and final optimisation problem, we solve it with respect to variable x

$$\min_{x|w} \quad f(x|w) + g^*(x)$$

We obtain the solution for x and get the optimal value-function in terms of value w

$$f^*(w) = \min_{x} \quad f(x|w) + g^*(x) \qquad \text{(optimal function value)}$$

$$x^*(w) = \arg\min_{x} \quad f(x|w) + g^*(x) \qquad \text{(minimiser, solution)}$$

Because w is fixed (we know its value), we have that $x^*(w)$ is completely determined

Optimising multi-stage functions (cont.)

$$\underbrace{\min_{x|w} \left(f(x|w) + \underbrace{\min_{y} \left(g(x,y) + \underbrace{\min_{z} \quad h(y,z)}_{h^*(y) \text{ at } z^*(y)} \right)}_{g^*(x) \text{ at } y^*(x)} \right)}_{f^*(w) \text{ at } x^*(w)}$$

Because we know $x^*(w)$, we have that $y^*(x^*(w))$ and $z^*(y^*(x^*(w)))$ are also known

$$\begin{aligned} \widetilde{y}^*(w) &= y^*(x^*(w)) \\ \widetilde{z}^*(w) &= z^*(\widetilde{y}^*(w)) \\ &= z^*(y^*(x^*(w))) \end{aligned}$$

Similarly, the optimal value of the objective function are computed by substitution

$$\underbrace{f^*(w)} + \underbrace{g^*(x^*(w))} + \underbrace{h^*(y^*(x^*(w)), z^*(y^*(x^*(w))))}$$

Optimising multi-stage functions (cont.)

This method to solve (unconstrained) multi-state optimisation problems can be an alternative approach to solve optimal control problems (**backward dynamic programming**)

- ⇝ The decision variables are determined, not jointly, but in reverse order

The solutions expressed as functions, of the variables to be optimised at the next stage

---

Its application is easiest for discrete-time systems with discrete state- and action-spaces

- ⇝ With continuous spaces, the applicability is achieved by discretisation
- ⇝ In continuous-time the problem is formulated as a PDE, the HJBE
- ⇝ (The Hamilton-Jacobi-Bellmann equation)

Discrete state- and action-spaces

Dynamic programming

Discrete state- and action-spaces

Consider the nonlinear dynamic equation of a discrete-time state-space model

$$x_{k+1} = f(x_k, u_k)$$

Then, suppose that the state- and the action-space be discrete and finite

$$\begin{aligned} x_k &\in \mathcal{X}, \quad \text{with } |\mathcal{X}| = N_{\mathcal{X}} \\ u_k &\in \mathcal{U}, \quad \text{with } |\mathcal{U}| = N_{\mathcal{U}} \end{aligned}$$

Based on the discrete dynamics, we formulate the optimal control problem

$$\begin{aligned} \min_{\substack{x_0, x_1, \ldots, x_{K-1}, x_K \\ u_0, u_1, \ldots, u_{K-1}}} \quad & E(x_K) + \sum_{k=0}^{K-1} L(x_k, u_k) \\ \text{subject to} \quad & f(x_k, u_k) - x_{k+1} = 0, \qquad k = 0, 1, \ldots, K-1 \\ & \overline{x}_0 - x_0 = 0 \end{aligned}$$

The initial state x_0 is assumed to be known, some fixed value $\overline{x}_0$

Discrete state- and action-spaces (cont.)

$$\begin{aligned}
\min_{\substack{x_0, x_1, \ldots, x_{K-1}, x_K \\ u_0, u_1, \ldots, u_{K-1}}} \quad & E(x_K) + \sum_{k=0}^{K-1} L(x_k, u_k) \\
\text{subject to} \quad & f(x_k, u_k) - x_{k+1} = 0, \qquad k = 0, 1, \ldots, K-1 \\
& \overline{x}_0 - x_0 = 0
\end{aligned}$$

Controls $\{u_k\}_{k=0}^{K-1}$ are the only decision variables of the optimisation (if x_0 is known)

We know that the state variables can be eliminated by forward-simulation

$$\begin{aligned}
\overline{x}_0 &= x_0 \\
\overline{x}_1(x_0, u_0) &= f(x_0, u_0) \\
\overline{x}_2(x_0, u_0, u_1) &= f(x_1, u_1) \\
&= f(f(x_0, u_0), u_1) \\
\overline{x}_3(x_0, u_0, u_1, u_2) &= f(x_2, u_2) \\
&= f(f(f(x_0, u_0), u_1), u_2) \\
\cdots &= \cdots \\
\overline{x}_K(x_0, u_0, u_1, \ldots, u_{K-2}, u_{K-1}) &= f(x_{K-1}, u_{K-1}) \\
&= f(f(\cdots f(x_0, u_0), u_{K-2}), u_{K-1})
\end{aligned}$$

Discrete state- and action-spaces (cont.)

$$\begin{aligned} \min_{\substack{x_0, x_1, \dots, x_{K-1}, x_K \\ u_0, u_1, \dots, u_{K-1}}} \quad & E(x_K) + \sum_{k=0}^{K-1} L(x_k, u_k) \\ \text{subject to} \quad & f(x_k, u_k) - x_{k+1} = 0, \qquad k = 0, 1, \dots, K-1 \\ & \overline{x}_0 - x_0 = 0 \end{aligned}$$

This formulation of discrete optimal control problem does not include path constraints

Path constraints can be implicitly included by letting stage-costs be equal to infinity

$\rightsquigarrow$ For any infeasible pair $(\tilde{x}_k, \tilde{u}_k)$, we have that $L(\tilde{x}_k, \tilde{u}_k) = \infty$

To include these, as well as other, inequality constraints we have

$$L : \mathcal{X} \times \mathcal{U} \rightarrow \mathcal{R} \cup \infty$$

Discrete state- and action-spaces (cont.)

$$\begin{aligned}
\min_{\substack{x_0, x_1, \ldots, x_{K-1}, x_K \\ u_0, u_1, \ldots, u_{K-1}}} \quad & E(x_K) + \sum_{k=0}^{K-1} L(x_k, u_k) \\
\text{subject to} \quad & f(x_k, u_k) - x_{k+1} = 0, \qquad k = 0, 1, \ldots, K-1 \\
& \overline{x}_0 - x_0 = 0
\end{aligned}$$

As each u_k can only take on one of $N_{\mathcal{U}}$ values, there are $N_{\mathcal{U}}^K$ possible control sequences

$$\underbrace{\underbrace{N_{\mathcal{U}}}_{\text{Stage } 0} \times \underbrace{N_{\mathcal{U}}}_{\text{Stage } 1} \times \cdots \times \underbrace{N_{\mathcal{U}}}_{\text{stage } K-2} \times \underbrace{N_{\mathcal{U}}}_{\text{stage } K-1}}_{K \text{ times}}$$

Each possible control sequence corresponds to a different trajectory $\{\{x_k, u_k\}_{k=0}^{K-1} \cup x_K\}$

- ⇝ Each such trajectory associates with a specific value of the objective function
- ⇝ The optimal solution, the sequence(s) of smallest function value

Discrete state- and action-spaces (cont.)

$$\begin{aligned} \min_{\substack{x_0, x_1, \dots, x_{K-1}, x_K \\ u_0, u_1, \dots, u_{K-1}}} \quad & E(x_K) + \sum_{k=0}^{K-1} L(x_k, u_k) \\ \text{subject to} \quad & f(x_k, u_k) - x_{k+1} = 0, \qquad k = 0, 1, \dots, K-1 \\ & \overline{x}_0 - x_0 = 0 \end{aligned}$$

Naive enumeration of all trajectories has a complexity that grows exponentially in K

$$\underbrace{N_{\mathcal{U}} \times N_{\mathcal{U}} \times \cdots \times N_{\mathcal{U}}}_{K \text{ times}}$$

The idea behind dynamic programming is to approach the enumeration task differently

---

We start by noting that each sub-trajectory of an optimal trajectory must be optimal

- ⇝ If $\{\{x_k^*, u_k^*\}_{k=0}^{K-1} \cup x_K^*\}$ is optimal, then any $\{\{x_k^*, u_k^*\}_{k>0}^{K-1} \cup x_K^*\}$ is optimal
- ⇝ This property is known as the **Bellman's principle of optimality**

Discrete state- and action-spaces (cont.)

We define the **value-function** or **cost-to-go** as the optimal cost that would be attained if, at time k, from state $\overline{x}_k \in \mathcal{X}$, we would solve the shorter optimal control problem

$$J_k(\overline{x}_k) = \min_{\substack{x_k, x_{k+1}, \dots, x_{K-1}, x_K \\ u_k, u_{k+1}, \dots, u_{K-1}}} \quad E(x_K) + \sum_{i=k}^{K-1} L(x_i, u_i)$$
$$\text{s.t.} \quad f(x_i, u_i) - x_{i+1} = 0, \quad i = k, k+1, \dots, K-1$$
$$\overline{x}_k - x_k = 0$$

Function $J_k : \mathcal{X} \rightarrow \mathcal{R} \cup \infty$ summarises the cost-to-go from x_k to the end of the horizon

- Starting from some initial state $\overline{x}_k$, under the optimal actions $\{u_i^*\}_{i=k}^{K-1}$

As there is a finite number $N_{\mathcal{X}}$ of possible initial states $\overline{x}_k$, at each stage k, we have

$$J_k\left(x_k^{(1)}\right)$$
$$\vdots$$
$$J_k\left(x_k^{(N_{\mathcal{X}})}\right)$$

Discrete state- and action-spaces (cont.)

The **Bellman equation**

The principle of optimality states that, for any $k \in \{0, \dots, K-1\}$, the following holds

$$J_k(\overline{x}_k) = \min_u \Big(L(\overline{x}_k, u) + J_{k+1}\Big(\underbrace{f(\overline{x}_k, u)}_{\overline{x}_{k+1}}\Big)\Big)$$
$$= \min_u \Big(L(\overline{x}_k, u) + J_{k+1}(\overline{x}_{k+1})\Big)$$

Similarly, we have that, at $k+1$, the following holds

$$J_k(\overline{x}_{k+1}) = \min_u \Big(L(\overline{x}_{k+1}, u) + J_{k+2}\Big(\underbrace{f(\overline{x}_{k+1}, u)}_{\overline{x}_{k+2}}\Big)\Big)$$
$$= \min_u \Big(L(\overline{x}_{k+1}, u) + J_{k+2}(\overline{x}_{k+2})\Big)$$

Discrete state- and action-spaces (cont.)

All the way to K, when there is no longer any time to apply any control action u_K

- The stage-cost at K then equals the terminal cost $E(x_K)$

$$J_K\left(\overline{x}_K\right) = \min_{u=u_K} \left(\underbrace{L\left(\overline{x}_K, u\right)}_{E(\overline{x}_K)} + \cancel{J_{K+1}\left(\underbrace{f\left(\overline{x}_K, u\right)}_{\overline{x}_{K+1}}\right)}\right)$$
$$= E(\overline{x}_K)$$

At the preceding stages, we have

$$J_{K-1}\left(\overline{x}_{K-1}\right) = \min_{u=u_{K-1}} \left(L\left(\overline{x}_{K-1}, u\right) + J_K\left(\underbrace{f\left(\overline{x}_{K-1}, u\right)}_{\overline{x}_K}\right)\right)$$
$$= \min_{u=u_{K-1}} \left(L\left(\overline{x}_{K-1}, u\right) + \underbrace{J_K\left(\overline{x}_K\right)}_{E(\overline{x}_K)}\right)$$

$$J_{K-2}\left(\overline{x}_{K-2}\right) = \min_{u=u_{K-2}} \left(L\left(\overline{x}_{K-2}, u\right) + J_{K-1}\left(\underbrace{f\left(\overline{x}_{K-2}, u\right)}_{\overline{x}_{K-1}}\right)\right)$$
$$= \min_{u=u_{K-2}} \left(L\left(\overline{x}_{K-2}, u\right) + J_{K-1}\left(\overline{x}_{K-1}\right)\right)$$

Discrete state- and action-spaces (cont.)

Remember the formulation of the optimal control problem, the objective is multi-stage

$$\begin{aligned}
\min_{\substack{x_0, x_1, \dots, x_K \\ u_0, u_1, \dots, u_{K-1}}} \quad & E(x_K) + \sum_{k=0}^{K-1} L(x_k, u_k) \\
\text{subject to} \quad & f(x_k, u_k) - x_{k+1} = 0, \qquad k = 0, 1, \dots, K-1 \\
& \overline{x}_0 - x_0 = 0
\end{aligned}$$

The initial state x_0 is fixed at $\overline{x}_0$, the controls $\{u_k\}_{k=0}^{K-1}$ are the actual decision variables

That is, we have the multi-stage objective function

$$\begin{aligned}
\min_{\substack{x_0, \\ u_0, u_1, \dots, u_{K-1}}} \quad & E(x_K) + \sum_{k=0}^{K-1} L(x_k, u_k) \\
\text{subject to} \quad & f(x_k, u_k) - x_{k+1} = 0, \qquad k = 0, 1, \dots, K-1 \\
& \overline{x}_0 - x_0 = 0
\end{aligned}$$

Discrete state- and action-spaces (cont.)

$$\min_{\substack{x_0,\\ u_0,u_1,\dots,u_{K-1}}} \underbrace{L(x_0,u_0)+L(x_1,u_1)+\cdots+L(x_{K-2},u_{K-2})+L(x_{K-1},u_{K-1})}_{\sum_{k=0}^{K-1} L(x_k,u_k)} + E(x_K)$$

$$\text{s.t. } f(x_k,u_k)-x_{k+1}=0, \quad k=0,1,\dots,K-1$$
$$\overline{x}_0 - x_0 = 0$$

With the explicit dependence only on the true decision variables, we have

$$\min_{\substack{x_0,\\ u_0,u_1,\dots,u_{K-1}}} L(x_0,u_0) + L\Big(\underbrace{x_0,u_0}_{x_1},u_1\Big) + L\Big(\underbrace{x_0,u_0,u_1}_{x_2},u_2\Big) + \cdots$$
$$+ L\Big(\underbrace{x_0,u_0 \rightsquigarrow u_{K-3}}_{x_{K-2}},u_{K-2}\Big) + L\Big(\underbrace{x_0,u_0 \rightsquigarrow u_{K-2}}_{x_{K-1}},u_{K-1}\Big)$$
$$+ E(x_K)$$
$$\text{s.t. } \overline{x}_0 - x_0 = 0$$

Discrete state- and action-spaces (cont.)

Furthermore, we can remove the (initial) equality constraint and write

$$\min_{u_0 \rightsquigarrow u_{K-1} | x_0} L_0 \left(u_0 | x_0\right) + L_1 \left(u_0 \rightsquigarrow u_1 | x_0\right) + L_2 \left(u_0 \rightsquigarrow u_2 | x_0\right) + \cdots$$
$$\cdots + L_{K-2} \left(u_0 \rightsquigarrow u_{K-2} | x_0\right) + L_{K-1} \left(u_0 \rightsquigarrow u_{K-1} | x_0\right) + E\left(x_K\right)$$

We can solve the equivalent problem as multi-stage optimisation

$$\min_{u_0 | x_0} \Bigg(L_0(u_0 | x_0) + \min_{u_1} \Bigg(L_1(u_0 \rightsquigarrow u_1) + \min_{u_2} \Bigg(L_2(u_0 \rightsquigarrow u_2) + \cdots$$
$$\cdots + \min_{u_{K-2}} \Bigg(L_{K-2}(u_0 \rightsquigarrow u_{K-2}) + \min_{u_{K-1}} L_{K-1}(u_0 \rightsquigarrow u_{K-1})\Bigg)\Bigg)\Bigg) + \min_{u_K} E(x_K)\Bigg)$$

Discrete state- and action-spaces (cont.)

We know that this backward recursion is denoted as the **dynamic programming recursion**

$$u_k^*(x_k) = \arg\min_u L(x_k, u) + J_{k+1}(f(x_k, u))$$

Once all the value-functions J_k are computed, we also have the **optimal feedback control**

$$x_{k+1} = f(x_k, u_k^*(x_k)), \quad k = 0, 1, \ldots, K-1$$

The computationally demanding step is the generation of the K value functions J_k

- Each recursion step requires to test $N_{\mathcal{U}}$ controls, for each of the $N_{\mathcal{X}}$ states
- Each recursion requires computing $f(x_k, u)$ and $L(x_k, u)$

The overall complexity is thus $K \times (N_{\mathcal{X}} \times N_{\mathcal{U}})$

Discrete state- and action-spaces (cont.)

One of the main advantages of the dynamic programming approach to optimal control is the possibility to be extended to continuous state- and action-spaces, by discretisation

- No assumptions on differentiability of the dynamics or convexity of the objective

However, it is important to notice that for a N_x dimensional state-space discretised along each dimension using M_x intervals, the total number of grid points is $N_{\mathcal{X}} = M_x^{N_x}$

- That is, complexity grows exponential with the dimension of the state-space

An example

Discrete state and action spaces

An example

Consider a total stage-cost given as sum of the state stage-cost and control stage-cost

$$L_k\left(x_k, u_k\right) = L_x^k\left(x_k\right) + L_u^k\left(x_k, u_k\right)$$

The stage-cost for the states (the positions on a (4×3) board)

- The target state is located in the position $(2, 2)$
- The associated state-cost (per stage) is zero

$$\underbrace{\begin{array}{c|cc} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{array}}_{\mathcal{X}} \quad \rightsquigarrow \quad \underbrace{\begin{array}{c|cc} 5 & 5 & 5 \\ 5 & \mathbf{0} & 5 \\ 5 & 5 & 5 \\ 5 & 5 & 5 \end{array}}_{L_x^k(x_k \in \mathcal{X})}$$

The stage-cost for the controls (the 9 possible 'moves')

- The control-cost per stage is one, or zero

$$\underbrace{\begin{array}{ccc} \nwarrow & \uparrow & \nearrow \\ \leftarrow & \cdot & \rightarrow \\ \swarrow & \downarrow & \searrow \end{array}}_{\mathcal{U}} \quad \rightsquigarrow \quad \underbrace{\begin{array}{ccc} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{array}}_{L_u^k(x_k, u_k \in \mathcal{U})}$$

An example (cont.)

The policy (control law) π specifies the action that we will perform at time step k

- The control policy is a function of the state (state-feedback), at stage k

$$\pi(x_k) = u_k(x_k)$$

A random example of a possible control policy

$$\pi(x_k) \quad = \quad \begin{array}{cccc} \cdot & | & \nwarrow & \leftarrow \\ \uparrow & | & \downarrow & \rightarrow \\ \uparrow & | & \cdot & \swarrow \\ \leftarrow & & \nearrow & \cdot \end{array}$$

At k, the objective is to find the policy that minimises the cost-to-go

$$\sum_{k}^{K} L_k \left(x_k, \underbrace{u_k}_{\pi(x_k)} \right)$$

The value-function of the control policy at k quantifies the goodness of the policy when at x_K

$$V_\pi(x_k) = L_k \left(x_k, \underbrace{u_k}_{\pi(x_k)} \right) + V_\pi(x_{k+1})$$

An example (cont.)

Stage K

At the final stage $k = K$, we have the following value-function of the policy function

$$V_\pi(x_K) = L_K\Big(x_K, \underbrace{\cancel{u_K}}_{\pi(x_K)}\Big) + \cancel{V_\pi(x_{K+1})}$$

$$= \underbrace{L_x^K(x_K) + L_u^K \cancel{\Big(x_K, \underbrace{u_K}_{\pi(x_K)}\Big)}}_{L_K(x_K, u_K)} + \cancel{V_\pi(x_{K+1})}$$

$$= \begin{array}{c|cc} 5 & 5 & 5 \\ 5 & 0 & 5 \\ 5 & 5 & 5 \\ 5 & 5 & 5 \end{array}$$

As there is no time left to apply any control $u_K = \pi(x_K)$, we have the optimal policy

$$\pi^*(x_K) = \begin{array}{c|cc} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{array}$$

An example (cont.)

We have the optimal policy,

$$\pi^*(x_K) = \begin{array}{c|cc} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{array}$$

The value-function for the optimal policy corresponds to the terminal cost $E(x_K)$

$$\begin{aligned} V_{\pi^*}(x_K) &= L_K\Bigg(x_K, \underbrace{\cancel{u_K}}_{\pi(x_K)}\Bigg) + \cancel{V_\pi(x_{K+1})} \\ &= E(x_K) \end{aligned}$$

The value of the policy,

$$V_{\pi^*}(x_K) = \begin{array}{c|cc} 5 & 5 & 5 \\ 5 & 0 & 5 \\ 5 & 5 & 5 \\ 5 & 5 & 5 \end{array}$$

The value of the optimal policy at stage K gives the total cost that would be incurred if, starting at some state $x_K \in \mathcal{X}$, the best sequence of actions would be performed

- The first optimal action of the sequence (!) was found to be 'do nothing'

An example (cont.)

According to the Bellman's principle of optimality, the optimal policy at stage $K-1$

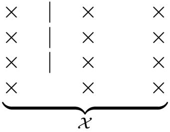
$$\pi^*(x^{K-1}) = \arg\min_{u} \; \big(L_{K-1}(x_{K-1}, u_{K-1}) + V_{\pi^*}(x_K)\big)$$

- We must compute the stage-cost $L_{K-1}(x_{K-1}, u_{K-1})$ at stage $K-1$
- We already know the value of the policy $V_{\pi^*}(x_K)$

$$V_{\pi^*}(x_K) = \begin{array}{c|cc} 5 & 5 & 5 \\ 5 & 0 & 5 \\ 5 & 5 & 5 \\ 5 & 5 & 5 \end{array}$$

An example (cont.)

Stage $K-1$



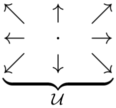
For each state $x_{K-1} \in \mathcal{X}$, compute the stage cost $L_{K-1}(x_{K-1}, u_{K-1})$, for all $u_{K-1} \in \mathcal{U}$

We add it to the optimal value-function at stage K, $V_{\pi^*}(x^K)$, and optimise

$$V_{\pi^*}(x^{K-1}) = \min_{u_{K-1}} \Big(L_{K-1}(x_{K-1}, u_{K-1}) + V_{\pi^*}(x^K) \Big)$$

From a minimisation of the value function, we compute the optimal policy to get u^*_{K-1}

$$\pi^*(x^{K-1}) = \arg\min_{u} \Big(L_{K-1}(x_{k-1}, u_{k-1}) + V_{\pi^*}(x^K) \Big)$$



An example (cont.)

$$
\begin{array}{cc|ccc}
\circ & & \times & & \times \\
\times & & \times & & \times \\
\times & & \times & & \times \\
\times & & \times & & \times
\end{array}
$$

Suppose that the system is at state $\mathcal{X}_{1,1}$ and consider control action $\uparrow$

- As a result the system stays at state $\mathcal{X}_{1,1}$

We have the total stage cost, as sum of state-cost and action-cost

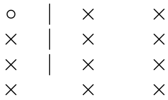
$$
\begin{aligned}
L_{K-1}(\mathcal{X}_{1,1},\uparrow) &= L_x^{K-1}(\mathcal{X}_{1,1}) + L_u^{K-1}(\mathcal{X}_{1,1},\uparrow) \\
&= 5 + 1 \\
&= 6
\end{aligned}
$$

The application of action $\uparrow$ leads to state $\mathcal{X}_{1,1}$

$$V_{\pi^*}(\mathcal{X}_{1,1}) = 5$$

We proceed similarly, for actions $\downarrow$, $\nwarrow$, $\nearrow$, $\swarrow$, $\searrow$, $\leftarrow$, $\cdot$, and $\rightarrow$ applied to state $\mathcal{X}_{1,1}$

An example (cont.)



For action $\downarrow$ applied to state $\mathcal{X}_{1,1}$, we have the total stage-cost

$$\begin{aligned} L_{K-1}(\mathcal{X}_{1,1}, \downarrow) &= J_x^{K-1}(\mathcal{X}_{1,1}) + J_u^{K-1}(\mathcal{X}_{1,1}, \downarrow) \\ &= 5 + 1 \\ &= 6 \end{aligned}$$

The application of action $\downarrow$ leads to state $\mathcal{X}_{2,1}$

$$V_{\pi^*}(\mathcal{X}_{2,1}) = 5$$

An example (cont.)

$$
\begin{array}{c|ccc}
\circ & & \times & \times \\
\times & & \times & \times \\
\times & & \times & \times \\
\times & & \times & \times
\end{array}
$$

For action $\cdot$ applied to state $\mathcal{X}_{1,1}$, we have the total stage-cost

$$
\begin{aligned}
L_{K-1}(\mathcal{X}_{1,1}, \cdot) &= J_x^{K-1}(\mathcal{X}_{1,1}) + J_u^{K-1}(\mathcal{X}_{1,1}, \cdot) \\
&= 5 + 0 \\
&= 5
\end{aligned}
$$

The application of action $\downarrow$ leads to state $\mathcal{X}_{1,1}$

$$
V_{\pi^*}(\mathcal{X}_{1,1}) = 5
$$

An example (cont.)

Summarising, for state $\mathcal{X}_{1,1}$

- At stage $K-1$

$$
\begin{aligned}
L_{K-1}(\mathcal{X}_{1,1}, \uparrow) + V_{\pi^*}(\mathcal{X}_{1,1}) &= 6 + 5 \\
&= 11 \\
L_{K-1}(\mathcal{X}_{1,1}, \nwarrow) + V_{\pi^*}(\mathcal{X}_{1,1}) &= 6 + 5 \\
&= 11 \\
L_{K-1}(\mathcal{X}_{1,1}, \nearrow) + V_{\pi^*}(\mathcal{X}_{1,1}) &= 6 + 5 \\
&= 11 \\
L_{K-1}(\mathcal{X}_{1,1}, \swarrow) + V_{\pi^*}(\mathcal{X}_{1,1}) &= 6 + 5 \\
&= 11 \\
L_{K-1}(\mathcal{X}_{1,1}, \searrow) + V_{\pi^*}(\mathcal{X}_{1,1}) &= 6 + 5 \\
&= 11 \\
L_{K-1}(\mathcal{X}_{1,1}, \leftarrow) + V_{\pi^*}(\mathcal{X}_{1,1}) &= 6 + 5 \\
&= 11 \\
L_{K-1}(\mathcal{X}_{1,1}, \rightarrow) + V_{\pi^*}(\mathcal{X}_{1,1}) &= 6 + 5 \\
&= 11 \\
L_{K-1}(\mathcal{X}_{1,1}, \downarrow) + V_{\pi^*}(\mathcal{X}_{2,1}) &= 6 + 5 \\
&= 11 \\
L_{K-1}(\mathcal{X}_{1,1}, \cdot) + V_{\pi^*}(\mathcal{X}_{1,1}) &= 5 + 5 \\
&= 10
\end{aligned}
$$

An example (cont.)

The optimal action that we can do when at state $\mathcal{X}_{1,1}$ at stage $K-1$ is to not move, $\cdot$

$$\pi^*(\mathcal{X}_{1,1}) = \begin{array}{c|cc} \cdot & - & - \\ - & - & - \\ - & - & - \\ - & - & - \end{array}$$

The value of the optimal action, at stage $K-1$

$$V_{\pi^*}(x_{K-1}) = \begin{array}{c|cc} 10 & - & - \\ - & - & - \\ - & - & - \\ - & - & - \end{array}$$

The value-function $V_{\pi^*}(\mathcal{X}_{1,1})$ gives the cost that would be incurred if, starting at state $\mathcal{X}_{1,1}$ and from that stage on, we performed the best possible sequence of actions

- The first action would be the one given by the optimal policy $\pi^*(\mathcal{X}_{1,1} \in \mathcal{X})$

An example (cont.)

Analogously for the other states $x_{K-1} \in \mathcal{X}$ at stage $K-1$, we have the optimal policy

$$\pi^*(x_{K-1} \in \mathcal{X}) = \begin{array}{cccc} \cdot & | & \downarrow & \swarrow \\ \cdot & | & \cdot & \leftarrow \\ \cdot & | & \uparrow & \nwarrow \\ \cdot & & \cdot & \cdot \end{array}$$

The value of the optimal policy, at stage $K-1$

$$V_{\pi^*}(x_{K-1} = \mathcal{X}_{1,1}) = \begin{array}{cccc} 10 & | & 6 & 6 \\ 10 & | & 0 & 6 \\ 10 & | & 6 & 6 \\ 10 & & 10 & 10 \end{array}$$

The value-function $V_{\pi^*}(x_{K-1})$ gives the cost that would be incurred if, starting at any state x_{K-1} and from that stage on, we performed the best possible sequence of actions

- The first action would be the one given by the optimal policy $\pi^*(x_{K-1} \in \mathcal{X})$

An example (cont.)

Stage $K-2$

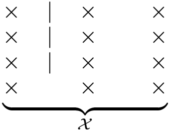
The value of the optimal policy at stage $K-1$ gives the total cost that would be incurred if, starting at state $x_{K-1} \in \mathcal{X}$, the best sequence of actions would be performed

$$V_{\pi^*}(x_{K-1}) = \begin{array}{ccc} 10 & \vert \quad 6 & 6 \\ 10 & \vert \quad 0 & 6 \\ 10 & \vert \quad 6 & 6 \\ 10 & \quad 10 & 10 \end{array}$$

The first optimal action of the sequence

$$\pi^*(x_{K-1} \in \mathcal{X}) = \begin{array}{ccc} \cdot & \vert \quad \downarrow & \swarrow \\ \cdot & \vert \quad \cdot & \leftarrow \\ \cdot & \vert \quad \uparrow & \nwarrow \\ \cdot & \quad \cdot & \cdot \end{array}$$

An example (cont.)



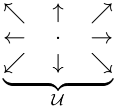
For each state $x_{K-2} \in \mathcal{X}$, compute the stage cost $L_{K-2}(x_{K-2}, u_{K-2})$ for all $u_{K-2} \in \mathcal{U}$

We add it to the optimal value-function at stage K and optimise

$$V_{\pi^*}(x_{K-2}) = \min_{u_{K-2}} \left(L_{K-2}(x_{K-2}, u_{K-2}) + V_{\pi^*}(x_{K-1}) \right)$$

From a minimisation of the value-function, we compute the optimal policy

$$\pi^*(x_{K-2}) = \arg\min_{u} \left(L_{K-2}(x_{K-2}, u_{K-2}) + V_{\pi^*}(x_{K-1}) \right)$$



An example (cont.)

At stage $K-2$, we have the optimal policy

$$\pi^*(x_{K-2} \in \mathcal{X}) = \begin{array}{cccc} \cdot & | & \downarrow & \swarrow \\ \cdot & | & \cdot & \leftarrow \\ \cdot & | & \uparrow & \nwarrow \\ \nearrow & & \uparrow & \uparrow \end{array}$$

The value of the optimal policy, at stage $K-2$

$$V_{\pi^*}(x_{K-2}) = \begin{array}{cccc} 15 & | & 6 & 6 \\ 15 & | & 0 & 6 \\ 15 & | & 6 & 6 \\ 12 & & 12 & 12 \end{array}$$

An example (cont.)

Stage $K-3$

At stage $K-3$, we have the optimal policy

$$\pi^*(x_{K-3} \in \mathcal{X}) = \begin{array}{c|cc} \cdot & \downarrow & \swarrow \\ \cdot & \cdot & \leftarrow \\ \downarrow & \uparrow & \nwarrow \\ \nearrow & \uparrow & \uparrow \end{array}$$

The value of the optimal policy, at stage $K-3$

$$V_{\pi^*}(x_{K-3}) = \begin{array}{c|cc} 20 & 6 & 6 \\ 20 & 0 & 6 \\ 18 & 6 & 6 \\ 12 & 12 & 12 \end{array}$$

An example (cont.)

Stage $K-4$

At stage $K-4$, we have the optimal policy

$$\pi^*(x_{K-4} \in \mathcal{X}) = \begin{array}{c|cc} \cdot & \downarrow & \swarrow \\ \downarrow & \cdot & \leftarrow \\ \downarrow & \uparrow & \nwarrow \\ \nearrow & \uparrow & \uparrow \end{array}$$

The value of the optimal policy, at stage $K-4$

$$V_{\pi^*}(x_{K-4}) = \begin{array}{c|cc} 25 & 6 & 6 \\ 24 & 0 & 6 \\ 18 & 6 & 6 \\ 12 & 12 & 12 \end{array}$$

An example (cont.)

Stage $K-5$

At stage $K-5$, we have the optimal policy

$$\pi^*(x_{K-5} \in \mathcal{X}) = \begin{array}{c|cc} \cdot & \downarrow & \swarrow \\ \downarrow & \cdot & \leftarrow \\ \downarrow & \uparrow & \nwarrow \\ \nearrow & \uparrow & \uparrow \end{array}$$

$$= \pi^*(x_{K-4} \in \mathcal{X})$$

The value of the optimal policy, at stage $K-4$

$$V_{\pi^*}(x_{K-4}) = \begin{array}{c|cc} 30 & 6 & 6 \\ 24 & 0 & 6 \\ 18 & 6 & 6 \\ 12 & 12 & 12 \end{array}$$

□

The linear-quadratic regulator

Dynamic programming

The linear-quadratic regulator

An important class of optimal control problems is the linear-quadratic regulator, LQR

- The controller has to take the state of the system to the origin
- The system dynamics are deterministic and linear
- The objective function is quadratic

The problem is unconstrained and the horizon for control can be finite or infinite

- Their solution can be obtained with dynamic programming

The linear-quadratic regulator (cont.)

Consider first the case in which we are interested in stabilising the system in K steps

We define an objective function to quantify the distance of the pairs (x_k, u_k) from zero

$$V(x_0, u_0, x_1, u_1, \dots, x_{K-1}, u_{K-1}, x_K) = E(x_K) + \sum_{k=0}^{K-1} L(x_k, u_k)$$

- Terminal-stage cost

$$E(x_K) = \frac{1}{2} x_K^T Q_K x_K^T$$

- Stage-cost

$$L(x_k, u_k) = \frac{1}{2}\Big(x_k^T Q x_k + u_k^T R u_k\Big)$$

The objective depends on the control sequence $\{u_k\}_{k=0}^{K-1}$ and the state sequence $\{x_k\}_{k=0}^{K}$

- We assume that the initial state x_0 is fixed and a known quantity
- Remaining states are determined by $f(x_k, u_k)$ for $\{u_k\}_{k=0}^{K_1}$

Matrices Q and Q_K are positive semi-definite, R is positive definite

- They are tuning parameters

The linear-quadratic regulator | Baby LQR

Consider a linear and time-invariant process with single state variable and single input

The system dynamics, in discrete-time

$$x_{k+1} = a x_k + b u_k, \quad \text{with } x_k, u_k \in \mathcal{R}$$

The control problem, in discrete-time

$$\underset{u_0, u_1, \dots, u_{K-1}}{\text{minimise}} \quad \underbrace{\frac{1}{2} x_K^T q_K x_K}_{E(x_K)} + \frac{1}{2} \sum_{k=0}^{K-1} \underbrace{\left(x_k^T q x_k + u_k^T r u_k \right)}_{(2) L(x_k, u_k)}$$

Consider a finite-horizon of length one ($K = 1$)

$$\underset{u_0}{\text{minimise}} \quad \frac{1}{2} x_1^T q_K x_1 + \frac{1}{2} \sum_{k=0}^{1-1} \left(x_k^T q x_k + u_k^T r u_k \right)$$

We have,

$$\underset{u_0}{\text{minimise}} \quad \frac{1}{2} \left(x_1^T q_K x_1 + x_0^T q x_0 + u_0^T r u_0 \right)$$

The linear-quadratic regulator | Baby LQR (cont.)

$$\underset{u_0}{\text{minimise}} \quad \frac{1}{2}\Big(x_1^T q_K x_1 + x_0^T q x_0 + u_0^T r u_0\Big)$$

In this simple case, we only need to (optimise to) find a single control action, u_0

- Under the dynamic constraint that $x_1 = a x_0 + b u_0$
- The initial state x_0 is fixed and known

After embedding the dynamics in the objective function, we get

$$\underset{u_0}{\text{minimise}} \ \frac{1}{2}\Bigg(\underbrace{x_1^T}_{a x_0 + b u_0} q_K \underbrace{x_1}_{a x_0 + b u_0} + x_0^T q x_0 + u_0^T r u_0\Bigg)$$

All the terms (a, b, q, q_K, r and x_0) in the cost are known, except for u_0

- Control action u_0 is the decision variable, it is a scalar

The linear-quadratic regulator | Baby LQR (cont.)

$$\underset{u_0}{\text{minimise}} \ \frac{1}{2}\left(\underbrace{x_1^T}_{a x_0 + b u_0} q_K \underbrace{x_1}_{a x_0 + b u_0} + x_0^T q x_0 + u_0^T r u_0 \right)$$

Substituting and rearranging, we have a quadratic equation u_0

$$\underset{u_0}{\text{minimise}} \ \underbrace{\frac{1}{2}\left(q x_0^2 + r u_0^2 + q_K (a x_0 + b u_0)^2 \right)}_{f(u_0)}$$

- We are interested in value u_0 that minimises this function

---

After some algebra, we see that the cost function is a parabola

$$\begin{aligned} f(u_0) &= \frac{1}{2}\left(q x_0^2 + r u_0^2 + q_K (a x_0 + b u_0)^2 \right) \\ &= \frac{1}{2}\left((q + a^2 q_K) x_0^2 + 2(b a q_K x_0) u_0 + (b^2 q_K + r) u_0^2 \right) \end{aligned}$$

We know how to locate the minimum of parabola, its vertex

The linear-quadratic regulator | Baby LQR (cont.)

$$f(u_0) = \frac{1}{2}\big((q + a^2 q_K)x_0^2 + 2(b\,a\,q_K x_0)u_0 + (b^2 q_K + r)u_0^2\big)$$

$f(u_0)$ is a parabola and it is smallest at the value u_0 that makes its derivative zero

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}u_0}f(u_0) &= b\,q_K\,a x_0 + (b^2 q_K + r)u_0 \\ &= 0\end{aligned}$$

We have the solution to the optimisation/control problem

$$\begin{aligned}u_0^* &= -\underbrace{\frac{b\,q_K\,a}{b^2 q_K + r}}_{k} x_0 \\ &= -k x_0\end{aligned}$$

□

The linear-quadratic regulator (cont.)

For systems with multiple state variables and multiple inputs, the structure is identical

The system dynamics, in discrete-time

$$x_{k+1} = Ax_k + Bu_k, \quad \text{with } x_k \in \mathcal{R}^{N_x} \text{ and } u_k \in \mathcal{R}^{N_u}$$

The control problem, in discrete-time

$$\underset{u_0, u_1, \dots, u_{K-1}}{\text{minimise}} \quad \underbrace{\frac{1}{2} x_K^T Q_K x_K}_{E(x_K)} + \frac{1}{2} \sum_{k=0}^{K-1} \underbrace{\left(x_k^T Q x_k + u_k^T R u_k \right)}_{L(x_k, u_k)}$$

Consider a finite-horizon of length one ($K = 1$)

$$\underset{u_0}{\text{minimise}} \quad \frac{1}{2} x_1^T Q_K x_1 + \frac{1}{2} \sum_{k=0}^{1-1} \left(x_k^T Q x_k + u_k^T R u_k \right)$$

The linear-quadratic regulator (cont.)

After substituting the dynamics, we get

$$\underset{u_0}{\text{minimise}} \quad \frac{1}{2}\left(\underbrace{x_1}_{Ax_0+Bu_0}{}^T Q_K \underbrace{x_1}_{Ax_0+Bu_0} + x_0^T Q x_0 + u_0^T R u_0\right)$$

After some algebra and rearranging, we have

$$\underset{u_0}{\text{minimise}} \quad \frac{1}{2}\left(x_0^T \left(Q + A^T PA\right) x_0 + 2u_0^T B^T Q_K A x_0 + u_0^T \left(B^T Q_K B + R\right) u_0\right)$$

Taking the derivative and setting it to zero, we get

$$\begin{aligned}\frac{\mathrm{d}f\left(u_0\right)}{\mathrm{d}u_0} &= B^T Q_K A x_0 + \left(B^T Q_K B + R\right) u_0 \\ &= 0\end{aligned}$$

Solving this linear system of equations for the unknown u_0, we get

$$u_0 = -\underbrace{\left(B^T Q_f B + R\right)^{-1} B^T Q_K A}_{K} x_0$$

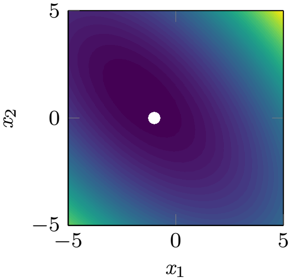
To be able to solve for longer control-horizons, we use backward dynamic programming

Intermezzo

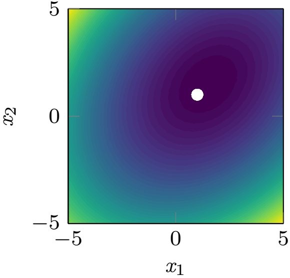
Sum of quadratic functions

The LQR | Sum of quadratic functions

Consider two quadratic functions



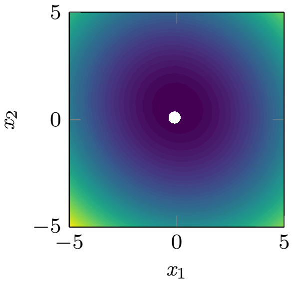
$$V_1(x) = \frac{1}{2}(x-a)^T A\,(x-a) = \frac{1}{2}\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} -1 \\ 0 \end{bmatrix}\right)^T \underbrace{\begin{bmatrix} 1.25 & 0.75 \\ 0.75 & 1.25 \end{bmatrix}}_{\succ 0} \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} -1 \\ 0 \end{bmatrix}\right)$$



$$V_2(x) = \frac{1}{2}(x-b)^T B\,(x-b) = \frac{1}{2}\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix}\right)^T \underbrace{\begin{bmatrix} 1.5 & 0.5 \\ 0.5 & 1.5 \end{bmatrix}}_{\succ 0} \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix}\right)$$

The LQR | Sum of quadratic functions (cont.)

We compute function $V(x) = V_1(x) + V_2(x)$ and show that it is a quadratic function



$$V(x) = \frac{1}{2}\Big((x-v)^T H (x-v) + d\Big)$$

where

$$H = A + B$$
$$v = H^{-1}(Aa - Bb)$$
$$d = -(Aa+Bb)^T H^{-1}(Aa+Bb) + a^T Aa + b^T Bb$$

Matrix H is a positive definite matrix, because both A and B are positive definite

$$\begin{aligned} V(x) &= \frac{1}{2}\Big((x-v)^T H (x-v) + d\Big) \\ &= \frac{1}{2}\left(\left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} -0.1 \\ 0.1 \end{bmatrix}\right)^T \underbrace{\begin{bmatrix} 2.75 & 0.25 \\ 0.25 & 2.75 \end{bmatrix}}_{\succ 0} \left(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - \begin{bmatrix} -0.1 \\ 0.1 \end{bmatrix}\right) + 3.2\right) \end{aligned}$$

The LQR | Sum of quadratic functions (cont.)

Consider two quadratic functions, one of which with a linear combination of variable x

$$V_1(x) = \frac{1}{2}(x - a)^T A (x - a)$$
$$V_2(x) = \frac{1}{2}(Cx - b)^T B (Cx - b)$$

We can compute function $V(x) = V_1(x) + V_2(x)$,

$$V(x) = \frac{1}{2}\Big((x - v)^T H (x - v) + d\Big)$$

where

$$H = A + C^T BC$$
$$v = H^{-1}(Aa - CBb)$$
$$d = -(Aa + CBb)^T H^{-1} (Aa + CBb) + a^T Aa + b^T Bb$$

□

The linear quadratic regulator (cont.)

Dynamic programming

The linear-quadratic regulator (cont.)

We have the optimal control problem, with quadratic cost terms and linear dynamics

$$\begin{aligned} \min_{\substack{x_0, x_1, \dots, x_{K-1}, x_K \\ u_0, u_1, \dots, u_{K-1}}} \quad & E(x_K) + \sum_{k=0}^{K-1} L(x_k, u_k) \\ \text{subject to} \quad & A x_k + B u_k - x_{k+1} = 0, \qquad k = 0, 1, \dots, K-1 \\ & \overline{x}_0 - x_0 = 0 \end{aligned}$$

The optimisation problem can be re-written in the equivalent form

$$\min_{\substack{\overline{x}_0 \\ x_1, \dots, x_{K-1}, x_K \\ u_0, u_1, \dots, u_{K-1}}} \quad \underbrace{L(\overline{x}_0, u_0) + L(x_1, u_1) + \cdots L(x_{K-1}, u_{K-1}) + E(x_K)}_{V(u_0, x_1, u_1, \dots, u_{K-1} | x_0)}$$

We will consider the usual quadratic stage- $L(\cdot, \cdot)$ and terminal- $E(\cdot)$ cost functions

$$\begin{aligned} L(x_k, u_k) &= x_k^T Q_k x_k + u_k^T R_k u_k \\ E(x_k) &= x_K^T Q_K x_K^T \end{aligned}$$

The linear-quadratic regulator (cont.)

After isolating the last two stages, we get

$$\min_{\substack{\overline{x}_0 \\ x_1,\dots,x_{K-2} \\ u_0,u_1,\dots,u_{K-2}}} \Big(L(\overline{x}_0, u_0) + L(x_1, u_1) + \cdots + L(x_{K-2}, u_{K-2}) + \min_{u_{K-1}, x_K} \Big(\underbrace{L(x_{K-1}, u_{K-1}) + E(x_K)}\Big)\Big)$$

At the last stage, we have the problem

$$\begin{aligned} \min_{u_{K-1}, x_K} \quad & L(x_{K-1}, u_{K-1}) + E(x_K) \\ \text{subject to} \quad & Ax_{K-1} + Bu_{K-1} - x_K = 0 \end{aligned}$$

The state x_{K-1} appears as parameter

We define optimal cost (the minimum) and optimal decision variables (the minimiser)

- ⇝ The optimal decision variables $u^*_{K-1}(x_{K-1})$ and $x^*_K(x_{K-1})$
- ⇝ The optimal cost $V^*(x_{K-1})$

The linear-quadratic regulator (cont.)

$$\begin{aligned} \min_{u_{K-1}, x_K} \quad & L(x_{K-1}, u_{K-1}) + E(x_K) \\ \text{subject to} \quad & A x_{K-1} + B u_{K-1} - x_K = 0 \end{aligned}$$

To solve this optimisation problem, we firstly substitute the dynamics then re-arrange

$$\begin{aligned} E(x_K) + L(x_{K-1}, u_{K-1}) &= \underbrace{\frac{1}{2}(A x_{K-1} + B u_{K-1})^T Q_K (A x_{K-1} + B u_{K-1})}_{E(x_K)} \\ &\quad + \underbrace{\frac{1}{2}\left(x_{K-1}^T \ Q \ x_{K-1} + u_{N-1}^T \ R \ u_{N-1}\right)}_{L(x_{K-1}, u_{K-1})} \\ &= \frac{1}{2}\left(x_{K-1}^T \ Q \ x_{K-1} + (u_{K-1} - v)^T H (u_{K-1} - v) + d\right) \end{aligned}$$

where

$$\begin{aligned} H &= R + B^T Q_K B \\ v &= -\underbrace{\left(B^T Q_K B + R\right)^{-1} B^T Q_K A}\, x_{K-1} \\ d &= x_{K-1}^T \left(A^T Q_K A - A^T Q_K B \left(B^T Q_K B + R\right)^{-1} B^T Q_K A\right) x_{K-1} \end{aligned}$$

The linear-quadratic regulator (cont.)

The optimal control action $u^*_{K-1} = v$ is a linear function of the state x_{K-1}

$$u^*_{K-1} = \underbrace{-\left(B^T Q_K B + R\right)^{-1} B^T Q_K A}_{K_{K-1}} x_{K-1}$$

We can compute the terminal state x^*_K from the optimal action

$$\begin{aligned} x^*_K &= A x_{K-1} + B u^*_{K-1} \\ &= A x_{K-1} + B \left(B^T Q_K B + R\right)^{-1} B^T Q_K A x_{K-1} \\ &= \left(A + B \underbrace{\left(B^T Q_K B + R\right)^{-1} B^T Q_K A}_{-K_{K-1}} \right) x_{K-1} \end{aligned}$$

The cost of the optimal control action is quadratic in x_{K-1}

$$V^*_K = \frac{1}{2} \left(x^T_{K-1} Q x_{K-1} + \underbrace{\left(u^*_{K-1} - \underbrace{v}_{u^*_{K-1}} \right)^T H \left(u^*_{K-1} - \underbrace{v}_{u^*_{K-1}} \right)}_{=0} + d \right)$$

The linear-quadratic regulator (cont.)

$$
\begin{aligned}
V_K^* &= \frac{1}{2}\left(x_{K-1}^T Q x_{K-1} + \underbrace{\left(u_{K-1}^* - \underbrace{v}_{u_{K-1}^*} \right)^T H \left(u_{K-1}^* - \underbrace{v}_{u_{K-1}^*} \right)}_{=0} + d \right) \\
&= \frac{1}{2}\left(x_{K-1}^T Q x_{K-1} + \underbrace{x_{K-1}^T \left(A^T Q_K A - A^T Q_K B \left(B^T Q_K B + R \right)^{-1} B^T Q_K A \right) x_{K-1}}_{d} \right) \\
&= \frac{1}{2} x_{K-1}^T \underbrace{\left(Q + A^T Q_K A - A^T Q_K B \left(B^T Q_K B + R \right)^{-1} B^T Q_K A \right)}_{\Pi_{K-1}} x_{K-1}
\end{aligned}
$$

The linear-quadratic regulator (cont.)

$$K_{K-1} = -\left(B^T Q_K B + R\right)^{-1} B^T Q_K A$$

Summarising, we have

$$\begin{aligned} u^*_{K-1}(x_{K-1}) &= K_{K-1} x_{K-1} \\ x^*_K(x_{K-1}) &= (A + BK_{K-1}) x_{K-1} \\ V^*_K(x_{K-1}) &= \frac{1}{2} x^T_{K-1} \Pi_{K-1} x_{K-1} \end{aligned}$$

Function V^*_K defines the optimal cost-to-go from x_{K-1}, under optimal control u^*_{K-1}

⇝ It depends only on x_{K-1}, it allows us to move backwards to stage $K-2$

$$\min_{\substack{\overline{x}_0 \\ x_1,\dots,x_{K-2} \\ u_0,u_1,\dots,u_{K-2}}} L(\overline{x}_0, u_0) + L(x_1, u_1) + \cdots + L(x_{K-2}, u_{K-2}) + V^*(x_{K-1})$$

The linear-quadratic regulator (cont.)

$$\min_{\substack{\overline{x}_0 \\ x_1,\ldots,x_{K-2} \\ u_0,u_1,\ldots,u_{K-2}}} \underbrace{L(\overline{x}_0, u_0) + L(x_1, u_1) + \cdots + L(x_{K-2}, u_{K-2}) + V^*(x_{K-1})}_{V(u_0,x_1,u_1,\ldots,u_{K-2}|x_0)}$$

After isolating the last two stages, we get

$$\min_{\substack{\overline{x}_0 \\ x_1,\ldots,x_{K-3} \\ u_0,u_1,\ldots,u_{K-3}}} \Big(L(\overline{x}_0, u_0) + L(x_1, u_1) + \cdots + L(x_{K-3}, u_{K-3}) +$$
$$\min_{u_{K-2},x_{K-1}} \underbrace{\Big(L(x_{K-2}, u_{K-2}) + \underbrace{V^*(x_{K-1})}\Big)}\Big)$$

At the last stage, we have the problem

$$\begin{aligned} \min_{u_{K-2},x_{K-1}} \quad & V^*(x_{K-1}) + L(x_{K-2}, u_{K-2}) \\ \text{subject to} \quad & Ax_{K-2} + Bu_{K-2} - x_{K-1} = 0 \end{aligned}$$

The state x_{K-2} appears as parameter

The linear-quadratic regulator (cont.)

$$\begin{aligned} \min_{u_{K-2}, x_{K-2}} \quad & V^*(x_{K-1}) + L(x_{K-2}, u_{K-2}) \\ \text{subject to} \quad & A x_{K-2} + B u_{K-2} - x_{K-1} = 0 \end{aligned}$$

We define optimal cost (the minimum) and optimal decision variables (the minimiser)

- The optimal decision variables $u^*_{K-2}(x_{K-2})$ and $x^*_{K-2}(x_{K-2})$

$$\begin{aligned} u^*_{K-2}(x_{K-2}) &= K_{K-2} x_{K-2} \\ x^*_{K-1}(x_{K-2}) &= (A + B K_{K-2}) x_{K-2} \end{aligned}$$

- The optimal cost $V^*(x_{K-2})$ from stage $K-2$ to K

$$V^*_{K-1}(x_{K-2}) = \frac{1}{2} x^T_{K-2} \, \Pi_{K-2} \, x_{K-2}$$

We used,

$$\begin{aligned} K_{K-2} &= -\left(B^T \Pi_{K-1} B + R\right)^{-1} B^T \Pi_{K-1} A \\ \Pi_{K-2} &= Q + A^T \, \Pi_{K-1} \, A - A^T \, \Pi_{K-1} \, B \left(B^T \Pi_{K-1} B + R\right)^{-1} B^T \Pi_{K-1} A \end{aligned}$$

The linear-quadratic regulator (cont.)

The recursion that gives Π_{K-2} from Π_{K-1} is known as the **backward Riccati iteration**

In the general form, the recursion starts from $\Pi_K = Q_K$

$$\Pi_{k-1} = Q + A^T \Pi_k A - A^T \Pi_k B \left(B^T \Pi_k B + R \right)^{-1} B^T \Pi_k A \qquad (k = K, K-1, \cdots, 1)$$

We can also define the general form of the optimal cost and optimal decision variables

- For the optimal decision variables $u_k^*(x_k)$ and $x_k^*(x_k)$, we have

$$\begin{aligned} u_k^*(x_k) &= -K_k x_k \\ x_k^*(x_k) &= (A + BK_k)\, x_k \end{aligned}$$

- For the optimal cost-to-go $V^*(x_k)$ from stage k to K, we have

$$V_k^*(x_k) = \frac{1}{2} x_k^T \, \Pi_{k+1} \, x_k$$

An example

The linear quadratic regulator

The linear-quadratic regulator (cont.)

Example

Consider the linear and time-invariant dynamical system with measurement process

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t)$$

Consider the following system matrices and associate IO representation

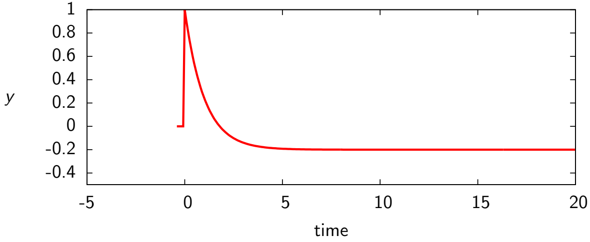
$$A = -b \qquad y(s) = g(s)u(s)$$
$$B = -(a+b)$$
$$C = k \qquad g(s) = k\frac{s-a}{s+b}$$
$$D = k$$

For $(a, b) = (0.2, 1) > 0$ and $k = 1$, system has inverse response (right-half-plane zero)

The linear-quadratic regulator (cont.)

Step response, by solving the ODE with $u(t) = 1$ and initial condition $x(0) = 0$

- ⇝ We observe what happens from the measurements $y(t)$
- ⇝ The response to a unit step of the control $u(t)$



Suppose that we request a unit step of the output $y(t)$, say a set-point change

- We ask what is the optimal control action
- The best action capable to deliver it

The linear-quadratic regulator (cont.)

$$y(s) = \underbrace{k\frac{s-a}{s+b}}_{g(s)} u(s)$$

In the Laplace domain, we have the requested output

$$\overline{y}(s) = \frac{1}{s}$$

WE substitute it and solve for $\overline{u}(s)$, we get

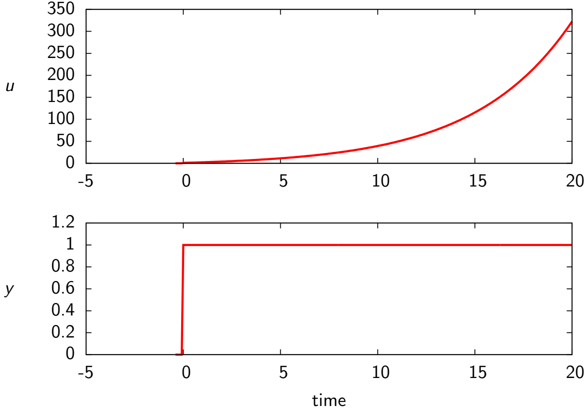
$$\begin{aligned}\overline{u}(s) &= \frac{\overline{y}}{g(s)} \\ &= \frac{s+b}{ks(s-a)}\end{aligned}$$

Back to the time-domain, the control

$$u(t) = \frac{1}{ka}\Bigg(-b + (a+b)\underbrace{e^{at}}_{a>0\ (!)}\Bigg)$$

The linear-quadratic regulator (cont.)

Output response $y(t)$ is perfectly on target, with an exponentially growing input $u(t)$



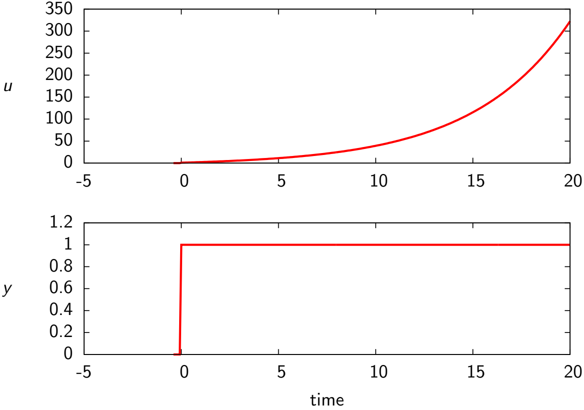
We are capable of achieving perfect tracking in $y(t)$ by using applying an optimal $u(t)$

The linear-quadratic regulator (cont.)

$$g(s) = k\frac{s-a}{s+b}, \text{ with } \overline{u}(s) = \frac{1}{s-a}\frac{s+b}{ks}$$

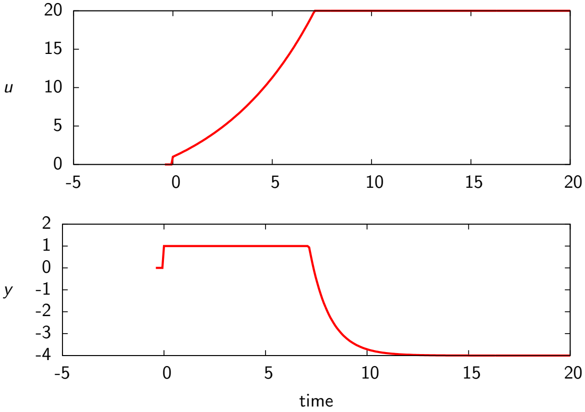
The zeros at $s = a$ in $g(s)$ and $\overline{u}(s)$ cancel out, tracking of output $y(t)$ looks perfect

- The input-blocking property of the zero in the transfer function



The linear-quadratic regulator (cont.)

Clearly, inputs $u(t)$ cannot grow unboundedly, at some point they will hit constraints



The saturation of the input at the constraint destroys the perfect output response $y(t)$

Linear-quadratic optimal control | LTV-QR

We can also consider the more general formulation of a linear-quadratic optimal control

$$\begin{aligned}\min_{x,u} \quad & \underbrace{x_K^T Q_K x_K}_{E(x_K)} + \sum_{k=0}^{K-1} \underbrace{\begin{bmatrix} x_k \\ u_k \end{bmatrix}^T \begin{bmatrix} Q_k & S_k^T \\ S_k & R_k \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix}}_{L_k(x_k, u_k)} \\ \text{subject to} \quad & x_{k+1} - A_k x_k - B_k u_k = 0, \quad k = 0, 1, \dots, K-1 \\ & x_0 - \overline{x}_0 = 0\end{aligned}$$

At each step k of the recursion, we must compute the (varying) stage-cost $L_k(x_k, u_k)$

$$L_k(x_k, u_k) = \begin{bmatrix} x_k \\ u_k \end{bmatrix}^T \begin{bmatrix} Q_k & S_k^T \\ S_k & R_k \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix}$$

Matrices Q_k and R_k are time-varying and positive semi-definite and positive definite

- Also matrix Q_K is positive definite

Moreover, we may add further flexibility in tuning by including the mixing matrix S_k

Linear-quadratic optimal control | LTV-QR (cont.)

$$\min_{x,u} \quad \underbrace{x_K^T Q_K x_K}_{E(x_K)} + \sum_{k=0}^{K-1} \underbrace{\begin{bmatrix} x_k \\ u_k \end{bmatrix}^T \begin{bmatrix} Q_k & S_k^T \\ S_k & R_k \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix}}_{L_k(x_k,u_k)}$$

$$\text{subject to} \quad x_{k+1} - A_k x_k - B_k u_k = 0, \qquad k = 0, 1, \dots, K-1$$

$$x_0 - \overline{x}_0 = 0$$

Furthermore, we allow the system dynamics to be time-varying,

$$f_k(x_k, u_k) = A_k x_k + B_k u_k$$

The optimal cost $V_k^*(x_k)$ from stage k to $k+1$ is still quadratic

$$V_k^*(x_k) = \frac{1}{2} x_k^T \Pi_{k+1} x_k$$

The backward Riccati recursion is used to compute Π_{k+1}

Linear-quadratic optimal control | LTV-QR (cont.)

Using the terminal condition $\Pi_K = Q_K$, we have

$$\Pi_k = Q_k + A_k^T \Pi_{k+1} A_k - \Big(S_k^T + A_k^T \Pi_{k+1} B_k\Big)\Big(R_k + B_k^T \Pi_{k+1} B_k\Big)^{-1}\Big(S_k + B_k^T \Pi_{k+1} A_k\Big)$$

The optimal decision variables are obtained from the feedback law,

$$u_k^*(x_k) = \underbrace{-\Big(R_k + B_k^T \Pi_{k+1} B_k\Big)^{-1}\Big(S_k + B_k^T \Pi_{k+1} A_k\Big)}_{K_k} x_k$$

The forward simulation from $\overline{x}_0$ determines the state variables

$$x_{k+1} = A_k x_k + B_k u_k^*$$

■

Linear-quadratic optimal control | AQR

We consider even more general formulations, to get an affine-quadratic optimal control

$$\min_{x,u} \quad \underbrace{\begin{bmatrix} 1 \\ x_K \end{bmatrix}^T \begin{bmatrix} * & q_K^T \\ q_K & Q_K \end{bmatrix} \begin{bmatrix} 1 \\ x_K \end{bmatrix}}_{E(x_K)} + \sum_{k=0}^{K-1} \underbrace{\begin{bmatrix} 1 \\ x_k \\ u_k \end{bmatrix}^T \begin{bmatrix} * & q_k^T & s_k^T \\ q_k & Q_k & S_k^T \\ s_k & S_k & R_k \end{bmatrix} \begin{bmatrix} 1 \\ x_k \\ u_k \end{bmatrix}}_{L_k(x_k, u_k)}$$

$$\text{subject to} \quad x_{k+1} - A_k x_k - B_k u_k - c_k = 0, \quad k = 0, 1, \dots, K-1$$
$$x_0 - \overline{x}_0 = 0$$

These optimisations often result from trajectory linearisations of nonlinear dynamics

The general dynamic programming solution is retained by augmenting the state

$$\widetilde{x}_k = \begin{bmatrix} 1 \\ x_k \end{bmatrix}$$

The augmented dynamics take the form

$$\widetilde{x}_{k+1} = \begin{bmatrix} 1 & 0 \\ c_k & A_k \end{bmatrix} \widetilde{x}_k + \begin{bmatrix} 0 \\ B_k \end{bmatrix} u_k$$

The fixed initial-value is $\overline{\widetilde{x}}_0 = \begin{bmatrix} 1 & \overline{x}_0 \end{bmatrix}^T$

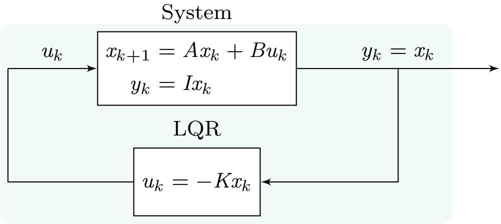
The linear-quadratic regulator | Infinite-horizon

We discussed the linear-quadratic regulator over a finite horizon of some duration K

Linear-quadratic regulators can de-stabilise a stable system over finite horizons

- Setting $Q, R \succ 0$ is not sufficient to guarantee closed-loop stability

$$\begin{cases} x_{k+1} = Ax_k + B\Big(\underbrace{-Kx_k}_{u_k}\Big) \\ y(t) = x(t) \end{cases}$$



The stability of the closed-loop is determined by the eigenvalues of matrix A_{CL}

The closed-loop dynamics,

$$\begin{aligned} x_{k+1} &= Ax_k - BKx_k \\ &= \underbrace{(A - BK)}_{A_{CL}} x_k \end{aligned}$$

An example

The linear quadratic regulator

The linear-quadratic regulator | Infinite-horizon (cont.)

Example

Consider a discrete-time linear and time-invariant dynamical system with LQR ($K = 5$)

$$x_{k+1} = \underbrace{\begin{bmatrix} 4/3 & -2/3 \\ 1 & 0 \end{bmatrix}}_{A} x_k + \underbrace{\begin{bmatrix} 1 \\ 0 \end{bmatrix}}_{B} u_k$$

$$y_k = \underbrace{\begin{bmatrix} -2/3 \\ 1 \end{bmatrix}}_{C} x_K$$

The discrete-time transfer function has a zero ($z = 3/2$), non-minimum phase system

$$\begin{aligned} \min_{\substack{x_0, x_1, \dots, x_4, x_5 \\ u_0, u_1, \dots, u_4}} \quad & x_5^T \; Q_5 \; x_5 + \sum_{k=0}^{4} x_k^T \; Q \; x_k + u_k^T R u_k \\ \text{subject to} \quad & A x_k + B u_k - x_{k+1} = 0, \quad k = 0, 1, \dots, 4 \\ & \overline{x}_0 - x_0 = 0 \end{aligned}$$

We use $Q = Q_5 = C^T C + 0.001 I$ and $R = 0.001$ that barely penalises control actions

The linear-quadratic regulator | Infinite-horizon (cont.)

Based on the Riccati equation, we iterate four times from $\Pi_K = Q_K = Q$

$$K_4^{(5)}, K_3^{(5)}, K_2^{(5)}, K_1^{(5)}, K_0^{(5)}$$

Assuming that we use the first feedback gain $K_0^{(5)}$, we have

$$u_k = K_0^{(5)} x_k$$
$$x_k = \left(A + BK_0^{(5)}\right)^k x_0$$

In closed-loop, the eigenvalues of $\left(A + BK_0^{(5)}\right) = A_{\text{CL}}^{(5)}$

$$\lambda\left(A_{\text{CL}}^{(5)}\right) = (\underbrace{1.307}_{>1}, 0.001)$$

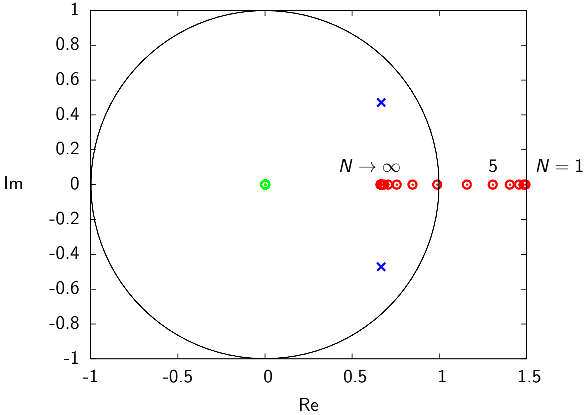
One of the eigenvalues is outside the unit circle

- The closed-loop system is unstable
- The state grows exponentially
- $x_k \to \infty$ as $k \to \infty$

The linear-quadratic regulator | Infinite-horizon (cont.)

The closed-loop eigenvalues of $\left(A + BK_0^{(K)}\right)$ for horizons L of different duration ($\circ$)

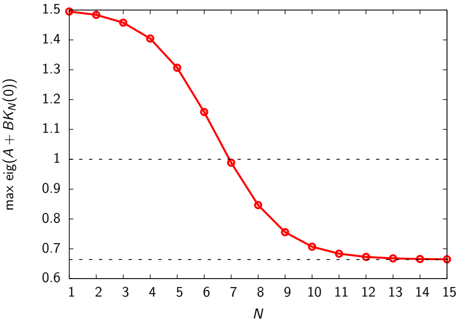
- For reference, the open-loop eigenvalues of A ($\times$) are both stable



When we start with a finite horizon LQR, we move both the open-loop eigenvalues

- ⇝ From $K = 1$, until we enter the unit disc at $K = 7$
- ⇝ The stability margin grows with K

The linear-quadratic regulator | Infinite-horizon (cont.)



Stability margin as function of the control horizon

- Finite-horizon may return unstable controllers
- More robustness is gained as the horizon grows

$$\lambda\left(A_{\text{CL}}^{(\infty)}\right) = (\underbrace{0.664}_{<1}, 0.001)$$

A feedback gain $K_0^{(\infty)}$ corresponds to an infinite-horizon linear-quadratic regulator

$$\begin{aligned}
\min_{\substack{x_0, x_1, \dots, \\ u_0, u_1, \dots}} \quad & \sum_{k=0}^{\infty} x_k^T Q x_k + u_k^T R u_k \\
\text{subject to} \quad & A x_k + B u_k - x_{k+1} = 0, \quad k = 0, 1, \dots \\
& \overline{x}_0 - x_0 = 0
\end{aligned}$$

The linear-quadratic regulator | Infinite-horizon (cont.)

$$\begin{aligned}
\min_{\substack{x_0, x_1, \dots, \\ u_0, u_1, \dots}} \quad & \sum_{k=0}^{\infty} x_k^T Q x_k + u_k^T R u_k \\
\text{subject to} \quad & A x_k + B u_k - x_{k+1} = 0, \quad k = 0, 1, \dots \\
& \overline{x}_0 - x_0 = 0
\end{aligned}$$

If we are interested in controlling a continuous process, without a final-time, then the natural formulation of the optimal control problem is with an infinite-horizon cost

- In this case, the Riccati recursion has a stationary solution $\Pi_k = \Pi_{k+1}$,

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

Given Π, we have the classic optimal control feedback

$$u^* = -\underbrace{\left(R + B^T \Pi B \right)^{-1} B^T \Pi A}_{K} \, x_k$$

Closed-loop stability is not relevant for batch processes, finite-horizon LQRs are fine

The linear-quadratic regulator | Infinite-horizon (cont.)

$$\begin{aligned}
\min_{\substack{x_0, x_1, \ldots, \\ u_0, u_1, \ldots}} \quad & \sum_{k=0}^{\infty} x_k^T Q x_k + u_k^T R u_k \\
\text{subject to} \quad & A x_k + B u_k - x_{k+1} = 0, \quad k = 0, 1, \ldots \\
& \overline{x}_0 - x_0 = 0
\end{aligned}$$

Infinite-horizon solutions exist as long as the cost function is bounded

- In this case, the cost function is an infinite sum
- But, ... the result must not be infinitely big

This is possible when the linear-time invariant system is controllable

- ⇝ We can transfer its state from anywhere to anywhere
- ⇝ And, there exists a control sequence to do that
- ⇝ And, it can be done in finite time

The linear-quadratic regulator | Infinite-horizon (cont.)

If the pair (A, B) is controllable, the there exists a finite horizon of length K and a sequence of inputs that can transfer the state of the system from any x to any x'

That is, by forward simulation

$$x^+ = A^K x + \begin{bmatrix} B & AB & \cdots & A^{K-1}B \end{bmatrix} \begin{bmatrix} u_{K_1} \\ u_{K-1} \\ \vdots \\ u_0 \end{bmatrix}$$

Similarly, rearranging we get

$$\underbrace{\begin{bmatrix} B & AB & \cdots & A^{K-1}B \end{bmatrix}}_{\mathcal{C}} \begin{bmatrix} u_{K_1} \\ u_{K-1} \\ \vdots \\ u_0 \end{bmatrix} = x^+ - A^K x$$

Controllability matrix $\mathcal{C}$ must be full rank for the equation to have a solution $\{u_k\}_{k=0}^{K-1}$

- If cannot reach x' in K moves, then we cannot reach it in any number of moves