Linear Quadratic Regulator

- Problem Formulation and Overview
- Discrete-Time Linear-Quadratic Optimal Control
- Dynamic Programming
- Riccati Differential Equations

Contents

Continuous-Time Linear-Quadratic Optimal Control

Goal:

Solve the continuous-time linear-quadratic optimal control problem

$$\min_{x,u} \quad \int_0^T \left\{ x(\tau)^\intercal Q x(\tau) + u(\tau)^\intercal R u(\tau) \right\} \mathrm{d}\tau + x(T) \mathcal{P}_N x(T)$$

$$\text{s.t.} \quad \begin{cases} \dot{x}(t) & = & Ax(t) + Bu(t)\,, \quad t \in [0, T] \\ x(0) & = & x_0 \end{cases}$$

Assumption: The weighting matrices Q and R are positive definite.

Direct Methods

Overview: In order to solve the continuous-time LQR problem, we use a so-called "direct approach". This means that we proceed in three steps:

- First, we discretize the problem (in this lecture: Euler's method)
- Second, we solve the discrete-time optimal control problem
- And third, we take the limit to solve the original problem.

Contents

Euler Discretization

Let us use am equidistant piecewise-constant control discretization,

$$u(t) \approx \begin{cases} v_0 & \text{if } t \in [t_0, t_1] \\ v_1 & \text{if } t \in [t_1, t_2] \\ \vdots & \\ v_{N-1} & \text{if } t \in [t_{N-1}, t_N] \end{cases} \qquad \text{with} \qquad t_k = kh$$

and $h = \frac{T}{N}$ in combinaion with Euler's discretization method

$$y_{k+1} = y_k + h\left(Ay_k + Bv_k\right) \quad \text{with} \quad y_0 = x_0 \; .$$

This discretization can be made arbitrarily accurate by chooising sufficiently small h,

$$y_k = x(t_k) + \mathbf{O}(h) \; .$$

Euler Discretization

Let us use am equidistant piecewise-constant control discretization,

$$u(t) \approx \begin{cases} v_0 & \text{if } t \in [t_0, t_1] \\ v_1 & \text{if } t \in [t_1, t_2] \\ \vdots & \\ v_{N-1} & \text{if } t \in [t_{N-1}, t_N] \end{cases} \qquad \text{with} \qquad t_k = kh$$

and $h = \frac{T}{N}$ in combinaion with Euler's discretization method

$$y_{k+1} = y_k + h\left(Ay_k + Bv_k\right) \quad \text{with} \quad y_0 = x_0 \ .$$

This discretization can be made arbitrarily accurate by chooising sufficiently small h,

$$y_k = x(t_k) + \mathbf{O}(h) \ .$$

Euler Discretization

The result of the discretization is a linear discrete-time system

$$y_{k+1} = \mathcal{A} y_k + \mathcal{B} v_k \quad \text{with} \quad \mathcal{A} = I + hA \quad \text{and} \quad \mathcal{B} = hB \; .$$

The objective can be approximated, too,

$$\int_0^T \left\{ x(\tau)^\intercal Q x(\tau) + u(\tau)^\intercal R u(\tau) \right\} \mathrm{d}\tau = \sum_{k=0}^{N-1} \left\{ y_k^\intercal \mathcal{Q} y_k + v_k^\intercal \mathcal{R} v_k \right\} + \mathbf{O}(h)$$

with matrices

$$\mathcal{Q} = hQ \quad \text{and} \quad \mathcal{R} = hR \; .$$

Euler Discretization

The result of the discretization is a linear discrete-time system

$$y_{k+1} = \mathcal{A} y_k + \mathcal{B} v_k \quad \text{with} \quad \mathcal{A} = I + hA \quad \text{and} \quad \mathcal{B} = hB \ .$$

The objective can be approximated, too,

$$\int_0^T \{x(\tau)^\intercal Q x(\tau) + u(\tau)^\intercal R u(\tau)\} \, \mathrm{d}\tau = \sum_{k=0}^{N-1} \{y_k^\intercal \mathcal{Q} y_k + v_k^\intercal \mathcal{R} v_k\} + \mathbf{O}(h)$$

with matrices

$$\mathcal{Q} = hQ \quad \text{and} \quad \mathcal{R} = hR \ .$$

Discrete-Time Linear-Quadratic Optimal Control

By substituting the above discretizations of the system and the quadratic objective, we obtain a finite dimensional optimization problem

$$
\begin{array}{ll}
\underset{y,v}{\text{minimize}} & \displaystyle\sum_{k=0}^{N-1} \{y_k^\mathsf{T} \mathcal{Q} y_k + v_k^\mathsf{T} \mathcal{R} v_k\} + y_N \mathcal{P}_N y_N \\
\text{subject to} & \left\{ \begin{array}{rcl} y_{k+1} & = & \mathcal{A} y_k + \mathcal{B} v_k \,, \quad k \in 0, \ldots, N-1 \\ y_0 & = & x_0 \end{array} \right.
\end{array}
$$

Contents

Cost-To-Go Function

We call the function $J_i : \mathbb{R}^{n_x} \to \mathbb{R}_+$,

$$J_i(z) = \quad \underset{x,u}{\text{minimize}} \quad \sum_{k=i}^{N-1} \{y_k^\intercal \mathcal{Q} y_k + u_k^\intercal \mathcal{R} u_k\} + y_N^\intercal P_N y_N$$

$$\text{subject to} \quad \left\{ \begin{array}{rcl} y_{k+1} & = & \mathcal{A} y_k + \mathcal{B} u_k, \quad k \in \{i, \ldots, N-1\} \\ y_i & = & z \ , \end{array} \right.$$

the i-th cost-to-go function. It is defined for all $z \in \mathbb{R}^{n_x}$.

Bellman's Principle of Optimality

The cost-to-go function satisfies the dynamic programming recursion

$$\begin{aligned} J_i(y_i) = \quad & \underset{y_{i+1}, u_i}{\text{minimize}} \quad y_i^\mathsf{T} \mathcal{Q} y_i + u_i^\mathsf{T} \mathcal{R} u_i + J_{i+1}(y_{i+1}) \\ & \text{subject to} \quad y_{i+1} = \mathcal{A} y_i + \mathcal{B} u_i \ , \end{aligned}$$

for all $i \in \{0, \ldots, N-1\}$ with

$$J_N(y_N) = y_N^\mathsf{T} \mathcal{P}_N y_N$$

(also known as "Bellman's principle of optimality")

Riccati Recursions

Theorem: The cost-to-go function is quadratic, $J_i(x) = x^\intercal P_i x$.

Proof: The proof uses induction over i.

- Induction start: $J_N(z) = z^\intercal \mathcal{P}_N z$.
- Induction step: if $J_{i+1}(z) = z^\intercal \mathcal{P}_{i+1} z$, then

$$
\begin{aligned}
J_i(z) \quad &= \quad \min_{v_i} \; z^\intercal \mathcal{Q} z + v_i^\intercal \mathcal{R} v_i + (\mathcal{A}z + \mathcal{B}v_i)^\intercal \mathcal{P}_{i+1} (\mathcal{A}z + \mathcal{B}v_i) \\
&\qquad \Longrightarrow \quad v_i^\star = -\left(\mathcal{R} + \mathcal{B}^\intercal \mathcal{P}_{i+1} \mathcal{B}\right)^{-1} \left[\mathcal{A}^\intercal P_{i+1} \mathcal{B}\right]^\intercal z \\
&\qquad \Longrightarrow \quad J_i(z) = z^\intercal \mathcal{P}_i z
\end{aligned}
$$

with

$$
\mathcal{P}_i = \mathcal{A}^\intercal \mathcal{P}_{i+1} \mathcal{A} + \mathcal{Q} - \left[\mathcal{A}^\intercal \mathcal{P}_{i+1} \mathcal{B}\right] \left(\mathcal{R} + \mathcal{B}^\intercal \mathcal{P}_{i+1} \mathcal{B}\right)^{-1} \left[\mathcal{A}^\intercal \mathcal{P}_{i+1} \mathcal{B}\right]^\intercal .
$$

Riccati Recursions

Theorem: The cost-to-go function is quadratic, $J_i(x) = x^\intercal P_i x$.

Proof: The proof uses induction over i.

- Induction start: $J_N(z) = z^\intercal \mathcal{P}_N z$.
- Induction step: if $J_{i+1}(z) = z^\intercal \mathcal{P}_{i+1} z$, then

$$\begin{aligned} J_i(z) \quad &= \quad \min_{v_i} \; z^\intercal \mathcal{Q} z + v_i^\intercal \mathcal{R} v_i + (\mathcal{A}z + \mathcal{B}v_i)^\intercal \mathcal{P}_{i+1} (\mathcal{A}z + \mathcal{B}v_i) \\ &\Longrightarrow \quad v_i^\star = -\left(\mathcal{R} + \mathcal{B}^\intercal \mathcal{P}_{i+1} \mathcal{B}\right)^{-1} \left[\mathcal{A}^\intercal P_{i+1} \mathcal{B}\right]^\intercal z \\ &\Longrightarrow \quad J_i(z) = z^\intercal \mathcal{P}_i z \end{aligned}$$

with

$$\mathcal{P}_i = \mathcal{A}^\intercal \mathcal{P}_{i+1} \mathcal{A} + \mathcal{Q} - \left[\mathcal{A}^\intercal \mathcal{P}_{i+1} \mathcal{B}\right] \left(\mathcal{R} + \mathcal{B}^\intercal \mathcal{P}_{i+1} \mathcal{B}\right)^{-1} \left[\mathcal{A}^\intercal \mathcal{P}_{i+1} \mathcal{B}\right]^\intercal \; .$$

Riccati Recursions

Theorem: The cost-to-go function is quadratic, $J_i(x) = x^\intercal P_i x$.

Proof: The proof uses induction over i.

- Induction start: $J_N(z) = z^\intercal \mathcal{P}_N z$.
- Induction step: if $J_{i+1}(z) = z^\intercal \mathcal{P}_{i+1} z$, then

$$
\begin{aligned}
J_i(z) \quad &= \quad \min_{v_i} \; z^\intercal \mathcal{Q} z + v_i^\intercal \mathcal{R} v_i + (\mathcal{A}z + \mathcal{B}v_i)^\intercal \, \mathcal{P}_{i+1} \, (\mathcal{A}z + \mathcal{B}v_i) \\
&\Longrightarrow \quad v_i^\star = -\left(\mathcal{R} + \mathcal{B}^\intercal \mathcal{P}_{i+1} \mathcal{B}\right)^{-1} \left[\mathcal{A}^\intercal P_{i+1} \mathcal{B}\right]^\intercal z \\
&\Longrightarrow \quad J_i(z) = z^\intercal \mathcal{P}_i z
\end{aligned}
$$

with

$$
\mathcal{P}_i = \mathcal{A}^\intercal \mathcal{P}_{i+1} \mathcal{A} + \mathcal{Q} - \left[\mathcal{A}^\intercal \mathcal{P}_{i+1} \mathcal{B}\right] \left(\mathcal{R} + \mathcal{B}^\intercal \mathcal{P}_{i+1} \mathcal{B}\right)^{-1} \left[\mathcal{A}^\intercal \mathcal{P}_{i+1} \mathcal{B}\right]^\intercal \; .
$$

Riccati Recursions

Theorem: The cost-to-go function is quadratic, $J_i(x) = x^\mathsf{T} P_i x$.

Proof: The proof uses induction over i.

- Induction start: $J_N(z) = z^\mathsf{T} \mathcal{P}_N z$.
- Induction step: if $J_{i+1}(z) = z^\mathsf{T} \mathcal{P}_{i+1} z$, then

$$
\begin{aligned}
J_i(z) \quad &= \quad \min_{v_i} \; z^\mathsf{T} \mathcal{Q} z + v_i^\mathsf{T} \mathcal{R} v_i + (\mathcal{A}z + \mathcal{B}v_i)^\mathsf{T} \, \mathcal{P}_{i+1} \, (\mathcal{A}z + \mathcal{B}v_i) \\
&\Longrightarrow \quad v_i^\star = -\left(\mathcal{R} + \mathcal{B}^\mathsf{T} \mathcal{P}_{i+1} \mathcal{B}\right)^{-1} \left[\mathcal{A}^\mathsf{T} P_{i+1} \mathcal{B}\right]^\mathsf{T} z \\
&\Longrightarrow \quad J_i(z) = z^\mathsf{T} \mathcal{P}_i z
\end{aligned}
$$

with

$$
\mathcal{P}_i = \mathcal{A}^\mathsf{T} \mathcal{P}_{i+1} \mathcal{A} + \mathcal{Q} - \left[\mathcal{A}^\mathsf{T} \mathcal{P}_{i+1} \mathcal{B}\right] \left(\mathcal{R} + \mathcal{B}^\mathsf{T} \mathcal{P}_{i+1} \mathcal{B}\right)^{-1} \left[\mathcal{A}^\mathsf{T} \mathcal{P}_{i+1} \mathcal{B}\right]^\mathsf{T} \; .
$$

Riccati Recursions

- The backward recursion

$$\mathcal{P}_i = \mathcal{A}^\mathsf{T}\mathcal{P}_{i+1}\mathcal{A} + \mathcal{Q} - [\mathcal{A}^\mathsf{T}\mathcal{P}_{i+1}\mathcal{B}]\left(\mathcal{R} + \mathcal{B}^\mathsf{T}\mathcal{P}_{i+1}\mathcal{B}\right)^{-1}[\mathcal{A}^\mathsf{T}\mathcal{P}_{i+1}\mathcal{B}]^\mathsf{T}$$

  is called an algebraic (discrete-time) Riccati recursion.

- The optimal solution of the linear-quadratic optimal control problem can be found by forward simulation,

$$\begin{aligned} v_i &= K_i y_i \quad \text{with} \quad K_i = -\left(\mathcal{R} + \mathcal{B}^\mathsf{T}\mathcal{P}_{i+1}\mathcal{B}\right)^{-1}[\mathcal{A}^\mathsf{T}\mathcal{P}_{i+1}\mathcal{B}]^\mathsf{T} \ , \\ y_{i+1} &= (\mathcal{A} + \mathcal{B}K_i)\, y_i \quad \text{with} \quad y_0 = x_0 \ . \end{aligned}$$

- The matrices K_i are called the optimal feedback gains.

Riccati Recursions

- The backward recursion

$$\mathcal{P}_i = \mathcal{A}^\intercal \mathcal{P}_{i+1}\mathcal{A} + \mathcal{Q} - [\mathcal{A}^\intercal \mathcal{P}_{i+1}\mathcal{B}]\left(\mathcal{R} + \mathcal{B}^\intercal \mathcal{P}_{i+1}\mathcal{B}\right)^{-1}[\mathcal{A}^\intercal \mathcal{P}_{i+1}\mathcal{B}]^\intercal$$

  is called an algebraic (discrete-time) Riccati recursion.

- The optimal solution of the linear-quadratic optimal control problem can be found by forward simulation,

$$\begin{aligned} v_i &= K_i y_i \quad \text{with} \quad K_i = -\left(\mathcal{R} + \mathcal{B}^\intercal \mathcal{P}_{i+1}\mathcal{B}\right)^{-1}[\mathcal{A}^\intercal \mathcal{P}_{i+1}\mathcal{B}]^\intercal \ , \\ y_{i+1} &= \left(\mathcal{A} + \mathcal{B}K_i\right) y_i \quad \text{with} \quad y_0 = x_0 \ . \end{aligned}$$

- The matrices K_i are called the optimal feedback gains.

Riccati Recursions

- The backward recursion

$$\mathcal{P}_i = \mathcal{A}^\mathsf{T}\mathcal{P}_{i+1}\mathcal{A} + \mathcal{Q} - [\mathcal{A}^\mathsf{T}\mathcal{P}_{i+1}\mathcal{B}]\left(\mathcal{R} + \mathcal{B}^\mathsf{T}\mathcal{P}_{i+1}\mathcal{B}\right)^{-1}[\mathcal{A}^\mathsf{T}\mathcal{P}_{i+1}\mathcal{B}]^\mathsf{T}$$

  is called an algebraic (discrete-time) Riccati recursion.

- The optimal solution of the linear-quadratic optimal control problem can be found by forward simulation,

$$\begin{aligned} v_i &= K_i y_i \quad \text{with} \quad K_i = -\left(\mathcal{R} + \mathcal{B}^\mathsf{T}\mathcal{P}_{i+1}\mathcal{B}\right)^{-1}[\mathcal{A}^\mathsf{T}\mathcal{P}_{i+1}\mathcal{B}]^\mathsf{T} \ , \\ y_{i+1} &= \left(\mathcal{A} + \mathcal{B}K_i\right) y_i \quad \text{with} \quad y_0 = x_0 \ . \end{aligned}$$

- The matrices K_i are called the optimal feedback gains.

Contents

Back to Continuous-Time...

Start with the disrete time Riccati recursion and substitute

$$\mathcal{A} = I + hA, \quad \mathcal{B} = hB, \quad \mathcal{Q} = hQ, \quad \text{and} \quad \mathcal{R} = hR\,.$$

This gives

$$\mathcal{P}_i \;=\; \mathcal{P}_{i+1} + h\left[A^\mathsf{T}\mathcal{P}_{i+1} + \mathcal{P}_{i+1}A + Q - \mathcal{P}_{i+1}BR^{-1}B^\mathsf{T}\mathcal{P}_{i+1}\right] + \mathbf{O}(h^2)$$

Set $P(t_i) = \mathcal{P}_i = \mathcal{P}_{i+1} + \mathbf{O}(h)$ and take the limit for $h \to 0$:

$$-\dot{P}(t) \;=\; A^\mathsf{T}P(t) + P(t)A + Q - P(t)BR^{-1}B^\mathsf{T}P(t)$$

$$\text{with} \quad P(T) \;=\; \mathcal{P}_N\,.$$

This differential equation is called a Riccati differential equation.

Back to Continuous-Time...

Start with the disrete time Riccati recursion and substitute

$$\mathcal{A} = I + hA, \quad \mathcal{B} = hB, \quad \mathcal{Q} = hQ, \quad \text{and} \quad \mathcal{R} = hR\,.$$

This gives

$$\mathcal{P}_i \quad = \quad \mathcal{P}_{i+1} + h\left[A^\mathsf{T}\mathcal{P}_{i+1} + \mathcal{P}_{i+1}A + Q - \mathcal{P}_{i+1}BR^{-1}B^\mathsf{T}\mathcal{P}_{i+1}\right] + \mathbf{O}(h^2)$$

Set $P(t_i) = \mathcal{P}_i = \mathcal{P}_{i+1} + \mathbf{O}(h)$ and take the limit for $h \to 0$:

$$\begin{aligned} -\dot{P}(t) &= A^\mathsf{T}P(t) + P(t)A + Q - P(t)BR^{-1}B^\mathsf{T}P(t) \\ \text{with} \quad P(T) &= \mathcal{P}_N\,. \end{aligned}$$

This differential equation is called a Riccati differential equation.

Summary: Continuous-Time LQR

The optimal control problem

$$\min_{x,u} \quad \int_0^T \{x(\tau)^\mathsf{T} Q x(\tau) + u(\tau)^\mathsf{T} R u(\tau)\} \, \mathrm{d}\tau + x(T)\mathcal{P}_N x(T)$$

$$\text{s.t.} \quad \begin{cases} \dot{x}(t) & = & Ax(t) + Bu(t) \,, \quad t \in [0,T] \\ x(0) & = & x_0 \end{cases}$$

can be solved explicitly by passing trough 3 steps:

Summary: Continuous-Time LQR

Step 1: Solve the Riccati differential equation

$$\begin{aligned} -\dot{P}(t) &= A^\mathsf{T} P(t) + P(t)A + Q - P(t)BR^{-1}B^\mathsf{T} P(t) \\ \text{with} \quad P(T) &= \mathcal{P}_N \end{aligned}$$

Step 2: Compute the optimal control gains

$$K(t) = -R^{-1}B^\mathsf{T} P(t)$$

Step 3: Simulate the closed-loop system

$$\dot{x}(t) = (A + BK(t))x(t) \quad \text{with} \quad x(0) = x_0$$

or (in practice) implement the control law $\mu(r, x) = K(t)x(t)$.