Linear Quadratic Regulation: Continuous-Time Case

Continuous-Time LQR Problem

Continuous-time LTI system

$$\dot{x} = Ax + Bu, \quad x(0) = x_0$$

Problem: Given a time horizon $t \in [0, t_f]$, find the optimal input $u(t)$, $t \in [0, t_f]$, that minimizes the cost function

$$J(u) = \int_0^{t_f} \underbrace{\Big(x(t)^T Q x(t) + u(t)^T R u(t)\Big)}_{\text{running cost}} dt + \underbrace{x(t_f)^T Q_f x(t_f)}_{\text{terminal cost}}.$$

- State weight matrix $Q = Q^T \succeq 0$
- Control weight matrix $R = R^T \succ 0$
- Final state weight matrix $Q_f = Q_f^T \succeq 0$
- Time horizon t_f (could be infinity)

Value Function

Value function at time $t \in [0, t_f]$ and state $x \in \mathbb{R}^n$:

$$V_t(x) = \min_{\substack{u(\tau),\ \tau \in [t, t_f] \\ x(t) = x}} \int_t^{t_f} \left(x(s)^T Q x(\tau) + u(\tau)^T R u(\tau) \right) \, d\tau + x(t_f)^T Q_f x(t_f)$$

- Optimal cost of LQR problem on a shorter time horizon $[t, t_f]$
- Optimal cost-to-go assuming the state starts from x at time t
- $V_0(x_0)$ is the optimal cost of the original LQR problem

Solution Overview

- Value function at terminal time is $V_{t_f}(x) = x^T Q_f x$
- Value function at any time $t \in [0, t_f]$ is quadratic: $V_t(x) = x^T P(t) x$
- Value functions satisfy a matrix differential equation

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

- Integrating the differential equation backward in time to yield $P(0)$
- Solution to the original problem is given by $V_0(x_0) = x_0^T P(0) x_0$
- Optimal control is a linear state feedback controller:

$$u^*(t) = -R^{-1}B^T P(t) x^*(t)$$

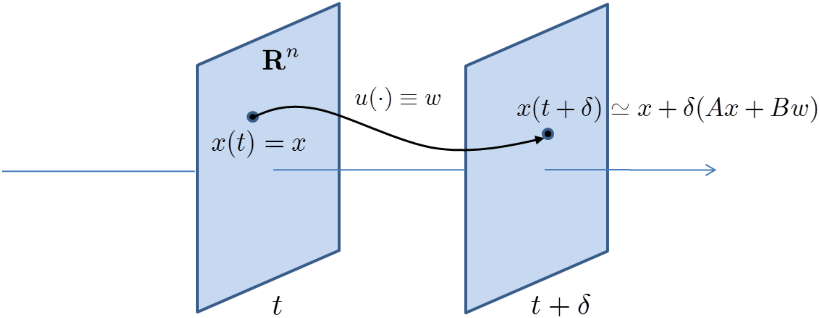
Heuristic Derivation of Value Functions

- Assume the system state starts from x at time t: $x(t) = x$
- Assume the control input is kept constant briefly:

$$u(s) \equiv w, \quad \forall s \in [t, t+\delta]$$

- At time $t+\delta$ for δ small, we have

$$x(t+\delta) = e^{A\delta}x(t) + \int_t^{t+\delta} e^{A(t+\delta-\tau)}Bu(\tau)\,d\tau \simeq x + \delta(Ax + Bw)$$



Dynamic Programming Principle

Bellman equation: The (optimal) cost-to-go at time t from x is

$$V_t(x) \simeq \min_w \left[\underbrace{\delta(x^T Q x + w^T R w)}_{\text{running cost during } [t,\, t+\delta]} + \underbrace{V_{t+\delta}(x + \delta(Ax + Bw))}_{\text{cost-to-go from time } t+\delta} \right]$$

Expand and let $\delta \to 0$, we have

$$-x^T \dot{P}(t) x = \min_w \left\{ x^T Q x + w^T R w + x^T P(t)(Ax + Bw) + (Ax + Bw)^T P(t) x \right\}$$

Continuous-Time Riccati Equation

As a result, the optimal control for state x at time t is

$$u^*(t) = w^* = -K(t)x = -\underbrace{R^{-1}B^T P(t)}_{\text{Kalman gain}}x$$

and $P(t)$ satisfies the **continuous-time Riccati differential equation**

$$\boxed{-\dot{P}(t) = Q + P(t)A + A^T P(t) - P(t)BR^{-1}B^T P(t), \quad 0 \le t \le t_f}$$

with (terminal) condition $P(t_f) = Q_f$

CT LQR Solution Algorithm

1. Set $P(t_f) = Q_f$
2. Solve the Riccati equation backward in time:

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

3. Return $V_0(x_0) = x_0^T P(0) x_0$ as the optimal cost
4. Solve forward in time the closed-loop system dynamics under the linear state feedback control $u(t) = -K(t)x(t)$:

$$\dot{x}^*(t) = (A - BK(t))x^*(t), \quad x^*(0) = x_0$$

   where $K(t)$ is the Kalman gain $K(t) = R^{-1}B^T P(t)$
5. Return $x^*(t)$ as the optimal state trajectory and return $u^*(t) = -K(t)x^*(t)$ as the optimal control input

Infinite Horizon Problem

Problem: Find the optimal control $u(t)$, $t \geq 0$, to

$$\text{minimize} \quad \int_0^\infty \left(x(t)^T Q x(t) + u(t)^T R u(t) \right) dt$$

subject to the constraint $\dot{x} = Ax + Bu$, $x(0) = x_0$

- State weight $Q \succeq 0$ and control weight $R \succ 0$
- No terminal cost

Value function:

$$V(x) = \min_u \int_0^\infty \left(x(t)^T Q x(t) + u(t)^T R u(t) \right) dt$$

subject to $\dot{x} = Ax + Bu$, $x(0) = x$

- Value function is independent of the starting time
- Optimal cost of the original problem: $V(x_0)$

Infinite Horizon Problem

Fact

If (A, B) is stabilizable, then $V(x) = x^T P x$ for some $P = P^T \succ 0$ is a finite quadratic function, and the optimal control is a static state feedback control $u(t) = -Kx(t)$, where $K = R^{-1} B^T P$.

- P solves the Continuous-time Algebraic Riccati Equation (CARE)

$$Q + PA + A^T P - PBR^{-1} B^T P = 0$$

- P can be approximated by solving the LQR problem over sufficiently large time horizon (with $Q_f = 0$), or by Matlab command `care`

Fact

If (A, B) is stabilizable and $Q = C^T C$ with (C, A) detectable, then closed-loop system $A - BK$ under the optimal control $u = -Kx$ is stable.

Alternative Solution by Lagrange Multiplier

Finite horizon LQR problem posed as **constrained optimization problem**:

$$\text{minimize } J(u) = \frac{1}{2}\int_0^{t_f} \left(x(t)^T Q x(t) + u(t)^T R u(t)\right)\, dt + \frac{1}{2}x(t_f)^T Q_f x(t_f)$$

$$\text{subject to } \dot{x}(t) = Ax(t) + Bu(t), \quad t \in [0, t_f]$$

- Optimization variables are $u(t)$ and $x(t)$ for $t \in [0, t_f]$
- Inifinite number of equality constraints, one for each $t \in [0, t_f]$

Convert the above problem to **unconstrained optimization problem**

$$L(u, x, \lambda) = J(u) + \int_0^{t_f} \lambda(t)^T \left(Ax(t) + Bu(t) - \dot{x}(t)\right)\, dt$$

- Lagrange multiplier function $\lambda : [0, t_f] \to \mathbb{R}^n$
- Original problem solution satisfies

$$\min_u J(u) = \min_{u,x} \max_\lambda L(u, x, \lambda) = \max_\lambda \min_{u,x} L(u, x, \lambda)$$

Optimality Conditions

Use integration by part to rewrite L as

$$L = J(u) + \int_0^{t_f} \left[\lambda(t)^T \left(Ax(t) + Bu(t)\right) + \dot{\lambda}(t)^T x(t)\right] dt - \lambda(t)^T x(t)\Big|_0^{t_f}$$

Optimal solution (u^*, x^*, λ^*) must satisfy $\frac{\partial L}{\partial u} = 0$ $\frac{\partial L}{\partial x} = 0$ for each $t \in [0, t_f]$:

$$\begin{aligned}
\nabla_{u(t)} L = Ru(t) + B^T\lambda(t) = 0 \quad &\Rightarrow \quad u(t) = -R^{-1}B^T\lambda(t) \\
\nabla_{x(t)} L = Qx(t) + A^T\lambda(t) + \dot{\lambda}(t) = 0 \quad &\Rightarrow \quad \dot{\lambda}(t) = -A^T\lambda(t) - Qx(t) \\
\nabla_{x(t_f)} L = Q_f x(t_f) - \lambda(t_f) = 0 \quad &\Rightarrow \quad \lambda(t_f) = Q_f x(t_f)
\end{aligned}$$

- λ is called the co-state, and satisfies the **co-state equation**:

$$\dot{\lambda}(t) = -A^T\lambda(t) - Qx(t), \quad t \in [0, t_f]$$

with terminal boundary condition $\lambda(t_f) = Q_f x(t_f)$

Hamiltonian Equation

Fact

The optimal state x^* and co-state λ^* satisfy

$$\frac{d}{dt}\begin{bmatrix} x^*(t) \\ \lambda^*(t) \end{bmatrix} = \underbrace{\begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix}}_{\textit{Hamiltonian}} \begin{bmatrix} x^*(t) \\ \lambda^*(t) \end{bmatrix}, \quad t \in [0, t_f]$$

with the boundary condition $x^*(0) = x_0$ and $\lambda^*(t_f) = Q_f x^*(t_f)$. The optimal control $u^*(t)$ is given by

$$u^*(t) = -R^{-1}B^T\lambda^*(t), \quad t \in [0, t_f]$$

- Two-point boundary value problem
- Solved numerically using the shooting method

Connecting Riccati and Hamiltonian Solutions

- Dynamical programming method says $u^*(t) = -R^{-1}B^T P(t)x^*(t)$ where $P(t)$ solves the Riccati differential equation

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

- Variational method says that $u^*(t) = -R^{-1}B^T \lambda^*(t)$ where $\lambda^*(t)$ solves the co-state equation

$$\dot{\lambda}^*(t) = -A^T \lambda^*(t) - Qx^*(t), \quad \lambda^*(t_f) = Q_f x^*(t_f)$$

- A natural guess is

$$\boxed{\lambda^*(t) = P(t)x^*(t), \quad t \in [0, t_f]}$$

- Indeed, this is the case: if $P(t)$ solves the Riccati equation, then $\lambda^*(t) := P(t)x^*(t)$ must solve the co-state equation

Matrix Hamiltonian Equations

Consider the matrix Hamiltonian differential equation

$$\frac{d}{dt}\begin{bmatrix} X(t) \\ Y(t) \end{bmatrix} = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} \begin{bmatrix} X(t) \\ Y(t) \end{bmatrix}$$

where $X(t), Y(t) \in \mathbb{R}^{n \times n}$

Fact

Suppose $X(t), Y(t) \in \mathbb{R}^{n \times n}$ solve the matrix Hamiltonian differential equation with boundary condition $X(0) = I$ and $Y(t_f) = Q_f$. Then $P(t) := Y(t)X(t)^{-1}$ is the solution to the Riccati differential equation.

- Hence the (nonlinear) Riccati differential equation can be solved via solving the (linear) matrix Hamiltonian differential equation