Linear Quadratic Regulation for Continuous-Time System

Baboo J. Cui, Yangang Cao

June 22, 2019

Contents

1 LQR for CT Case Summary

1.1 Continuous-Time LQR Problem Formulation

Given a continuous-time LTI system

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

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

$$J(u) = \underbrace{\int_0^{t_f} \left(x^T(t)Qx(t) + u^T(t)Ru(t)\right) dt}_{\text{running cost}} + \underbrace{x^T(t_f)\, Q_f x(t_f)}_{\text{terminal cost}}$$

This is similar to DT case, the sum becomes integration, again:

- **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)

1.2 Value Function of CT LQR Problem

The value function at time $t \in [0, t_f]$ is:

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

where $x \in \mathbb{R}^n$ is the state vector with the initial condition $x(t) = x$(here x can be any feasible vector). Note that:

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

1.3 Solution Overview(VIP)

- value function at terminal time obviously is quadratic: $V_{t_f}(x) = x^T Q_f x$
- value function at any time $t \in [0, t_f]$ is also **quadratic**:

  $$V_t(x) = x^T P(t) x$$

  can be proved by the way for DT case

- value functions at different times are related by the (**CT Riccati**) matrix differential equation(on each term in the matrix)

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

- integrating the differential equation backward in time yields $P(0)$
- minimum cost 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:

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

  tip: in DT case, gain depends on P of next time, here the gain depend on current time P_t

2 Derivation of Value Functions

2.1 Linear Approximation

Assume the system state starts from x at time t, namely $x(t) = x$ and assume the control input is kept **constant** briefly in a very short period time δ:

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

Attention: here $\delta \in \mathbb{R}$ is a small **scalar** instead of an operator. Then:

$$\begin{aligned} x(t+\delta) &= e^{A\delta} x(t) + \int_t^{t+\delta} e^{A(t+\delta-\tau)} Bu(\tau) d\tau \quad \text{analytic solution} \\ &\simeq x + \delta(Ax + Bw) \quad \textbf{linear approximation} \end{aligned}$$

2.2 Dynamic Programming Principle

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

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

For the first term(linear approximation):

$$\delta \left(x^T Q x + w^T R w \right) \approx \int_t^{t+\delta} \left(x^T(\tau) Q x(\tau) + u^T(\tau) R u(\tau) \right) d\tau$$

For the second term:

$$V_{t+\delta}(x + \delta(Ax + Bw)) = V_{t+\delta}(x + \delta \dot{x}_t) \approx V_{t+\delta}(x_{t+\delta})$$

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

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

3 Conclusions

Here is the list of conclusions.

3.1 Continuous-Time Riccati Equation

The optimal control for state x at time t is

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

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

$$-\dot{P}(t) = Q + P(t)A + A^TP(t) - P(t)BR^{-1}B^TP(t), 0 \leq t \leq t_f$$

with (terminal) condition $P(t_f) = Q_f$. This is not easy to solve because non-linear term $P(t)BR^{-1}B^TP(t)$. A CT system can be converted to DT to solve for simplicity.

3.2 CT LQR Solution Algorithm Summary

1. set $P(t_f) = Q_f$
2. solve the Riccati differential equation backward in time:

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

3. return $V_0(x_0) = x_0^TP(0)x_0$ as the optimal cost
4. solve forward in time the closed-loop system dynamics under the linear state feedback control:

$$u(t) = -K(t)x(t)$$

   with optimal system dynamic:

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

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

4 Infinite Horizon Problem

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

$$\int_0^\infty \left(x^T(t)Qx(t) + u^T(t)Ru(t)\right) dt$$

subject to system dynamic constraints:

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

Again state weight $Q \succeq 0$ and control weight $R \succ 0$, and there is no terminal cost. And value function is:

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

- value function is independent of the starting time(steady state)
- optimal cost of the original problem: $V(x_0)$

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()*

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.

5 Alternative Solution by Lagrange Multiplier

Finite horizon LQR problem can be considered as **constrained optimization problem**:

$$\begin{aligned} \min_u J(u) &= \frac{1}{2} \int_0^{t_f} \left(x^T(t) Q x(t) + u^T(t) R u(t) \right) dt + \frac{1}{2} x^T(t_f) Q_f x(t_f) \\ \text{s.t.} \quad \dot{x}(t) &= Ax(t) + Bu(t), t \in [0, t_f] \end{aligned}$$

- optimization variables are $u(t), t \in [0, t_f]$
- there are **infinite** number of equality constraints, one for each $t \in [0, t_f]$

Convert the above problem to **unconstrained optimization problem** by **Lagrange** method:

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

- **Lagrange multiplier function**:$\lambda : [0, t_f] \to \mathbb{R}^n$, same dimension as state variable, infinite number of λ for each t
- original problem solution:

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

  this is **strong duality** and detailed info are covered in **game theory**. A quick intuitive way to see this is big λ force the state trajectory in track while at the same time treat x and u independently

Optimal solution (u^*, x^*, λ^*) must satisfy:

$$\frac{\partial L}{\partial u} = 0 \quad \text{and} \quad \frac{\partial L}{\partial x} = 0$$

Use integration by part to rewrite L as:

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

Needle-like variations: for each $t \in [0, t_f]$, perturb $u(t)$ and $x(t)$ locally:

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

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 $\lambda(t_f) = Q_f x(t_f)$.

- can be adopted for state or input constraints
- known as Pontryagin's maximum principle

6 Hamiltonian Equation

The optimal control $u^*(t)$ is given by

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

while the optimal state x^* and co-state λ^* satisfies

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

with two-point boundary condition: $x^*(0) = x_0, \lambda^*(t_f) = Q_f x^*(t_f)$

- two-point boundary value problem
- solved numerically using the shooting method
- nice form that a linear system govern both x and λ
- use shooting method to solve it(adaptive guess)

6.1 Connecting Riccati and Hamiltonian Solutions

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

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

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

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

- a natural guess is

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

Indeed, this is the case: if $P(t)$ solves the Riccati equation, then

$$\lambda^*(t) := P(t)x^*(t)$$

must solve co-state equation.

6.2 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 $X(t), Y(t) \in \mathbb{R}^{n\times n}$, Suppose $X(t), Y(t) \in \mathbb{R}^{n\times n}$ solve the matrix Hamiltonian differential equation with boundary condition

$$X(t_f) = I \quad \text{and} \quad Y(t_f) = Q_f$$

Then $P(t) := Y(t)X(t)^{-1}$ is the solution to Riccati differential equation. Hence the (nonlinear) Riccati differential equation can be solved via solving the (linear) matrix Hamiltonian differential equation(good for numerical calculation).