ECE599 Lecture Notes: LQR/iLQR

Fei Liu

May 13, 2025

1 LQR with a recursive formulation

The LQR problem is formulated as the minimization of the cost

$$c(x_1, \mathbf{u}) = c_T(x_T) + \sum_{t=1}^{T-1} c_t(x_t, u_t) = x_T^\top \mathbf{Q}_T x_T + \sum_{t=1}^{T-1} \left(x_t^\top \mathbf{Q}_t x_t + u_t^\top \mathbf{R}_t u_t \right), \quad \text{s.t.} \quad x_{t+1} = A_t x_t + B_t u_t, \tag{1}$$

where c without index refers to the total cost (cumulative cost).

We also define the partial cumulative cost

$$v_t(x_t, u_{t:T-1}) = c_T(x_T) + \sum_{s=t}^{T-1} c_s(x_s, u_s), \tag{2}$$

which depends on the states and control commands, except for v_T that only depends on the state.

We define the *value function* as the value taken by v_t when applying optimal control commands, namely

$$\hat{v}_t(x_t) = \min_{u_{t:T-1}} v_t(x_t, u_{t:T-1}). \tag{3}$$

We will use the dynamic programming principle to reduce the minimization in Eq. 3 over the entire sequence of control commands $u_{t:T-1}$ to a sequence of minimization problems over control commands at a single time step, by proceeding backwards in time.

By inserting Eq. 2 in Eq. 3, we observe that

$$\begin{aligned} \hat{v}_t(x_t) &= \min_{u_{t:T-1}} \left(c_T(x_T) + \sum_{s=t}^{T-1} c_s(x_s, u_s) \right) \\ &= \min_{u_t} \left(c(x_t, u_t) + \min_{u_{t+1:T-1}} \left(c_T(x_T) + \sum_{s=t+1}^{T-1} c_s(x_s, u_s) \right) \right) \\ &= \min_{u_t} \underbrace{c(x_t, u_t) + \hat{v}_{t+1}(x_{t+1})}_{q_t(x_t, u_t)} \end{aligned} \tag{4}$$

where q_t is called the *q-function*.

1.1 Inductive Step

We start from the last time step T, and by relabeling $\mathbf{Q}_T$ as $\mathbf{V}_T$, we can see that $\hat{v}_T(x_T) = c_T(x_T)$ is independent of the control commands, taking the quadratic error form

$$\hat{v}_T = x_T^\top \mathbf{V}_T x_T, \tag{5}$$

which only involves the final state x_T. We will show that $\hat{v}_{T-1}$ has the same quadratic form as $\hat{v}_T$, enabling the backward recursive computation of $\hat{v}_t$ from $t = T-1$ to $t = 1$.

Assume that at time $t+1$, the value function is quadratic in the state:

$$\hat{v}_{t+1}(x_{t+1}) = x_{t+1}^\top \mathbf{V}_{t+1} x_{t+1}$$

According to Eq. 4, the dynamic programming recursion takes the form

$$\hat{v}_t = \min_{u_t} \begin{bmatrix} x_t \\ u_t \end{bmatrix}^\top \begin{bmatrix} \mathbf{Q}_t & 0 \\ 0 & \mathbf{R}_t \end{bmatrix} \begin{bmatrix} x_t \\ u_t \end{bmatrix} + x_{t+1}^\top \mathbf{V}_{t+1} x_{t+1}. \tag{6}$$

By substituting $x_{t+1} = A_t x_t + B_t u_t$ into above, $\hat{v}_t$ can be rewritten as

$$\hat{v}_t = \min_{u_t} \begin{bmatrix} x_t \\ u_t \end{bmatrix}^\top \begin{bmatrix} Q_{xx,t} & Q_{xu,t} \\ Q_{ux,t} & Q_{uu,t} \end{bmatrix} \begin{bmatrix} x_t \\ u_t \end{bmatrix}, \quad \text{where} \begin{cases} Q_{xx,t} = A_t^\top \mathbf{V}_{t+1} A_t + \mathbf{Q}_t, \\ Q_{uu,t} = B_t^\top \mathbf{V}_{t+1} B_t + \mathbf{R}_t, \\ Q_{ux,t} = B_t^\top \mathbf{V}_{t+1} A_t. \end{cases} \tag{7}$$

An optimal control command $\hat{u}_t$ can be computed by differentiating above with respect to u_t and equating to zero, providing a feedback law

$$\hat{u}_t = K_t x_t, \quad \text{with} \quad K_t = -Q_{uu,t}^{-1} Q_{ux,t}. \tag{8}$$

By introducing 8 back into Eq. 9, the resulting value function $\hat{v}_t$ has the quadratic form

$$\hat{v}_t = x_t^\top \mathbf{V}_t x_t, \quad \text{with} \quad \mathbf{V}_t = Q_{xx,t} - Q_{ux,t}^\top Q_{uu,t}^{-1} Q_{ux,t}. \tag{9}$$

That's

$$\mathbf{V}_t = \mathbf{Q}_t + A_t^\top \mathbf{V}_{t+1} A_t - A_t^\top \mathbf{V}_{t+1} B_t (\mathbf{R}_t + B_t^\top \mathbf{V}_{t+1} B_t)^{-1} B_t^\top \mathbf{V}_{t+1} A_t$$

We observe that Eq. 9 has the same quadratic form as Eq. 5, so that Eq. 4 can be solved recursively. We thus obtain a backward recursion procedure in which $\mathbf{V}_t$ is evaluated recursively from $t = T-1$ to $t = 1$, starting from $\mathbf{V}_T = \mathbf{Q}_T$, which corresponds to a Riccati equation.

After all feedback gain matrices K_t have been computed by backward recursion, a forward recursion can be used to compute the evolution of the state, starting from x_1, by using the linear system $x_{t+1} = A_t x_t + B_t u_t$ and the control policy $u_t = K_t x_t$.

2 iLQR

Optimal control problems are defined by a cost function $\sum_{t=1}^{T} c(x_t, u_t)$ to minimize and a dynamical system $x_{t+1} = d(x_t, u_t)$ describing the evolution of a state x_t driven by control commands u_t during a time window of length T.

Iterative LQR (iLQR) solves such constrained nonlinear models by carrying out Taylor expansions on the cost and on the dynamical system so that a solution can be found iteratively by solving a LQR problem at each iteration, similarly to differential dynamic programming (DDP) approaches.

iLQR employs a first order Taylor expansion of the dynamical system $x_{t+1} = d(x_t, u_t)$ around the point $(\hat{x}_t, \hat{u}_t)$, namely

$$\begin{aligned} & x_{t+1} \approx d(\hat{x}_t, \hat{u}_t) + \frac{\partial d}{\partial x_t}(x_t - \hat{x}_t) + \frac{\partial d}{\partial u_t}(u_t - \hat{u}_t) \\ \iff \quad & \Delta x_{t+1} \approx A_t \Delta x_t + B_t \Delta u_t \end{aligned} \tag{10}$$

with residual vectors

$$\Delta x_t = x_t - \hat{x}_t, \quad \Delta u_t = u_t - \hat{u}_t,$$

and Jacobian matrices

$$A_t = \left.\frac{\partial d}{\partial x_t}\right|_{\hat{x}_t,\hat{u}_t}, \quad B_t = \left.\frac{\partial d}{\partial u_t}\right|_{\hat{x}_t,\hat{u}_t}.$$

The cost function $c(x_t, u_t)$ for time step t can similarly be approximated by a second order Taylor expansion around the point $(\hat{x}_t, \hat{u}_t)$, namely

$$c(x_t, u_t) \approx c(\hat{x}_t, \hat{u}_t) + \Delta x_t^\top \frac{\partial c}{\partial x_t} + \Delta u_t^\top \frac{\partial c}{\partial u_t} + \frac{1}{2}\Delta x_t^\top \frac{\partial^2 c}{\partial x_t^2}\Delta x_t \tag{11}$$

$$+ \Delta x_t^\top \frac{\partial^2 c}{\partial x_t \partial u_t}\Delta u_t + \frac{1}{2}\Delta u_t^\top \frac{\partial^2 c}{\partial u_t^2}\Delta u_t \tag{12}$$

$$\iff \quad c(x_t, u_t) \approx c(\hat{x}_t, \hat{u}_t) + \frac{1}{2}\begin{bmatrix}1\\ \Delta x_t\\ \Delta u_t\end{bmatrix}^\top \begin{bmatrix}0 & g_{x,t}^\top & g_{u,t}^\top\\ g_{x,t} & H_{xx,t} & H_{xu,t}\\ g_{u,t} & H_{ux,t} & H_{uu,t}\end{bmatrix}\begin{bmatrix}1\\ \Delta x_t\\ \Delta u_t\end{bmatrix}, \tag{13}$$

with gradients

$$g_{x,t} = \left.\frac{\partial c}{\partial x_t}\right|_{\hat{x}_t,\hat{u}_t}, \quad g_{u,t} = \left.\frac{\partial c}{\partial u_t}\right|_{\hat{x}_t,\hat{u}_t},$$

and Hessian matrices

$$H_{xx,t} = \left.\frac{\partial^2 c}{\partial x_t^2}\right|_{\hat{x}_t,\hat{u}_t}, \quad H_{uu,t} = \left.\frac{\partial^2 c}{\partial u_t^2}\right|_{\hat{x}_t,\hat{u}_t}, \quad H_{ux,t} = H_{xu,t}^\top = \left.\frac{\partial^2 c}{\partial u_t \partial x_t}\right|_{\hat{x}_t,\hat{u}_t}.$$

2.1 Recursive formulation of iLQR

Similarly to Eq. 4, we have here

$$\hat{v}_t(\Delta x_t) = \min_{\Delta u_t} \underbrace{c_t(\Delta x_t, \Delta u_t) + \hat{v}_{t+1}(\Delta x_{t+1})}_{q_t(\Delta x_t, \Delta u_t)} \tag{14}$$

Similarly to LQR, by starting from the last time step T, $\hat{v}_T(\Delta x_T)$ is independent of the control commands. By relabeling $g_{x,T}$ and $H_{xx,T}$ as $v_{x,T}$ and $V_{xx,T}$, we can show that $\hat{v}_{T-1}$ has the same quadratic form as $\hat{v}_T$, enabling the backward recursive computation of $\hat{v}_t$ from $t = T-1$ to $t = 1$.

This dynamic programming recursion takes the form

$$\hat{v}_{t+1} = \begin{bmatrix}1\\ \Delta x_{t+1}\end{bmatrix}^\top \begin{bmatrix}0 & v_{x,t+1}^\top\\ v_{x,t+1} & V_{xx,t+1}\end{bmatrix}\begin{bmatrix}1\\ \Delta x_{t+1}\end{bmatrix}, \tag{15}$$

and we then have with Eq. 14 that

$$\hat{v}_t = \min_{\Delta u_t} \begin{bmatrix}1\\ \Delta x_t\\ \Delta u_t\end{bmatrix}^\top \begin{bmatrix}0 & g_{x,t}^\top & g_{u,t}^\top\\ g_{x,t} & H_{xx,t} & H_{xu,t}\\ g_{u,t} & H_{ux,t} & H_{uu,t}\end{bmatrix}\begin{bmatrix}1\\ \Delta x_t\\ \Delta u_t\end{bmatrix} + \begin{bmatrix}1\\ \Delta x_{t+1}\end{bmatrix}^\top \begin{bmatrix}0 & v_{x,t+1}^\top\\ v_{x,t+1} & V_{xx,t+1}\end{bmatrix}\begin{bmatrix}1\\ \Delta x_{t+1}\end{bmatrix}, \tag{16}$$

By substituting $\Delta x_{t+1} = A_t \Delta x_t + B_t \Delta u_t$ into above, $\hat{v}_t$ can be rewritten as

$$\hat{v}_t = \min_{\Delta u_t} \begin{bmatrix} 1 \\ \Delta x_t \\ \Delta u_t \end{bmatrix}^\top \begin{bmatrix} 0 & q_{x,t}^\top & q_{u,t}^\top \\ q_{x,t} & Q_{xx,t} & Q_{xu,t} \\ q_{u,t} & Q_{ux,t} & Q_{uu,t} \end{bmatrix} \begin{bmatrix} 1 \\ \Delta x_t \\ \Delta u_t \end{bmatrix}, \quad \text{where} \begin{cases} q_{x,t} = g_{x,t} + A_t^\top v_{x,t+1}, \\ q_{u,t} = g_{u,t} + B_t^\top v_{x,t+1}, \\ Q_{xx,t} \approx H_{xx,t} + A_t^\top V_{xx,t+1} A_t, \\ Q_{uu,t} \approx H_{uu,t} + B_t^\top V_{xx,t+1} B_t, \\ Q_{ux,t} \approx H_{ux,t} + B_t^\top V_{xx,t+1} A_t. \end{cases} \tag{17}$$

with gradients

$$g_{x,t} = \frac{\partial c}{\partial x_t}\bigg|_{\hat{x}_t,\hat{u}_t}, \quad g_{u,t} = \frac{\partial c}{\partial u_t}\bigg|_{\hat{x}_t,\hat{u}_t}, \quad v_{x,t} = \frac{\partial \hat{v}}{\partial x_t}\bigg|_{\hat{x}_t,\hat{u}_t}, \quad q_{x,t} = \frac{\partial q}{\partial x_t}\bigg|_{\hat{x}_t,\hat{u}_t}, \quad q_{u,t} = \frac{\partial q}{\partial u_t}\bigg|_{\hat{x}_t,\hat{u}_t},$$

Jacobian matrices

$$A_t = \frac{\partial d}{\partial x_t}\bigg|_{\hat{x}_t,\hat{u}_t}, \quad B_t = \frac{\partial d}{\partial u_t}\bigg|_{\hat{x}_t,\hat{u}_t},$$

and Hessian matrices

$$H_{xx,t} = \frac{\partial^2 c}{\partial x_t^2}\bigg|_{\hat{x}_t,\hat{u}_t}, \quad H_{uu,t} = \frac{\partial^2 c}{\partial u_t^2}\bigg|_{\hat{x}_t,\hat{u}_t}, \quad H_{ux,t} = \frac{\partial^2 c}{\partial u_t \partial x_t}\bigg|_{\hat{x}_t,\hat{u}_t}, \quad V_{xx,t} = \frac{\partial^2 \hat{v}}{\partial x_t^2}\bigg|_{\hat{x}_t,\hat{u}_t},$$

$$Q_{xx,t} = \frac{\partial^2 q}{\partial x_t^2}\bigg|_{\hat{x}_t,\hat{u}_t}, \quad Q_{uu,t} = \frac{\partial^2 q}{\partial u_t^2}\bigg|_{\hat{x}_t,\hat{u}_t}, \quad Q_{ux,t} = \frac{\partial^2 q}{\partial u_t \partial x_t}\bigg|_{\hat{x}_t,\hat{u}_t}.$$

Minimizing Eq. 17 w.r.t. Δu_t can be achieved by differentiating the equation and equating to zero, yielding the controller

$$\Delta \hat{u}_t = k_t + K_t \Delta x_t, \quad \text{with} \quad \begin{cases} k_t = -Q_{uu,t}^{-1} q_{u,t}, \\ K_t = -Q_{uu,t}^{-1} Q_{ux,t}, \end{cases} \tag{18}$$

where k_t is a feedforward command and K_t is a feedback gain matrix.

By inserting above Eq. 18 into Eq. 17, we get the recursive updates

$$\begin{aligned} v_{x,t} &= q_{x,t} - Q_{ux,t}^\top Q_{uu,t}^{-1} q_{u,t}, \\ V_{xx,t} &= Q_{xx,t} - Q_{ux,t}^\top Q_{uu,t}^{-1} Q_{ux,t}. \end{aligned} \tag{19}$$

In this recursive iLQR formulation, at each iteration step of the optimization algorithm, the nominal trajectories $(\hat{x}_t, \hat{u}_t)$ are refined, together with feedback matrices K_t. Thus, at each iteration step of the optimization algorithm, a backward and forward recursion is performed to evaluate these vectors and matrices. There is thus two types of iterations: one for the optimization algorithm, and one for the dynamic programming recursion performed at each given iteration.

After convergence, by using Eq. 18 and the nominal trajectories in the state and control spaces $(\hat{x}_t, \hat{u}_t)$, the resulting controller at each time step t is

$$u_t = \hat{u}_t + K_t(\hat{x}_t - x_t), \tag{20}$$

where the evolution of the state is described by $x_{t+1} = d(x_t, u_t)$.

The complete iLQR procedure is described in Algorithm 1 (including backtracking line search).

2.2 Updates by considering step sizes

iLQR typically requires to estimate a step size α at each iteration to scale the control command updates.

For the recursive formulation, this can be achieved by setting the update as

$$\hat{u}_t \leftarrow \hat{u}_t + \alpha\, k_t + K_t \Delta x_t. \tag{21}$$

In practice, a simple backtracking line search procedure can be considered, as shown in Algorithm 1, by considering a small value for $\alpha_{\min}$.

Algorithm 1 Recursive formulation of iLQR

Define cost $c(\cdot)$, dynamics $d(\cdot)$, x_1, $\alpha_{\min}$, $\Delta_{\min}$
Initialize all $\hat{u}_t$
repeat
    Compute $\hat{x}$ using $\hat{u}$, $d(\cdot)$, and x_1
`// Evaluating derivatives`
    Compute all A_t, B_t in Eq. 10
    Compute all $g_{\cdot,t}$ and $H_{\cdot,t}$ in Eq. 11
`// Backward pass`
    Set $v_{x,T} = g_{x,T}$ and $V_{xx,T} = H_{xx,T}$
    for $t \leftarrow T-1$ to 1 **do**
        Compute $q_{\cdot,t}$ and $Q_{\cdot,t}$ with Eq. 17
        Compute k_t and K_t with Eq. 18
        Compute $v_{x,t}$ and $V_{xx,t}$ with Eq. 19
    end for
`// Forward pass`
    $\alpha \leftarrow 2$
    while $c(\hat{u} + \alpha\Delta\hat{u}) > c(\hat{u})$ **and** $\alpha > \alpha_{\min}$ **do**
        $\alpha \leftarrow \alpha/2$
        for $t \leftarrow 1$ to $T-1$ **do**
            Update $\hat{u}_t$ with Eq. 21
            Compute $\hat{x}_{t+1}$ with $d(\cdot)$
        end for
    end while
until $\|\Delta\hat{u}\| < \Delta_{\min}$
Use Eq. 20 and $d(\cdot)$ for reproduction

Readings

1. https://rexlab.ri.cmu.edu/papers/iLQR_Tutorial.pdf