AA 203
Optimal and Learning-Based Control
LQR-based methods

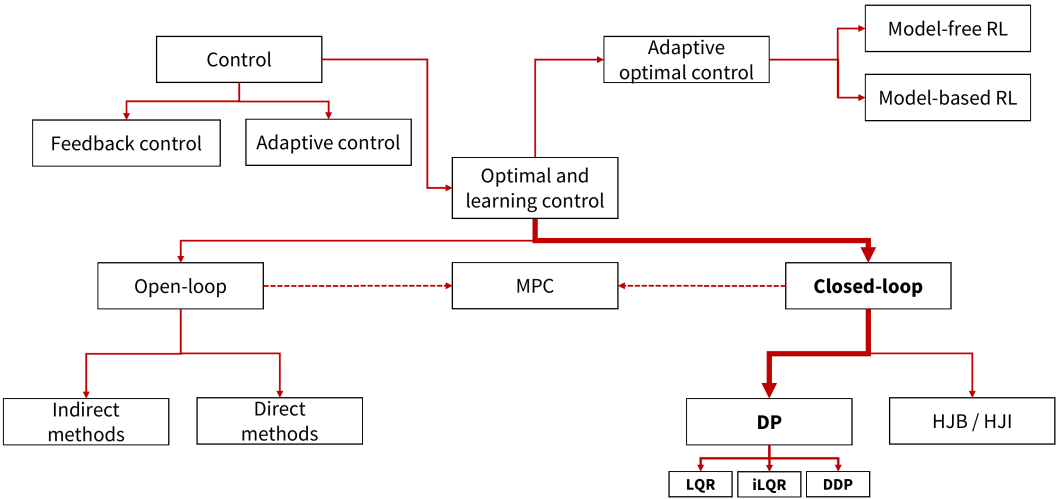
Spencer M. Richards

Autonomous Systems Laboratory, Stanford University

April 24, 2023
(last updated May 3, 2023)

Stanford
University

Course overview



Agenda

1. LQR feedback for linear systems with quadratic costs
2. Linear and nonlinear tracking LQR
3. iLQR and DDP for trajectory optimization

Agenda

1. LQR feedback for linear systems with quadratic costs
2. Linear and nonlinear tracking LQR
3. iLQR and DDP for trajectory optimization

Review: LQR feedback for linear systems with quadratic costs

Consider the discrete-time OCP

$$\begin{aligned} \underset{u}{\text{minimize}} \;\; & \frac{1}{2} x_T^\mathsf{T} Q_T x_T + \sum_{t=0}^{T-1} \left(\frac{1}{2} x_t^\mathsf{T} Q_t x_t + \frac{1}{2} u_t^\mathsf{T} R_t u_t + x_t^\mathsf{T} S_t u_t \right) \\ \text{subject to} \;\; & x_{t+1} = A_t x_t + B_t u_t, \;\; \forall t \in \{0, 1, \ldots, T-1\} \end{aligned}$$

which is parameterized by the initial state x_0 and minimized over the control inputs u alone, for $Q_T \succeq 0$, $Q_t \succeq 0$, and $R_t \succ 0$.

We solved this recursively via dynamic programming, during which we encountered the Bellman optimality equation

$$J_t^*(x_t) = \min_{u_t} \underbrace{\frac{1}{2} \left(\begin{pmatrix} x_t \\ u_t \end{pmatrix}^\mathsf{T} \begin{bmatrix} Q_t & S_t \\ S_t^\mathsf{T} & R_t \end{bmatrix} \begin{pmatrix} x_t \\ u_t \end{pmatrix} + \underbrace{(A_t x_t + B_t u_t)^\mathsf{T} P_{t+1} (A_t x_t + B_t u_t)}_{= J_{t+1}^*(x_{t+1})} \right)}_{\text{state-action value function } Q^*(x_t, u_t)}$$

Review: LQR feedback for linear systems with quadratic costs

Consider the discrete-time OCP

$$\begin{aligned} \underset{u}{\text{minimize}} \;\; & J_0(x_0) \coloneqq \frac{1}{2} x_T^\mathsf{T} Q_T x_T + \sum_{t=0}^{T-1} \left(\frac{1}{2} x_t^\mathsf{T} Q_t x_t + \frac{1}{2} u_t^\mathsf{T} R_t u_t + x_t^\mathsf{T} S_t u_t \right) \\ \text{subject to} \;\; & x_{t+1} = A_t x_t + B_t u_t, \;\; \forall t \in \{0, 1, \ldots, T-1\} \end{aligned}$$

which is parameterized by $x_0 \in \mathbb{R}^n$, $Q_T \succeq 0$, $Q_t \succeq 0$, and $R_t \succ 0$.

The optimal control $u^* = \pi^*(t, x) = K_t x$ is *closed-loop* and *linear*. It can be computed offline via the backwards Riccati recursion

$$\begin{aligned} P_T &\coloneqq Q_T \\ K_t &= -(R_t + B_t^\mathsf{T} P_{t+1} B_t)^{-1} (B_t^\mathsf{T} P_{t+1} A_t + S_t^\mathsf{T}) \\ P_t &= Q_t + A_t^\mathsf{T} P_{t+1} A_t - (A_t^\mathsf{T} P_{t+1} B_t + S_t)(R_t + B_t^\mathsf{T} P_{t+1} B_t)^{-1} (B_t^\mathsf{T} P_{t+1} A_t + S_t^\mathsf{T}) \\ &= Q_t + A_t^\mathsf{T} P_{t+1} (A_t + B_t K_t) + S_t K_t \end{aligned}$$

LQR feedback for affine systems with quadratic and linear costs

Consider the discrete-time LQR problem with $Q_T \succeq 0$, $Q_t \succeq 0$, $R_t \succ 0$, and now

$$\begin{aligned}
\ell_T(x_T) &= \frac{1}{2} x_T^\mathsf{T} Q_T x_T + q_T^\mathsf{T} x_T + \alpha_T \\
\ell_t(x_t, u_t) &= \frac{1}{2} x_t^\mathsf{T} Q_t x_t + \frac{1}{2} u_t^\mathsf{T} R_t u_t + x_t^\mathsf{T} S_t u_t + q_t^\mathsf{T} x_t + r_t^\mathsf{T} u_t + \alpha_t, \ \forall t \in \{0, 1, \ldots, T-1\} \\
f(t, x_t, u_t) &= A_t x_t + B_t u_t + c_t, \ \forall t \in \{0, 1, \ldots, T-1\}
\end{aligned}$$

Define the 0th-, 1st-, and 2nd-order terms

$$\begin{aligned}
\eta_t &\coloneqq \alpha_t + \beta_{t+1} + p_{t+1}^\mathsf{T} c_t + \frac{1}{2} c_t^\mathsf{T} P_{t+1} c_t & H_{xx,t} &\coloneqq Q_t + A_t^\mathsf{T} P_{t+1} A_t \\
h_{x,t} &\coloneqq q_t + A_t^\mathsf{T} (p_{t+1} + P_{t+1} c_t) & H_{uu,t} &\coloneqq R_t + B_t^\mathsf{T} P_{t+1} B_t \\
h_{u,t} &\coloneqq r_t + B_t^\mathsf{T} (p_{t+1} + P_{t+1} c_t) & H_{xu,t} &\coloneqq S_t + A_t^\mathsf{T} P_{t+1} B_t
\end{aligned}$$

LQR feedback for affine systems with quadratic and linear costs

Define the 0^{th}-, 1^{st}-, and 2^{nd}-order terms

$$\eta_t \coloneqq \alpha_t + \beta_{t+1} + p_{t+1}^\mathsf{T} c_t + \frac{1}{2} c_t^\mathsf{T} P_{t+1} c_t \qquad H_{xx,t} \coloneqq Q_t + A_t^\mathsf{T} P_{t+1} A_t$$
$$h_{x,t} \coloneqq q_t + A_t^\mathsf{T}(p_{t+1} + P_{t+1} c_t) \qquad H_{uu,t} \coloneqq R_t + B_t^\mathsf{T} P_{t+1} B_t$$
$$h_{u,t} \coloneqq r_t + B_t^\mathsf{T}(p_{t+1} + P_{t+1} c_t) \qquad H_{xu,t} \coloneqq S_t + A_t^\mathsf{T} P_{t+1} B_t$$

The optimal control $u^* = \pi^*(t, x) = K_t x + k_t$ is *closed-loop* and *affine*, and given by

$$P_T \coloneqq Q_T \qquad K_t = -H_{uu,t}^{-1} H_{xu,t}^\mathsf{T} \qquad P_t = H_{xx,t} + H_{xu,t} K_t$$
$$p_T \coloneqq q_T \qquad k_t = -H_{uu,t}^{-1} h_{u,t} \qquad p_t = h_{x,t} + H_{xu,t} k_t$$
$$\beta_T \coloneqq \alpha_T \qquad \beta_t = \eta_t + \frac{1}{2} h_{u,t}^\mathsf{T} k_t$$

with cost-to-go $J_t^*(x_t) = \frac{1}{2} x_t^\mathsf{T} P_t x_t + p_t^\mathsf{T} x_t + \beta_t$.

Agenda

1. LQR feedback for linear systems with quadratic costs
2. Linear and nonlinear tracking LQR
3. iLQR and DDP for trajectory optimization

Tracking LQR for affine systems

Suppose we know a *nominal trajectory* $(\bar{x}, \bar{u})$ with affine dynamics, i.e.,

$$\bar{x}_{t+1} = A_t \bar{x}_t + B_t \bar{u}_t + c_t, \ \forall t \in \{0, 1, \ldots, T-1\}.$$

Define the errors $\tilde{x}_t \coloneqq x_t - \bar{x}_t$ and $\tilde{u}_t \coloneqq u_t - \bar{u}_t$. Then the *error dynamics* are given by

$$\tilde{x}_{t+1} = A_t \tilde{x}_t + B_t \tilde{u}_t.$$

If we want to track $(\bar{x}, \bar{u})$, we can use the quadratic cost function

$$J_0(\tilde{x}_0) = \frac{1}{2} \tilde{x}_T^\mathsf{T} Q_T \tilde{x}_T + \sum_{t=0}^{T-1} \left(\frac{1}{2} \tilde{x}_t^\mathsf{T} Q_t \tilde{x}_t + \frac{1}{2} \tilde{u}_t^\mathsf{T} R_t \tilde{u}_t \right)$$

with $Q_T \succeq 0$, $Q_t \succeq 0$, and $R_t \succ 0$ to penalize *deviations* of (x, u) from $(\bar{x}, \bar{u})$.

Standard LQR for this problem gives us an optimal policy such that $\tilde{u}_t^* = K_t \tilde{x}_t$, so

$$u_t^* = \pi^*(t, x_t, \bar{x}_t, \bar{u}_t) = \bar{u}_t + K_t(x_t - \bar{x}_t).$$

Tracking LQR for nonlinear systems

Suppose we know a *nominal trajectory* $(\bar{x}, \bar{u})$ with nonlinear dynamics, i.e.,

$$\bar{x}_{t+1} = f(t, \bar{x}_t, \bar{u}_t), \ \forall t \in \{0, 1, \dots, T-1\}.$$

Then the error dynamics are *approximately* given by

$$x_{t+1} \approx f(t, \bar{x}_t, \bar{u}_t) + \frac{\partial f}{\partial x}(t, \bar{x}_t, \bar{u}_t)(x_t - \bar{x}_t) + \frac{\partial f}{\partial u}(t, \bar{x}_t, \bar{u}_t)(u_t - \bar{u}_t)$$

$$\tilde{x}_{t+1} \approx \underbrace{\frac{\partial f}{\partial x}(t, \bar{x}_t, \bar{u}_t)}_{=:A_t} \tilde{x}_t + \underbrace{\frac{\partial f}{\partial u}(t, \bar{x}_t, \bar{u}_t)}_{=:B_t} \tilde{u}_t$$

If we remain "close" to $(\bar{x}, \bar{u})$, then we can use standard LQR with the quadratic cost function from the previous slide to compute a *locally* optimal policy
$u_t^* = \pi^*(t, x_t, \bar{x}_t, \bar{u}_t) = \bar{u}_t + K_t(x_t - \bar{x}_t)$.

Agenda

1. LQR feedback for linear systems with quadratic costs
2. Linear and nonlinear tracking LQR
3. iLQR and DDP for trajectory optimization

LQR-based methods for solving unconstrained nonlinear OCPs

Consider the discrete-time OCP

$$\begin{aligned} \underset{\bar{x},\bar{u}}{\text{minimize}} \;\; & J(\bar{x},\bar{u}) \coloneqq \ell_T(\bar{x}_T) + \sum_{t=0}^{T-1} \ell(t,\bar{x}_t,\bar{u}_t) \\ \text{subject to} \;\; & \bar{x}_{t+1} = f(t,\bar{x}_t,\bar{u}_t), \;\; \forall t \in \{0,1,\dots,T-1\} \\ & \bar{x}_0 = x_0 \end{aligned}$$

We can use LQR to approximately solve this problem for an *open-loop* trajectory $(\bar{x},\bar{u})$ and a *locally optimal policy* $u_t^* = \pi_t^*(t,x_t,\bar{x}_t,\bar{u}_t) = \bar{u}_t + K_t(x_t - \bar{x}_t)$ simultaneously!

Specifically, we will consider two *iterative* methods:

iterative LQR (iLQR) Approximate the cost and dynamics as quadratic and affine, respectively, then solve the optimal Bellman equation recursively.

differential dynamic programming (DDP) Approximate the value function and Bellman equation as quadratic, then solve the optimal Bellman equation recursively.

Iterative LQR (iLQR)

In *iterative LQR (iLQR)*, we approximate the cost and dynamics as quadratic and affine, respectively, then exactly solve the resulting LQR problem.

We initialize $\bar{u}$ and start with a "rollout" of the nonlinear dynamics $\bar{x}_{t+1} = f(t, \bar{x}_t, \bar{u}_t)$ to compute $\bar{x}$ and $J(\bar{x}, \bar{u})$. Then we approximate the dynamics and cost as

$$\tilde{x}_{t+1} \approx \underbrace{\frac{\partial f}{\partial x}(t, \bar{x}_t, \bar{u}_t)}_{=:A_t} \tilde{x}_t + \underbrace{\frac{\partial f}{\partial u}(t, \bar{x}_t, \bar{u}_t)}_{=:B_t} \tilde{u}_t + \underbrace{0}_{=:c_t}$$

$$\ell_T(x_T) \approx \underbrace{\ell_T(\bar{x}_T)}_{=:\alpha_T} + \underbrace{\nabla \ell_T(\bar{x}_T)}_{=:q_T}^\mathsf{T} \tilde{x}_T + \frac{1}{2} \tilde{x}_T \underbrace{\nabla^2 \ell_T(\bar{x}_T)}_{=:Q_T} \tilde{x}_T$$

$$\begin{aligned} \ell_t(t, x_t, u_t) \approx{}& \underbrace{\ell_t(t, \bar{x}_t, \bar{u}_t)}_{=:\alpha_t} + \underbrace{\nabla_x \ell(t, \bar{x}_t, \bar{u}_t)}_{=:q_t}^\mathsf{T} \tilde{x}_t + \underbrace{\nabla_u \ell(t, \bar{x}_t, \bar{u}_t)}_{=:r_t}^\mathsf{T} \tilde{u}_t \\ &+ \frac{1}{2} \tilde{x}_t^\mathsf{T} \underbrace{\nabla^2_{xx} \ell_t(t, \bar{x}_t, \bar{u}_t)}_{=:Q_t} \tilde{x}_t + \frac{1}{2} \tilde{u}_t^\mathsf{T} \underbrace{\nabla^2_{uu} \ell_t(t, \bar{x}_t, \bar{u}_t)}_{=:R_t} \tilde{u}_t + \tilde{x}_t^\mathsf{T} \underbrace{\nabla^2_{xu} \ell_t(t, \bar{x}_t, \bar{u}_t)}_{=:S_t} \tilde{u}_t \end{aligned}$$

Iterative LQR (iLQR)

Now we solve the general LQR problem

$$\begin{aligned} &\underset{u}{\text{minimize}} \;\; \frac{1}{2}\tilde{x}_T^\mathsf{T} Q_T \tilde{x}_T + q_T^\mathsf{T} \tilde{x}_T + \sum_{t=0}^{T-1} \left(\frac{1}{2}\tilde{x}_t^\mathsf{T} Q_t \tilde{x}_t + \frac{1}{2}\tilde{u}_t^\mathsf{T} R_t \tilde{u}_t + \tilde{x}_t^\mathsf{T} S_t \tilde{u}_t + q_t^\mathsf{T} \tilde{x}_t + r_t^\mathsf{T} \tilde{u}_t \right) \\ &\text{subject to} \;\; \tilde{x}_{t+1} = A_t \tilde{x}_t + B_t \tilde{u}_t, \;\; \forall t \in \{0, 1, \ldots, T-1\} \end{aligned}$$

via dynamic programming for feedback gains $\{K_t\}_{t=0}^{T-1}$ and offsets $\{k_t\}_{t=0}^{T-1}$.

Then we update the nominal control trajectory via $\bar{u}_t \leftarrow \bar{u}_t + \tilde{u}_t$, where $\tilde{u}_t = K_t \tilde{x}_t + k_t$.

We repeat this whole process iteratively until convergence (e.g., change in $\tilde{u}$ or $J(\bar{x}, \bar{u})$ between iterations is small).

Differential dynamic programming (DDP)

The *exact* Bellman equation for our problem is

$$J_t^*(x_t) = \min_{u_t} \big(\ell(t, x_t, u_t) + J_{t+1}^*(f(t, x_t, u_t))\big)$$

In iLQR, we approximate the cost and dynamics as quadratic and affine, respectively. The right-hand-side is then approximately quadratic, so we can minimize it to find $\tilde{u}_t^*$.

In *differential dynamic programming (DDP)*, we set $J_t^*(x_t) = \frac{1}{2} x_t^\mathsf{T} P_t x_t + p_t^\mathsf{T} x_t + \beta_t$ and approximate the right-hand-side of the Bellman equation by quadratizing it directly.

Minimizing this approximation recursively for $\tilde{u}_t^*$ is equivalent to iLQR, except the 2nd-order terms are now

$$\begin{aligned}
H_{xx,t} &\coloneqq Q_t + A_t^\mathsf{T} P_{t+1} A_t + \textstyle\sum_{i=1}^n p_{t+1,i} \nabla_{xx}^2 f_i(t, \bar{x}_t, \bar{u}_t) \\
H_{uu,t} &\coloneqq R_t + B_t^\mathsf{T} P_{t+1} B_t + \textstyle\sum_{i=1}^n p_{t+1,i} \nabla_{uu}^2 f_i(t, \bar{x}_t, \bar{u}_t) \\
H_{xu,t} &\coloneqq S_t + A_t^\mathsf{T} P_{t+1} B_t + \textstyle\sum_{i=1}^n p_{t+1,i} \nabla_{xu}^2 f_i(t, \bar{x}_t, \bar{u}_t)
\end{aligned}$$

Overall, DDP estimates the Bellman equation more accurately than iLQR, but requires computing 2nd-order derivatives of the dynamics. Practically, iLQR is usually sufficient.

iLQR and DDP

Input: initial state $x_0 \in \mathbb{R}^n$, convergence tolerance $\varepsilon > 0$, maximum iterations $N \in \mathbb{N}_{>0}$

initialize nominal control sequence $\bar{u} = \{\bar{u}_t\}_{t=0}^{T-1}$, initial cost change $\tilde{J} = \infty$.

Rollout $\bar{x}_{t+1} = f(t, \bar{x}_t, \bar{u}_t)$ to get $\bar{x} = \{\bar{x}_t\}_{t=0}^{T}$ and $J(\bar{x}, \bar{u})$.

for $i = 1, 2, \ldots, N$

Backward pass:

Compute the approximating terms $\{\eta_t, h_{x,t}, h_{u,t}, H_{xx,t}, H_{uu,t}, H_{xu,t}\}_{t=0}^{T-1}$.
Recursively compute $\{\beta_t, p_t, P_t\}_{t=0}^{T}$ and $\{k_t, K_t\}_{t=0}^{T-1}$.

Forward pass:

Rollout $\tilde{x}_{t+1} = f(t, \bar{x}_t + \tilde{x}_t, \bar{u}_t + \tilde{u}_t) - \bar{x}_{t+1}$ with $\tilde{u}_t = k_t + K_t \tilde{x}_t$.
Update $(\bar{x}, \bar{u}) \leftarrow (\bar{x} + \tilde{x}, \bar{u} + \tilde{u})$ and $\tilde{J} \leftarrow J(\bar{x} + \tilde{x}, \bar{u} + \tilde{u}) - J(\bar{x}, \bar{u})$.

if $\|\tilde{u}\|_\infty < \varepsilon$ and/or $|\tilde{J}| < \varepsilon$
break

return $\bar{x}$, $\bar{u}$, and $\{k_t, K_t\}_{t=0}^{T-1}$.

The output is an open-loop trajectory $(\bar{x}, \bar{u})$ that is locally optimal for the OCP, and a policy $\pi(t, x, \bar{x}, \bar{u}) = \bar{u} + k_t + K_t(x - \bar{x})$ that is locally optimal for closed-loop tracking.

Algorithmic details

Both iLQR and DDP produce an open-loop trajectory $(\bar{x}, \bar{u})$ that is *locally optimal* for the OCP, and a policy $\pi(t, x, \bar{x}, \bar{u}) = \bar{u} + k_t + K_t(x - \bar{x})$ that is *locally optimal* for closed-loop tracking.

Since these methods are local optimization techniques, they can get stuck in local minima or even diverge. A "good" initialization is often critical.

The second-order terms $H_{xx,t}$ and $H_{uu,t}$ may not be positive-semidefinite and positive-definite, respectively. We can try regularizing them (i.e., $H_{xx,t} + \mu I$ and $H_{uu,t} + \mu I$) or projecting them.

The termination criteria is a design choice. For example, we can stop when either the change in control trajectory is "small", or when the cost improvement is "small".

During the forward pass, we need to make sure the new trajectory does not stray too far from the linearization in the previous iteration. We could penalize deviations more heavily, or do a line search on the policy rollout.

A great collection of tips with mathematical details can be found in (Tassa, 2011, §2.2.3).

Next class

The Hamilton-Jacobi-Bellman (HJB) equation
(i.e., dynamic programming in continuous-time)

References

Y. Tassa. *Theory and Implementation of Biomimetic Motor Controllers*. PhD thesis, The Hebrew University of Jerusalem, 2011.