Iterative LQR

Consider the following discrete finite-horizon cost function.

$$J(X,U) = l_f(x_N) + \sum_{k=0}^{N-1} l(x_k, u_k)$$
s.t. $x_{k+1} = f_k(x_k, u_k)$

Backward Pass Algorithm

Shorten the trajectory to just the last timestep. Then:
$V_N(x_N) = l_f(x_N)$

Now, consider the short trajectory with the final timestep and the one prior.
$V_{N-1} = \min_{u_{N-1}}\{l(x_{N-1}, u_{N-1}) + V_N(x_N)\}$
$V_{N-1} = \min_{u_{N-1}}\{l(x_{N-1}, u_{N-1}) + V_N(f(x_{N-1}, u_{N-1}))\}$
… since x_N is now dependent on choice of u_{N-1}.

We can generalize this into:
$V_k(x_k) = \min_{u_k}\{l(x_k, u_k) + V_{k+1}(f(x_k, u_k))\}$
Why do we just optimize over u_k? This is because V_{k+1} is already optimal, and the overall trajectory remains optimal by Bellman's Optimality Principle.

$V_k(x_k) = \min_{u_k} Q_k(x_k, u_k)$

Linearization

Consider the linearized dynamics of $x_{k+1} = f_k(x_k, u_k)$, namely:
$\delta x_{k+1} \approx A_k \delta x_k + B_k \delta u_k$

Stage Cost

Let us linearize about the stage cost:

$$\delta l(x_k, u_k) \approx \frac{\partial l}{\partial x_k}\delta x_k + \frac{\partial l}{\partial u_k}\delta u_k + \frac{1}{2}\begin{bmatrix}\delta x_k \\ \delta u_k\end{bmatrix}^\top \begin{bmatrix}\frac{\partial^2 l}{\partial x_k^2} & \frac{\partial^2 l}{\partial x_k \partial u_k} \\ \frac{\partial^2 l}{\partial u_k \partial x_k} & \frac{\partial^2 l}{\partial u_k^2}\end{bmatrix}\begin{bmatrix}\delta x_k \\ \delta u_k\end{bmatrix}$$

Future Cost

Let us linearize about $V_{k+1}(f(x_k, u_k))$:

$$\delta V_{k+1}(f(x_k, u_k)) \approx \left.\frac{\partial V_{k+1}}{\partial x}\right|_{x_{k+1}}^\top \delta x_{k+1} + \frac{1}{2}\delta x_{k+1}^\top \left.\frac{\partial^2 V_{k+1}}{\partial x^2}\right|_{x_{k+1}} \delta x_{k+1}$$

We define $p_{k+1} = \left.\frac{\partial V_{k+1}}{\partial x}\right|_{x_{k+1}}$ and $P_{k+1} = \left.\frac{\partial^2 V_{k+1}}{\partial x^2}\right|_{x_{k+1}}$

$$\delta V_{k+1}(f(x_k, u_k)) \approx p_{k+1}\delta x_{k+1} + \frac{1}{2}\delta x_{k+1}^\top P_{k+1}\delta x_{k+1}$$

Finally, we realize that we are not looking for an expression in terms of δ_{x+1}, but rather in terms of δ_x and δ_u. Therefore, we use $\delta x_{k+1} \approx A_k \delta x_k + B_k \delta u_k$ to write:

$$\delta V_{k+1}(f(x_k, u_k)) \approx p_{k+1}[A_k\delta x_k + B_k\delta u_k] + \frac{1}{2}[A_k\delta x_k + B_k\delta u_k]^\top P_{k+1}[A_k\delta x_k + B_k\delta u_k]$$

$$\delta V_{k+1}(f(x_k, u_k)) \approx \begin{bmatrix}A_k^\top p_{k+1} \\ B_k^\top p_{k+1}\end{bmatrix}^\top \begin{bmatrix}\delta x_k \\ \delta u_k\end{bmatrix} + \frac{1}{2}\begin{bmatrix}\delta x_k \\ \delta u_k\end{bmatrix}^\top \begin{bmatrix}A_k^\top P_{k+1}A_k & A_k^\top P_{k+1}B_k \\ B_k^\top P_{k+1}A_k & B_k^\top P_{k+1}B_k\end{bmatrix}\begin{bmatrix}\delta x_k \\ \delta u_k\end{bmatrix}$$

Action-Value Function

Putting it together, we get:

$$\delta Q_k = \frac{1}{2}\begin{bmatrix}\delta x_k \\ \delta u_k\end{bmatrix}^\top \begin{bmatrix}\frac{\partial^2 l}{\partial x_k^2} + A_k^\top P_{k+1}A_k & \frac{\partial^2 l}{\partial x_k \partial u_k} + A_k^\top P_{k+1}B_k \\ \frac{\partial^2 l}{\partial u_k \partial x_k} + B_k^\top P_{k+1}A_k & \frac{\partial^2 l}{\partial u_k^2} + B_k^\top P_{k+1}B_k\end{bmatrix}\begin{bmatrix}\delta x_k \\ \delta u_k\end{bmatrix} + \begin{bmatrix}\frac{\partial l}{\partial x_k} + A_k^\top p_{k+1} \\ \frac{\partial l}{\partial u_k} + B_k^\top p_{k+1}\end{bmatrix}^\top \begin{bmatrix}\delta x_k \\ \delta u_k\end{bmatrix}$$

$$\delta Q_k = \frac{1}{2}\begin{bmatrix}\delta x_k \\ \delta u_k\end{bmatrix}^\top \begin{bmatrix}Q_{xx} & Q_{xu} \\ Q_{ux} & Q_{uu}\end{bmatrix}\begin{bmatrix}\delta x_k \\ \delta u_k\end{bmatrix} + \begin{bmatrix}Q_x \\ Q_u\end{bmatrix}^\top \begin{bmatrix}\delta x_k \\ \delta u_k\end{bmatrix}$$

Finding Minimum

$$\frac{\partial \delta Q_k}{\partial \delta u_k} = Q_u + Q_{ux}\delta x + Q_{uu}\delta u = 0$$
$\delta u_k = -Q_{uu}^{-1}(Q_{ux}\delta x_k + Q_u)$
Adding regularization and rewriting as feedback law:
$\delta u_k = -(Q_{uu} + \rho I)^{-1}(Q_{ux}\delta x_k + Q_u)$
$\delta u_k = K_k \delta x_k + d_k$

We found the optimal control for the **local quadratic model**. Plugging it into δQ_k:

$$\delta Q_k = \frac{1}{2}\begin{bmatrix}\delta x_k \\ K_k\delta x_k + d_k\end{bmatrix}^\top \begin{bmatrix}Q_{xx} & Q_{xu} \\ Q_{ux} & Q_{uu}\end{bmatrix}\begin{bmatrix}\delta x_k \\ K_k\delta x_k + d_k\end{bmatrix} + \begin{bmatrix}Q_x \\ Q_u\end{bmatrix}^\top \begin{bmatrix}\delta x_k \\ K_k\delta x_k + d_k\end{bmatrix}$$

$$\delta Q_k = \frac{1}{2}\left[\delta x_k^\top Q_{xx}\delta x_k + \delta x_k^\top Q_{xu}(K_k\delta x_k + d_k) + (K_k\delta x_k + d_k)^\top Q_{ux}\delta x_k + (K_k\delta x_k + d_k)^\top Q_{uu}(K_k\delta x_k + d_k)\right] + Q_x^\top \delta x_k + Q_u^\top(K_k\delta x_k + d_k)$$

$$\delta Q_k = Q_u^\top d_k + \frac{1}{2}d_k^\top Q_{uu}d_k + Q_x^\top\delta x_k + Q_u^\top(K_k\delta x_k) + \frac{1}{2}\left[\delta x_k^\top Q_{xu}d_k + d_k^\top Q_{ux}\delta x_k + (K_k\delta x_k)^\top Q_{uu}d_k + d_k^\top Q_{uu}(K_k\delta x_k)\right] + \frac{1}{2}\left[\delta x_k^\top Q_{xx}\delta x_k + \delta x_k^\top Q_{xu}(K_k\delta x_k) + (K_k\delta x_k)^\top Q_{ux}\delta x_k + (K_k\delta x_k)^\top Q_{uu}(K_k\delta x_k)\right]$$

Two astute observations. First of all, since we solved for $\delta V_k = \min_{\delta u_k}\delta Q_k$, we effectively have:

$$\delta V_k = Q_u^\top d_k + \frac{1}{2}d_k^\top Q_{uu}d_k + Q_x^\top\delta x_k + Q_u^\top(K_k\delta x_k) + d_k^\top Q_{uu}K_k\delta x_k + d_k^\top Q_{ux}\delta x_k + \frac{1}{2}\left[\delta x_k^\top Q_{xx}\delta x_k + \delta x_k^\top Q_{xu}(K_k\delta x_k) + (K_k\delta x_k)^\top Q_{ux}\delta x_k + (K_k\delta x_k)^\top Q_{uu}(K_k\delta x_k)\right]$$

Secondly, we notice that this has a very similar form to the quadratic we defined: $\delta V_{k+1} = p_{k+1}^\top\delta x_{k+1} + \frac{1}{2}\delta x_{k+1}^\top P_{k+1}\delta x_{k+1}$, except that this is for timestep k

INJUNCTION: Recall that we solving a combined optimization problem for both x and u. What we found is a solution that depends on both x and u, specifically via a feedback control law. The feed-forward term of this feedback control law manifests itself in the cost as a constant $\Delta V_k = Q_u^\top d_k + \frac{1}{2}d_k^\top Q_{uu}d_k$

Finally we have:

$$\delta V_k = \Delta V_k + p_k^\top\delta x_k + \frac{1}{2}\delta x_k^\top P_k\delta x_k$$
$$\Delta V_k = Q_u^\top d_k + \frac{1}{2}d_k^\top Q_{uu}d_k$$
$$p_k = Q_x + K_k^\top Q_{uu}d_k + K_k^\top Q_u + Q_{xu}d_k$$
$$P_k = Q_{xx} + K_k^\top Q_{uu}K_k + K_k^\top Q_{ux} + Q_{xu}K_k$$

Iterative LQR

Repeatedly update trajectories.

Iterative LQR

Start with a guess for X, U (random noise, Bdot)

Evaluate $J = l_f(x_N) + \sum_{k=0}^{N-1} l(x_k, u_k)$

while $|J - J_{prev}| > tolerance$

$J_{prev} = J$

$K, d, \Delta V \leftarrow \boldsymbol{BackwardPass}(X, U)$

$X, U, J \leftarrow \boldsymbol{ForwardPass}(X, U, K, d, \Delta V, J_{prev})$

$return\ X, U, J$

Backward Pass

$p_N = \frac{\partial l_f}{\partial x}, P_N = \frac{\partial^2 l_f}{\partial x^2}$

for $k = N - 1: -1: 0$ (Bellman's Optimality Principle)

$$\delta Q_k = \frac{1}{2}\begin{bmatrix}\delta x_k \\ \delta u_k\end{bmatrix}^\top \begin{bmatrix} \frac{\partial^2 l}{\partial x_k^2} + A_k^\top P_{k+1} A_k & \frac{\partial^2 l}{\partial x_k \partial u_k} + A_k^\top P_{k+1} B_k \\ \frac{\partial^2 l}{\partial u_k \partial x_k} + B_k^\top P_{k+1} A_k & \frac{\partial^2 l}{\partial u_k^2} + B_k^\top P_{k+1} B_k \end{bmatrix} \begin{bmatrix}\delta x_k \\ \delta u_k\end{bmatrix} + \begin{bmatrix} \frac{\partial l}{\partial x_k} + A_k^\top p_{k+1} \\ \frac{\partial l}{\partial u_k} + B_k^\top p_{k+1} \end{bmatrix}^\top \begin{bmatrix}\delta x_k \\ \delta u_k\end{bmatrix}$$

if Q_{uu} invertible:

$\delta u_k = -(Q_{uu} + \rho I)^{-1}(Q_{ux}\delta x_k + Q_u)$

else:

$\rho \uparrow$ and try again

return sequence of $K, d, \Delta V$ for every timestep

Forward Pass

$\alpha = 1$

$for\ k = 0: 1: N - 1$

$\bar{u}_k = u_k + K_k(\bar{x}_k - x_k) + \alpha d_k$

$\bar{x}_{k+1} = f(\bar{x}_k, \bar{u}_k)$

$$J = l_f(x_N) + \sum_{k=0}^{N-1} l(\bar{x}_k, \bar{u}_k)$$

If J has actually decreased (line search condition), return. Else, decrease α and try again