iLQR Basics

① model based method, system dynamics known.

locally linearized model : $x_{t+1} = f(x_t, u_t) = \underset{\Delta}{A} x_t + \underset{\Delta}{B} u_t$

known

② iLQR works for non-linear system

③ Objective defined : $\min_{u_1 \cdots u_T} \sum_{t=1}^{T} c(x_t, u_t)$, with constraint: $x_t = f(x_{t-1}, u_{t-1})$

known

$\tau = \{x_1, u_1, x_2, u_2, \cdots x_T, u_T\}$

eg: minimize $\sum_{t=2}^{T} \| y_t - y_t^d \|^2 + \rho \sum_{t=1}^{T-1} \| u_t \|^2$,

w/ $x_{t+1} = A x_t + B u_t$. $y_t = C x_t$

④ iLQR use quadratic approximation of objective function, while normal shooting method use 1st order approximation.

⑤ Applications :

For Trajectory Optimization, Set cost $J = \sum_i u_i^T R u_i + x_f^T Q_f x_f$

(think of monkey)

For Trajectory Tracking , Set cost $J = \sum_i (x - x^d)^T Q (x - x^d) + \sum u^T R u + x_f^T Q_f x_f$

iLQR Steps

① Init start state x_0 & Init inperfect control $U=\{u_{t_0} \cdots u_{t_{N-1}}\}$

② Forward Pass, Simulate using (x_0, U), get (X, U) & a lot of partial derivative

③ Backward Pass, evaluate the value function @ each (X, u)

④ Update Control $\hat{U}$, evaluate the cost using $(x_0, \hat{U})$

Use Levenberg-Marguardt heuristic to adjust update rate.

(done) if cost converge, done

(accept) if cost smaller, $U = \hat{U}$, update more aggresive in ②

(reject) if cost larger, update more modest in ③

Thinking: Why we do Back Pass to evaluate Value function?

Pontryagin's Minimum Principle.

Problem Formulation

iLQR defines:

Total cost function $J(x_0, U) = \sum_{t=0}^{N-1} l(x_t, u_t) + l_f(x_N)$, l is immediate cost, l_f is final cost

cost to go $J_t(x, U_t) = \sum_{i=t}^{N-1} l(x_i, u_i) + l_f(x_N)$

value function $V_t = \min_{U_t} J_t(x, U_t) = \min_{u} [l(x,u) + V(f(x,u))]$

$V(x_N) = l_f(x_N)$, Note: V is local optimal cost-to-go !!!

In Forward pass, we need partial derivative of f, l, l_f, w.r.t. x_t, u_t

$f_x, f_u, f_{xx}, f_{xu}, f_{uu}, l_x, l_u, l_{xx}, l_{xu}, l_{uu}$ at each time step

we could get this by finite differential, or through direct derivative

In Backward

Perturbated Value function:

$$Q(\delta x, \delta u) = V(x+\delta x, u+\delta u) - V(x, u)$$

$$= l(x+\delta x, u+\delta u) + \underbrace{V(f(x+\delta x, u+\delta u))}_{\triangleq V'(x)} - l(x,u) - V(f(x,u))$$

Value of next state

Derivatives are computed to get the second-order expansion !

$$\begin{cases} Q_x = \dfrac{\partial Q}{\partial x} = l_x + f_x^T \cdot V_x' \\ Q_u = \dfrac{\partial Q}{\partial u} = l_u + f_u^T \cdot V_x' \\ Q_{xx} = \dfrac{\partial^2 Q}{\partial x^2} = l_{xx} + f_x^T \cdot V_{xx}' f_x + V_x' \cdot f_{xx} \\ Q_{ux} = \dfrac{\partial^2 Q}{\partial x \partial u} = l_{ux} + f_u^T \cdot V_{xx}' f_x + V_x' \cdot f_{ux} \\ Q_{uu} = \dfrac{\partial^2 Q}{\partial u^2} = l_{uu} + f_u^T \cdot V_{xx}' f_u + V_x' \cdot f_{uu} \end{cases}$$

V_{xx}' is second-derivative of next step Value, comes from backward propagate, explained next page with second order expansion of Q. we could compute optimal modification to control δu^*

$$\delta u^*(\delta x) = \arg\min_{\delta u} Q(\delta x, \delta u) = k + K \cdot \delta x \quad , \quad k = -Q_{uu}^{-1} \cdot Q_u$$

(see next page)

$$K = -Q_{uu}^{-1} \cdot Q_{ux}$$

$\because \begin{cases} V_x = Q_x - K^T \cdot Q_{uu} \cdot k \\ V_{xx} = Q_{xx} - K^T \cdot Q_{uu} \cdot K \end{cases}$ from Todorov (see next page)

$\therefore V'_{xx} \to Q_{uu} \to V_{xx}$, we could do backward pass to get V_{xx} from V'_{xx}

$V'_x \to Q_x \to V_x$

$\therefore \quad \delta u^* \begin{cases} \delta x \checkmark \\ k, K \end{cases} \begin{cases} Q_{uu}, Q_{ux} \\ Q_u \end{cases} \begin{cases} l_u, f_u, l_{uu} \checkmark \\ V'_x, V'_{xx} \end{cases} \begin{cases} Q'_x \; Q'_{xx} \checkmark \\ K', k' \checkmark \end{cases}$ } known from last step

$\therefore$ with another forward pass, we have new traj $(\hat{X}, \hat{U})$

$$\begin{cases} \delta u^*(\delta x) = k + K \delta x, \quad \delta x_0 = 0 \; (\hat{x}_0 = x_0) \\ \hat{u}_t = u_t + \delta u_t^* = u_t + \alpha \cdot k_t + K_t(\hat{x}_t - x_t) \\ \hat{x}_{t+1} = f(\hat{x}_t, \hat{u}_t) \end{cases}$$

states with updated controls

current rolled-out states

Line Search (Levenberg-Marquardt heuristic)

with new trajectory $(\hat{X}, \hat{U})$, we compute cost $\hat{J}$,

if $\hat{J} > J$, we are too aggressive because, so just change the λ during inversion.

Note: iLQR approximate objective function into quadratic & with no constraint it is a QP subproblem so it could update U in one jump. This is so-called SQP

Am I right ???

Why $\delta u^* = k + K \cdot \delta x$?

Problem: given δx, find $\delta u^* = \arg\min_{\delta u} Q(\delta x, \delta u)$

$\because Q(\delta x, \delta u) = V(x+\delta x, u+\delta u) - V(x, u)$

expand Q wrt δu

$$Q(\delta x, \delta u) = \underbrace{\cancel{Q(0,0)}}_{=0} + \frac{\partial Q}{\partial \delta x} \cdot \delta x + \frac{\partial Q}{\partial \delta u} \cdot \delta u + \frac{1}{2} \frac{\partial Q}{\partial \delta u^2} \cdot \delta u^2 + \frac{\partial Q}{\partial \delta x \partial \delta u} \cdot \delta u \cdot \delta x + \frac{1}{2} \frac{\partial Q}{\partial \delta x^2} \cdot \delta x^2 + \cdots$$

let $\dfrac{\partial Q(\delta x, \delta u)}{\partial \delta u} = 0$, $\Leftrightarrow$ δu is 2^{nd} order expansion optimal.

$$\Rightarrow Q_u + Q_{uu} \cdot \delta u^* + Q_{ux} \cdot \delta x = 0$$

$$\therefore \delta u^* = - Q_{uu}^{-1} \cdot Q_u - Q_{uu}^{-1} \cdot Q_{ux} \cdot \delta x$$

$$= k + K \cdot \delta x$$

Now we know $\delta u^* = k + K \cdot \delta x$, plug back to $Q(\delta x, \delta u)$

$\therefore Q(\delta x, \delta u^*)$

$$= Q_x \cdot \delta x + Q_u \cdot \delta u^* + \frac{1}{2} Q_{xx} \delta x^2 + Q_{xu} \delta x \delta u^* + \frac{1}{2} Q_{uu} \delta u^{*2}$$

$$= Q_x \delta x + \frac{1}{2} Q_{xx} \cdot \delta x^2 - \frac{1}{2} Q_{uu} \cdot \delta u^{*2}$$

$$= \frac{1}{2} k^T Q_{uu} k + Q_x \delta x - K^T Q_{uu} k \delta x + \frac{1}{2} Q_{xx} \delta x^2 - \frac{1}{2} K^T Q_{uu} K \delta x^2$$

$$= \Delta V + V_x \cdot \delta x + \frac{1}{2} V_{xx} \cdot \delta x^2$$

At curr iteration (x_0, U^{it}), at step i, the expected optimal cost-to-go.

$$\Rightarrow \begin{cases} \Delta V = -\frac{1}{2} k^T Q_{uu} k = -\frac{1}{2} Q_u^T \cdot Q_{uu}^{-1} Q_u \\ V_x = Q_x - k^T Q_{uu} k = Q_x - Q_u Q_{uu}^{-1} Q_{ux} \\ V_{xx} = Q_{xx} - K^T Q_{uu} K = Q_{xx} - Q_{xu} Q_{uu}^{-1} Q_{ux} \end{cases}$$

Regularization

$\delta u = k + K\delta x$ is similar to Newton's method of finding minimum.

∴ when Hessian Q_{uu} is singular, the step is ill-conditioned (Q_{uu}^{-1} is infinity)

∴ A regularization is applied to Q_{uu} :

$$\tilde{Q}_{uu} = Q_{uu} + \mu I_m \quad \text{-----} \quad ①$$

or

$$\left\{\begin{aligned} \tilde{V}'_{xx} &= V_{xx} + \mu I_n \\ \tilde{Q}_{uu} &= l_{uu} + f_u^T \tilde{V}'_{xx} f_u + V'_x \cdot f_{uu} \\ \tilde{Q}_{ux} &= l_{ux} + f_u^T \tilde{V}'_{xx} f_x + V'_x f_{ux} \end{aligned}\right\} \quad ②$$

Either way, δu will change and is no longer δu^*.

∴ $\hat{\delta u} = \hat{k} + \hat{K}\delta x = -\tilde{Q}_{uu}^{-1} Q_u - \tilde{Q}_{uu}^{-1}\cdot\tilde{Q}_{ux}\delta x$, plug into $Q(\delta x, \delta u)$

∴ $Q(\delta x, \hat{\delta u})$

$$= Q_x\cdot\delta x + Q_u\cdot\hat{\delta u} + \frac{1}{2}Q_{xx}\delta x^2 + Q_{xu}\delta x\,\hat{\delta u} + \frac{1}{2}Q_{uu}\hat{\delta u}^2$$

$$= Q_x\delta x + Q_u\hat{k} + Q_u\cdot\hat{K}\delta x + \frac{1}{2}Q_{xx}\delta x^2 + Q_{xu}\delta x\,\hat{k} + Q_{xu}\delta x\,\hat{K}\delta x + \frac{1}{2}\hat{k}^T Q_{uu}\hat{k} + \hat{K}^T Q_{uu}\hat{k}\,\delta x + \frac{1}{2}\hat{K}^T Q_{uu}\hat{K}\,\delta x^2$$

$$= \frac{1}{2}\hat{k}^T Q_{uu}\hat{k} + \hat{k}^T Q_u + \left[Q_x + \hat{K}^T Q_{uu}\hat{k} + \hat{K}^T\cdot Q_u + Q_{ux}^T\hat{k}\right]\delta x + \frac{1}{2}\left[Q_{xx} + \hat{K}^T Q_{uu}\hat{K} + Q_{xu}^T\hat{K} + \hat{K}\cdot Q_{xu}\right]\delta x^2$$

$$= \Delta V + V_x\cdot\delta x + \frac{1}{2}\cdot V_{xx}\cdot\delta x^2$$

$$\Rightarrow \left\{\begin{aligned} \Delta V &= \frac{1}{2}\hat{k}^T Q_{uu}\hat{k} + \hat{k}^T Q_u \\ V_x &= Q_x + \hat{K}^T Q_{uu}\hat{k} + \hat{K}^T\cdot Q_u + Q_{ux}^T\hat{k} \\ V_{xx} &= Q_{xx} + \hat{K}^T Q_{uu}\hat{K} + \hat{K}\cdot Q_{xu} + Q_{xu}^T\hat{K} \end{aligned}\right.$$