LQR full version

Shaohui Yang

April 8, 2021

1

Following the same notation as in HPIPM document, the cost function of optimal control function is:

$$\sum_{k=0}^{N-1} \frac{1}{2} \begin{bmatrix} \Delta u_k \\ \Delta x_k \\ 1 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} R_k & P_k & r_k \\ P_k^{\mathrm{T}} & Q_k & q_k \\ r_k^{\mathrm{T}} & q_k^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \Delta u_k \\ \Delta x_k \\ 1 \end{bmatrix} + \frac{1}{2} \Delta x_N^{\mathrm{T}} Q_N \Delta x_N + \Delta x_N^{\mathrm{T}} q_N \tag{1}$$

which is equivalent to

$$\sum_{k=0}^{N-1} \frac{1}{2} \begin{bmatrix} \Delta u_k \\ \Delta x_k \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} R_k & P_k \\ P_k^{\mathrm{T}} & Q_k \end{bmatrix} \begin{bmatrix} \Delta u_k \\ \Delta x_k \end{bmatrix} + \begin{bmatrix} \Delta u_k \\ \Delta x_k \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} r_k \\ q_k \end{bmatrix} + \frac{1}{2} \Delta x_N^{\mathrm{T}} Q_N \Delta x_N + \Delta x_N^{\mathrm{T}} q_N \tag{2}$$

The dynamic constraints are

$$x_{k+1} = f(x_k, u_k) \tag{3a}$$

$$\Delta x_{k+1} = A_k \Delta x_k + B_k \Delta u_k + b_k \tag{3b}$$

$$A_k = \frac{\partial f}{\partial x_k}(x_k, u_k) \quad B_k = \frac{\partial f}{\partial u_k}(x_k, u_k) \quad b_k = f(x_k, u_k) - x_{k+1} \tag{3c}$$

Let's assume that the cost-to-go starting from stage $k+1$ is

$$V_{k+1}(z) = \frac{1}{2} z^{\mathrm{T}} S_{k+1} z + z^{\mathrm{T}} s_{k+1} + c_{k+1} \tag{4}$$

Then the cost-to-go starting from stage k is

$$V_k(z) = \frac{1}{2} z^{\mathrm{T}} Q_k z + z^{\mathrm{T}} q_k + \min_w \Big(\frac{1}{2} w^{\mathrm{T}} P_k z + \frac{1}{2} z^{\mathrm{T}} P_k^{\mathrm{T}} w + \frac{1}{2} w^{\mathrm{T}} R_k w + w^{\mathrm{T}} r_k + V_{k+1}(A_k z + B_k w + b_k) \Big) \tag{5}$$

In particular, the last term is

$$\begin{aligned} V_{k+1}(A_k z + B_k w + b_k) &= \frac{1}{2} (A_k z + B_k w + b_k)^{\mathrm{T}} S_{k+1} (A_k z + B_k w + b_k) + (A_k z + B_k w + b_k)^{\mathrm{T}} s_{k+1} + c_{k+1} \qquad (6) \\ &= \frac{1}{2} w^{\mathrm{T}} B_k^{\mathrm{T}} S_{k+1} B_k w + w^{\mathrm{T}} (B_k^{\mathrm{T}} S_{k+1} A_k z + B_k^{\mathrm{T}} S_{k+1} b_k + B_k^{\mathrm{T}} s_{k+1}) + TEMP \\ TEMP &= \frac{1}{2} z^{\mathrm{T}} A_k^{\mathrm{T}} S_{k+1} A_k z + z^{\mathrm{T}} (A_k^{\mathrm{T}} S_{k+1} b_k + A_k^{\mathrm{T}} s_{k+1}) + \frac{1}{2} b_k^{\mathrm{T}} S_{k+1} b_k + b_k^{\mathrm{T}} s_{k+1} + c_{k+1} \qquad (7) \end{aligned}$$

Let's focus on the second part here:

$$\begin{aligned} &\min_w \quad \frac{1}{2} w^{\mathrm{T}} P_k z + \frac{1}{2} z^{\mathrm{T}} P_k^{\mathrm{T}} w + \frac{1}{2} w^{\mathrm{T}} R_k w + w^{\mathrm{T}} r_k + V_{k+1}(A_k z + B_k w + b_k) \\ \to &\min_w \quad \frac{1}{2} w^{\mathrm{T}} (R_k + B_k^{\mathrm{T}} S_{k+1} B_k) w + w^{\mathrm{T}} (P_k z + B_k^{\mathrm{T}} S_{k+1} A_k z + r_k + B_k^{\mathrm{T}} S_{k+1} b_k + B_k^{\mathrm{T}} s_{k+1}) + TEMP \end{aligned} \tag{8}$$

Recall the following unconstrained optimization

$$\begin{aligned} &\min_x \quad \frac{1}{2}x^{\mathrm{T}}Qx + x^{\mathrm{T}}q \\ &\; x^* = -Q^{-1}q \\ &\min_x \quad \frac{1}{2}x^{\mathrm{T}}Qx + x^{\mathrm{T}}q = -\frac{1}{2}q^{\mathrm{T}}Q^{-1}q \end{aligned} \tag{9}$$

So the optimum w^* is

$$\begin{aligned} w^* &= -(R_k + B_k^{\mathrm{T}}S_{k+1}B_k)^{-1}(P_k + B_k^{\mathrm{T}}S_{k+1}A_k)z - (R_k + B_k^{\mathrm{T}}S_{k+1}B_k)^{-1}(r_k + B_k^{\mathrm{T}}S_{k+1}b_k + B_k^{\mathrm{T}}s_{k+1}) && (10) \\ &= K_k z + k_k \\ K_k &= -(R_k + B_k^{\mathrm{T}}S_{k+1}B_k)^{-1}(P_k + B_k^{\mathrm{T}}S_{k+1}A_k) && (11) \\ k_k &= -(R_k + B_k^{\mathrm{T}}S_{k+1}B_k)^{-1}(r_k + B_k^{\mathrm{T}}S_{k+1}b_k + B_k^{\mathrm{T}}s_{k+1}) && (12) \end{aligned}$$

The term $\min\limits_w$ is

$$\begin{aligned} \min_w = &-\frac{1}{2}(P_k z + B_k^{\mathrm{T}}S_{k+1}A_k z + r_k + B_k^{\mathrm{T}}S_{k+1}b_k + B_k^{\mathrm{T}}s_{k+1})^{\mathrm{T}}(R_k + B_k^{\mathrm{T}}S_{k+1}B_k)^{-1}(P_k z + B_k^{\mathrm{T}}S_{k+1}A_k z + r_k + B_k^{\mathrm{T}}S_{k+1}b_k + B_k^{\mathrm{T}}s_{k+1}) \\ &+ TEMP \\ &= -\frac{1}{2}z^{\mathrm{T}}(P_k + B_k^{\mathrm{T}}S_{k+1}A_k)^{\mathrm{T}}(R_k + B_k^{\mathrm{T}}S_{k+1}B_k)^{-1}(P_k + B_k^{\mathrm{T}}S_{k+1}A_k)z \\ &- z^{\mathrm{T}}(P_k + B_k^{\mathrm{T}}S_{k+1}A_k)^{\mathrm{T}}(R_k + B_k^{\mathrm{T}}S_{k+1}B_k)^{-1}(r_k + B_k^{\mathrm{T}}S_{k+1}b_k + B_k^{\mathrm{T}}s_{k+1}) \\ &- \frac{1}{2}(r_k + B_k^{\mathrm{T}}S_{k+1}b_k + B_k^{\mathrm{T}}s_{k+1})^{\mathrm{T}}(R_k + B_k^{\mathrm{T}}S_{k+1}B_k)^{-1}(r_k + B_k^{\mathrm{T}}S_{k+1}b_k + B_k^{\mathrm{T}}s_{k+1}) + TEMP \end{aligned} \tag{13}$$

Now go back to $V_k(z)$

$$\begin{aligned} V_k(z) &= \frac{1}{2}z^{\mathrm{T}}P_k z + z^{\mathrm{T}}p_k + c_k && (14) \\ S_k &= Q_k + A_k^{\mathrm{T}}S_{k+1}A_k - (P_k + B_k^{\mathrm{T}}S_{k+1}A_k)^{\mathrm{T}}(R_k + B_k^{\mathrm{T}}S_{k+1}B_k)^{-1}(P_k + B_k^{\mathrm{T}}S_{k+1}A_k) && (15) \\ s_k &= q_k + A_k^{\mathrm{T}}S_{k+1}b_k + A_k^{\mathrm{T}}s_{k+1} - (P_k + B_k^{\mathrm{T}}S_{k+1}A_k)^{\mathrm{T}}(R_k + B_k^{\mathrm{T}}S_{k+1}B_k)^{-1}(r_k + B_k^{\mathrm{T}}S_{k+1}b_k + B_k^{\mathrm{T}}s_{k+1}) && (16) \\ c_k &= \frac{1}{2}b_k^{\mathrm{T}}S_{k+1}b_k + b_k^{\mathrm{T}}s_{k+1} + c_{k+1} - \frac{1}{2}(r_k + B_k^{\mathrm{T}}S_{k+1}b_k + B_k^{\mathrm{T}}s_{k+1})^{\mathrm{T}}(R_k + B_k^{\mathrm{T}}S_{k+1}B_k)^{-1}(r_k + B_k^{\mathrm{T}}S_{k+1}b_k + B_k^{\mathrm{T}}s_{k+1}) && (17) \end{aligned}$$

2 Clarification on how feedback work for the delta form variables

Let's first define the notations clear.

$x^i(u^i)$ represents state(input) trajectory around which the linearzation of i-th iteration happens. $x^{i+1}(u^{i+1})$ represents state(input) trajectory that we hope our robot would follow in reality. We are looking for different ways to calculate the incremental parts between them. The results retrieved from hpipm, denoted as $\Delta x^{*i}, \Delta u^{*i}$, is optimal in the following problem setup:

$$\begin{aligned} \underset{\Delta u, \Delta x}{\text{minimize}} \quad & \sum_{k=0}^{N-1} \frac{1}{2} \begin{bmatrix} \Delta u_k \\ \Delta x_k \\ 1 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} R_k^i & P_k^i & r_k^i \\ (P_k^i)^{\mathrm{T}} & Q_k^i & q_k^i \\ (r_k^i)^{\mathrm{T}} & (q_k^i)^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \Delta u_k \\ \Delta x_k \\ 1 \end{bmatrix} + \frac{1}{2}\Delta x_N^{\mathrm{T}} Q_N^i \Delta x_N + \Delta x_N^{\mathrm{T}} q_N^i \\ \Delta x_{k+1} &= A_k^i \Delta x_k + B_k^i \Delta u_k + b_k^i \\ \Big(C_k^i \Delta x_k + D_k^i \Delta u_k + e_k^i &= 0\Big) \end{aligned} \tag{18}$$

$\Delta x^{*i}, \Delta u^{*i}$ has the following relationship in each stage

$$\Delta u_j^{*i} = K_j^i \Delta x_j^{*i} + k_j^i \quad \forall j \in \{0, 1, \ldots, N-1\} \tag{19}$$

To simplify the notations, we eliminate the stage subscript:

$$\Delta u^{*i} = K^i \Delta x^{*i} + k^i \tag{20}$$

An intuitive understanding of (18) is the following: if the system dynamics, cost function and constraints (may or may not exist) all remain linear as the Taylor first-order expansion around $x^i(u^i)$, $\Delta x^{*i}, \Delta u^{*i}$ will be optimal in the sense that $x^{i+1} - x^i = \Delta x^{*i}$, $u^{i+1} - u^i = \Delta u^{*i}$ and the new trajectory will be exactly $x^{i+1}(u^{i+1})$.

However, since everything is nonlinear, the update formula for $x^{i+1}(u^{i+1})$ will be different. Currently, with $\alpha \in [0, 1]$, the update formula is:

$$u^{i+1} = u^i + \alpha \Delta u^{*i} \tag{21}$$

$$x^{i+1} = x^i + \alpha \Delta x^{*i} \tag{22}$$

and as a result

$$u^{i+1} - u^i = K^i(x^{i+1} - x^i) + \alpha k^i \tag{23}$$

In reality, we would like to drive our state x^{real} to be as close to x^{i+1} as possible with input u^{real}, so it is quite reasonable if do the following by mimicing (23). We apply the input law (23) not for the planned state x^{i+1}, but on the real state x^{real}.

$$u^{real} - u^i = K^i(x^{real} - x^i) + \alpha k^i \tag{24}$$

$$u^{real} - u^i = K^i(x^{real} - x^i) + k^i \tag{25}$$

Question to Farbod, Ruben: which one do you think is right, (24) or (25)? I will go with αk^i because k^i only makes sense if we faithfully follow $\alpha = 1$. As for everything below, I will stick to my choice.

If we substract (23) and (24), we have

$$u^{real} - u^{i+1} = K^i(x^{real} - x^{i+1}) \tag{26}$$

$$u^{real} = u^{i+1} + K^i(x^{real} - x^{i+1}) \tag{27}$$

$$u^{real} = u^{i+1} - K^i x^{i+1} + K^i x^{real} \tag{28}$$

The linear controller of ocs2 has the convention of

$$u^{real} = u_{ff} + K_{fb} x^{real} \tag{29}$$

$$u_{ff} = u^{i+1} - K^i x^{i+1} \tag{30}$$

$$K_{fb} = K^i \tag{31}$$

Then we move on to the constrained case with projection method. By consistently notating things, the decision variables $\Delta x^{*i}, \Delta \tilde{u}^{*i}$ are retrieved from hpipm. The following holds naturally:

$$\Delta \tilde{u}^{*i} = \tilde{K}^i \Delta x^{*i} + \tilde{k}^i \tag{32}$$

Don't forget to include the projection of input

$$\Delta u^{*i} = P_u^i \Delta \tilde{u}^{*i} + P_x^i \Delta x^{*i} + P_e^i \tag{33}$$

$$= (P_u^i \tilde{K}^i + P_x^i)\Delta x^{*i} + P_u^i \tilde{k}^i + P_e^i \tag{34}$$

The update formula is still the same

$$u^{i+1} = u^i + \alpha \Delta u^{*i} \tag{35}$$

$$x^{i+1} = x^i + \alpha \Delta x^{*i} \tag{36}$$

So

$$u^{i+1} - u^i = (P_u^i \tilde{K}^i + P_x^i)(x^{i+1} - x^i) + \alpha(P_u^i \tilde{k}^i + P_e^i) \tag{37}$$

Same mimicing policy applies

$$u^{real} - u^i = (P_u^i \tilde{K}^i + P_x^i)(x^{real} - x^i) + \alpha(P_u^i \tilde{k}^i + P_e^i) \tag{38}$$

The difference between above two equations is

$$u^{real} - u^{i+1} = (P_u^i \tilde{K}^i + P_x^i)(x^{real} - x^{i+1}) \tag{39}$$

$$u^{real} = u^{i+1} - (P_u^i \tilde{K}^i + P_x^i)x^{i+1} + (P_u^i \tilde{K}^i + P_x^i)x^{real} \tag{40}$$

$$u_f f = u^{i+1} - (P_u^i \tilde{K}^i + P_x^i)x^{i+1} \tag{41}$$

$$K_{fb} = P_u^i \tilde{K}^i + P_x^i \tag{42}$$