Meeting 08/20/2020

Shuo Zhang

LQR

LQR Ext5: Trajectory Following for Non-Linear Systems

- A state sequence x_0^*, x_1^*, ..., x_H^* is a feasible target trajectory if and only if

$$\exists u_0^*, u_1^*, \ldots, u_{H-1}^* \;:\; \forall t \in \{0, 1, \ldots, H-1\} \;:\; x_{t+1}^* = f(x_t^*, u_t^*)$$

- Problem statement:

$$\min_{u_0, u_1, \ldots, u_{H-1}} \sum_{t=0}^{H-1} (x_t - x_t^*)^\top Q (x_t - x_t^*) + (u_t - u_t^*)^\top R (u_t - u_t^*)$$

$$\text{s.t.} \quad x_{t+1} = f(x_t, u_t)$$

- Transform into linear time varying case (LTV):

$$x_{t+1} \approx f(x_t^*, u_t^*) + \underbrace{\frac{\partial f}{\partial x}(x_t^*, u_t^*)}_{A_t}(x_t - x_t^*) + \underbrace{\frac{\partial f}{\partial u}(x_t^*, u_t^*)}_{B_t}(u_t - u_t^*)$$

$$x_{t+1} - x_{t+1}^* \approx A_t(x_t - x_t^*) + B_t(u_t - u_t^*)$$

Got Matrix A_t and B_t for all 4 tasks.

Is Q_t and R_t always Identity Matrix in our case?

LQR

LQR Ext5: Trajectory Following for Non-Linear Systems

- Transformed into linear time varying case (LTV):

$$\min_{u_0,u_1,\ldots,u_{H-1}} \sum_{t=0}^{H-1} (x_t - x_t^*)^\top Q(x_t - x_t^*) + (u_t - u_t^*)^\top R(u_t - u_t^*)$$

$$\text{s.t. } x_{t+1} - x_{t+1}^* = A_t(x_t - x_t^*) + B_t(u_t - u_t^*)$$

- Now we can run the standard LQR back-up iterations.

- Resulting policy at i time-steps from the end:

$$u_{H-i} - u_{H-i}^* = K_i(x_{H-i} - x_{H-i}^*)$$

- The target trajectory need not be feasible to apply this technique, however, if it is infeasible then the linearizations are not around the (state,input) pairs that will be visited

u_0 is calculated with K_H? Backward?

LQR

LQR Ext4: Linear Time Varying (LTV) Systems

$$
\begin{aligned}
x_{t+1} &= A_t x_t + B_t u_t \\
g(x_t, u_t) &= x_t^\top Q_t x_t + u_t^\top R_t u_t
\end{aligned}
$$

LQR Ext4: Linear Time Varying (LTV) Systems

Set $P_0 = 0$.
for $i = 1, 2, 3, \ldots$

$$
\begin{aligned}
K_i &= -(R_{H-i} + B_{H-i}^\top P_{i-1} B_{H-i})^{-1} B_{H-i}^\top P_{i-1} A_{H-i} \\
P_i &= Q_{H-i} + K_i^\top R_{H-i} K_i + (A_{H-i} + B_{H-i} K_i)^\top P_{i-1} (A_{H-i} + B_{H-i} K_i)
\end{aligned}
$$

The optimal policy for a i-step horizon is given by:

$$\pi(x) = K_i x$$

The cost-to-go function for a i-step horizon is given by:

$$J_i(x) = x^\top P_i x.$$

Next plans

After we get LQR solutions(Matrix K_i):

1) Try closed loop control using LQR(Matrix K_i) on real environments?
2) I can also try closed loop control using PPO (trained from model) on real environment?

3) Derive new equations, objective function and lambda-eta optimization procedures for our AIP?
4) Implement AIP

AIP Implementation (against TRPO,PPO)

Difference 1 Policy Network:

TRPO/PPO: u_final=pi_theta([x])

AIP: u_final=pi_theta([x, u_controller])

Difference 2 Constraint:

TRPO:

KL(pi_theta_new || pi_theta_old)<epsilon

PPO:

No constraint

Constraint (KL(pi_theta_new || pi_theta_old)<epsilon) combined into objective function

AIP:

KL(pi_theta_new || pi_theta_old)<epsilon

??KL(pi_theta || controller)<omega??

(Need to derive new equations and objective for optimization?)

Task	Open Loop A* +Rollout	Open Loop PPO + Rollout	Closed Loop PPO	Cloes Loop LQR based on A*	AIP (3 options)
Reacher (0.1% model)	Done + Done	Done + Done	Not yet	Not yet	Not yet
Gazebo Hand (0.1% model)	Done + Done	Done + Not yet	Not yet	Not yet	Not yet
Acrobot (100% model)	Done + Done	Done + Done	Not yet	Not yet	Not yet
Real Hand (100% model)	Done + Not yet	Done + Not yet	Not yet	Not yet	Not yet