Linear Quadratic Regulators and Iterative Learning

Zamboni Riccardo

e-Novia, Italy, ricc.zamboni@gmail.com

Keywords: LQR, Iterative Learning, Sparse Learning

Abstract

Linear Quadratic Regulators are well studied tools for control but they have been studied in a modern learning-oriented perspective just recently. Here is a summa of this kind of techniques and some notes about recent results, trying to fill the gap and show some possible directions.

1 Introduction

Linear quadratic regulator (LQR) is one of the most popular frameworks to tackle control problems in general and Markov Decision Processes (MDPs) in particular. The LQR case can be considered as the simplest scenario in the hard problems set of RL and it does admit elegant closed form formulae that are helpful for understanding many issues of this problem formulation. This peek is a tentative visit of the first step of the RL ladder.

2 LQR problem formulation

The problem framework here is MDPs in continuous state-space where $x_t \in \mathcal{X} \subset \mathcal{R}^n$, $u_t \in \mathcal{U} \in \mathcal{R}^m$, $w_t \in \mathcal{W} \in \mathcal{R}^N$, which are usually referred to as states, control actions and some random disturbance at time t, respectively. Further, a stage-cost function is defined as $g : \mathcal{X} \times \mathcal{U} \times \mathcal{X} \to \mathcal{R}$ together with a transition dynamic $f : \mathcal{X} \times \mathcal{U} \times \mathcal{W} \to \mathcal{X}$ for a control action u_t in the state x_t with some disturbance w_t, or often w_{t+1}, since this last term is fully revealed at time $t + 1$.

The goal is then to find an optimal (stationary) control policy $\pi : \mathcal{X} \to \mathcal{U}$. A definition of a "stationary distribution" can be found here in this post about [1], just to refresh some concepts. The optimal policy is expected to solve:

$$\begin{aligned} &\min \sum_{t=0}^{\infty} \gamma^t \mathbf{E}[g(x_t, u_t, x_{t+1})] \\ &s.t.\ x_{t+1} = f(x_t, u_t, w_t), x_0 \sim \mathcal{D} \end{aligned} \tag{1}$$

Infinite horizon LQR is then a subclass of the previous problem, where the dynamic is linear and the stage-cost is quadratic. This case is particularly nice due to some interesting noise-related properties, but Ben Recht does an extremely clear example talking about LQR as in [2]. In the best scenario without any disturbance (we will later discuss how the LQR behaves with some disturbances in the control or state axes), the system becomes

$$\begin{aligned} \min \mathbf{E}[&\sum_{t=0}^{\infty} x_t Q x_t^T + u_t R u_t^T] \\ s.t.\ &x_{t+1} = Ax_t + Bu_t, \\ &u_t = \pi(x_t),\ x_0 \sim \mathcal{D} \end{aligned} \tag{2}$$

where $A \in \mathcal{R}^{n \times n}$, $B \in \mathcal{R}^{n \times m}$, $Q \succeq 0$ and $R \succ 0$. In such case, the optimal control policy results to be a static state feedback and value function are known to be linear and convex quadratic on state respectively

$$\pi^\star(x) = Kx,\ V^\star(x) = x^T P x$$

where

$$\begin{aligned} P = A^T P A + Q - A^T P B (B^T P B + R)^{-1} B^T P A, \\ K = -(B^T P B + R)^{-1} B^T P A \end{aligned} \tag{3}$$

The first is known as the discrete time Algebraic Riccati Equation (DARE). In case of (A, B)-controllability and (Q, A)-detectability, the solution is unique and can be efficiently computed via the Riccati recursion or other alternatives.

In the continous time case

$$\dot{x}(t) = Ax(t) + Bu(t)$$

the same steps would lead to continuous algebraic Riccatti equation (CARE) for a matrix P

$$\begin{aligned} A^T P + PA - PBR^{-1}B^T P + Q = 0, \\ K = R^{-1} B^T P \end{aligned} \tag{4}$$

The LQR solution can be derived via adjoints method as well, which is now called back-propagation by the cool-kids, but I am no cool. Ben Recht did it as well talk about it but it is just too elegant not to include some short passages here. First, after defining the Lagrangian of the system

$$\mathcal{L}(x, u, p) = \sum_{t=0}^{N} [x_t^T Q x_t + u_t^T R u_t - p_{t+1} T (x_{t+1} - Ax_t - bu_t)] \tag{5}$$

We compute the gradients

$$\begin{aligned} \nabla_{x_t}\mathcal{L} &= Qx_t - p_t + A^T p_{t+1} \\ \nabla_{u_t}\mathcal{L} &= Ru_t + B^T p_{t+1} \\ \nabla_{p_t}\mathcal{L} &= -x_t + Ax_{t-1} + Bu_{t-1} \end{aligned} \tag{6}$$

We now need to find settings for all of the variables to make these gradients vanish. The simplest way to solve for the costates p_t and control actions u_t is to work backwards, leading to the (DARE) equation. How cool is this?

2.1 Iterative LQR

The linear dynamic can be manipulated to better interpret the role of a LQR control

$$Ax_t + Bu_t = F_t \begin{bmatrix} x_t \\ u_t \end{bmatrix} + f_t \tag{7}$$

where F_t, f_t represent the portions of the affine transformation of the dynamics. The stage-cost as well can be manipulated as

$$\begin{aligned} g(x_t, u_t) = \frac{1}{2} \begin{bmatrix} x_t \\ u_t \end{bmatrix}^T G_t \begin{bmatrix} x_t \\ u_t \end{bmatrix} + \begin{bmatrix} x_t \\ u_t \end{bmatrix}^T g_t, \\ G_t = \begin{bmatrix} G_{xx} & G_{xu} \\ G_{ux} & G_{uu} \end{bmatrix} \end{aligned} \tag{8}$$

it is then possible to solve for u_t only since it affects merely the last term

$$\begin{aligned} Q(x_t, u_t) &= const + g(x_t, u_t) \\ \nabla_{u_t} Q(x_t, u_t) &= G_{ux}x_t + G_{uu}u_t + g_u^T = 0 \end{aligned} \tag{9}$$

leading to

$$\begin{aligned} u_t &= -G_{uu}^{-1}(G_{ux}x_t + g_u) \\ u_t &= K_t x_t + k_t \end{aligned} \tag{10}$$

by eliminating u_t via sostitution we get

$$V(x_t) = Q(x_t, K_t x_t + k_t) = const + \frac{1}{2} x_t^T V_t x_t + v_t \tag{11}$$

it is then possible to go on with the substitution since

$$u_{t-1}$$

affects

$$x_t$$

through the dynamics and so on deriving a backward recursion after some trivial initialization:

for $t = T$ to 1:

1. $Q_t = G_t + F_t^T V_{t+1} F_t$
2. $q_t = c_t + F_t^T V_{t+1} f_t + F_t^T V^{t+1}$
3. $Q(x_t, u_t) = const + \frac{1}{2} \begin{bmatrix} x_t \\ u_t \end{bmatrix}^T Q_t \begin{bmatrix} x_t \\ u_t \end{bmatrix} + \begin{bmatrix} x_t \\ u_t \end{bmatrix}^T q_t$
4. $u_t \longleftarrow argmin_{u_t} Q(x_t, u_t) = K_t x_t + k_t$, $K_t = -Q_{uu}^{-1} Q_{ux}, k_t = -Q_{uu}^{-1} q_u$
5. $V_t = Q_{xx} + Q_{xu}K_t + K_t^T Q_{ux} + K_t^T Q_{uu} k_t$
6. $v_t = q_x + Q_{xu}k_t + K_t^T q_u + K_t^T Q_{uu} k_t$
7. $V(x_t) = const + \frac{1}{2} x_t^T V_t x_t + x_t^T v_t$

subsequently the forward recursion of the controlled system is performed

for $t = 1$ to T:

1. $u_t = K_t x_t + k_t$
2. $x_{t+1} = f(x_t, u_t)$

It is possible to generalize to stationary gaussian processes as well, as the one tackled by Kalman filtering, leading to the actual i-LQR formulation. In such cases

$$\begin{aligned} x_{t+1} &\sim p(x_{t+1}|x_t, u_t), \\ p(x_{t+1}|x_t, u_t) &= (N)(F_t \begin{bmatrix} x_t \\ u_t \end{bmatrix} + f_t,\ \Sigma) \end{aligned} \tag{12}$$

2.2 LQR with Structured Policy Iteration

It is possible to induce a specific structure on the policy as well, and this might be extremely useful while talking about sparse and decentralized feedback control. From the standard LQR formulation, a regularizer is added on the policy to induce such structure. The regularized LQR problem can be then stated in the discrete time case as

$$\begin{aligned} \min_K \mathbf{E}[\sum_{t=0}^{\infty} x_t Q x_t^T + u_t R u_t^T] + \lambda r(K) \\ \text{s.t. } x_{t+1} = Ax_t + Bu_t, \\ u_t = \pi(x_t),\ x_0 \sim \mathcal{D} \end{aligned} \tag{13}$$

for a nonnegative parameter $\lambda \geq 0$. The regularizer $r : \mathcal{R}^{n \times m} \to \mathcal{R}$ is a nonnegative convex regularizer inducing the structure of the policy K.

Different regularizers induce different types of structure on the policy K. It is possible to consider:

1. Lasso reg.: $r(K) = \|K\|_1 = \sum_{i,j} \|K_{i,j}\|$, which leads to a sparse gain matrix K.
2. Group lasso reg.: $r(K) = \|K\|_{\mathcal{G},2} = \sum_{g \in \mathcal{G}} \|K_g\|_2$, which induces a block-sparse gain matrix K.

3. Nuclear-form reg.: $r(K) = \|K\|_\star = \sum_i \sigma_i(K)$, being σ_i the $i-th$ largest singular value of K. This last induces a low-rank gain matrix K.

The same families of metrics can be extended to track reference policies with $K^{ref} \in \mathcal{R}^{n \times m}$.

The whole stage-cost function is then formulated as

$$F(K) := f(K) + \lambda r(K) \tag{14}$$

Here $f(K) = \mathbf{Tr}(\Sigma_0 P)$ where $\Sigma_0 = \mathbf{E}[x_0 x_0^T]$ is the covariance matrix of the initial state and P is the quadratic value matrix.

It is possible to solve the whole problem using the so called Structured Policy Iteration (S-PI) algorithm by iteratively evaluating the policy via the P^i and Σ^i matrices satisfying the Lyapunov equations:

$$\begin{aligned}(A + BK^i)^T P^i (A + BK^i) - P^i + Q + (K^i)^T R K^i = 0 \\ (A + BK^i)\Sigma^i (A + BK^i)^T - \Sigma^i + \Sigma_0 = 0\end{aligned} \tag{15}$$

and by defining a policy improvement accordingly using the gradient of K

$$\nabla_K f(K^i) = 2((R + B^T P^i B)K^i + B^T P^i A)\Sigma^i \tag{16}$$

and plugging it in a proximal gradient method adapted to the regularizer.

The major cost incurs when solving the Lyapunov equations in the policy (and covariance) evaluation step. Yet, a sequence of Lyapunov equations can be solved with less cost by using iterative methods with adopting the previous one (warm-start) or approximated one. For the whole algorithm structure and the model-free implementation via smoothing procedures you can check this recent article about it [3] .

References

[1] "Complex stuff with no bullshit." `https://jeremykun.com/2015/04/06/markov-chain-monte-carlo-without-all-the-bullshit/`.

[2] B. Recht, "The linear quadratic regulator." `https://www.argmin.net/2018/02/08/lqr/`.

[3] `https://proceedings.icml.cc/static/paper_files/icml/2020/3607-Paper.pdf`.