Stabilize Quadcopter Using LQR control

Chaojie Wang

08/15/2015

1 Introduction

Quadcopter needs a controller to control the state. LQR control is way to do that. In this review article, we will specify how LQR is used in controling and statbilize the state. For now, we only concern about the angles of the quadcopter, thus the state is comprised of angles and angular velocity.

2 LQR

LQR(Linear Quadratic Regulator) is specified by system equation with initial state x_0

$$\dot{x} = Ax + Bu \tag{1}$$

and a cost function

$$J = \int_0^\infty (x^T Q x + u^T R u)\, dt \tag{2}$$

Our goal is to choose input u that minimizes the cost function J. Doing the minimization, essentially we are saying that we want a best input that make x and u 0 in lowest cost specified by Q and R. The value of Q and R are desgin choices. Doing this minimization, one can show that [2]

$$u = \underbrace{-R^{-1} B^T P}_{K} x \tag{3}$$

Where P is a constant and can be determined by

$$0 = PA + A^T P - PBR^{-1}B^T P + Q$$

Note that we can use MATLAB to compute K.

3 LQR With Reference

LQR presented previously is only control the state and input to 0. But often, we want to specify the state the system. In the case of linear system, this is easy and only requires slightly motification of 3.

Let x_d the desire state, and u_d be the desire input. And we want the actual(x, u) be as close as possible to (x_d, u_d). We can derive the error term

$$\begin{aligned} \dot{x}_e &= \dot{x} - \dot{x}_d \\ &= Ax + Bu - Ax_d - Bu_d \\ &= Ax_e + Bu_e \end{aligned} \tag{4}$$

where $x_e := x - x_d$ and $u_e := u - u_d$

We can see that we can use LQR derived on the error term, and we will get the u_e that will vanish x_e(meaning making it zero),

$$u_e = Kx_e \implies u = K(x - x_d) + u_d \tag{5}$$

Note with the input, the behavior of the system is specified, which can be simulated. For example, we plug the 5 into the system equation, we can get

$$\dot{x} = Ax + B(K(x - x_d) + u_d) \tag{6}$$

4 State Equation of Quadcopter

Since we are only interesting controling angles, the system equation is quite easy. Let's assume the sensor is properly aligned with same configuration as the previous review I wrote [5]. Then, each angle can be easily expressed. For example for pitch angle θ, let us define the state $x_\theta := \begin{bmatrix}\theta & \dot{\theta}\end{bmatrix}^T$ [1]

$$\dot{x}_\theta = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x_\theta + \begin{bmatrix} 0 \\ \frac{l}{I_y} \end{bmatrix} u_\theta \tag{7}$$

Where $u_\theta := T_3 - T_1$ is the force, I_y is the moment of inertia along y-axis, and l is the length of the wing. I_y and l can be measured. Examine this equation carefully, one can see that this equation is derived from a very simple equation, $\frac{u_\theta}{l} = I\ddot{\theta}$.

The state equation for roll angle ϕ is the same, with the input $u_\phi := T_2 - T_4$

The state equation for yaw angle ψ is a little bit different. From the equation (7) in [4] for ψ, we can expressed yaw angle equation:

$$K_r[(T_1 + T_3) - (T_2 + T_4)] = mI_z\psi$$

From which we can derive

$$\dot{x}_\psi = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x_\psi + \begin{bmatrix} 0 \\ C \end{bmatrix} u_\psi \tag{8}$$

where $x_\psi := \begin{bmatrix} \psi \\ \dot{\psi} \end{bmatrix}$, $u_\psi := T_1 - T_2$, and $C = \frac{\ddot{\psi}}{u_\psi} = \frac{2K_r}{mI_z}$, under the condition that $T_1 = T_3$ and $T_2 = T_4$. This condition is for keeping roll and pitch angle unchanged. Note that there are two ways to determine C; We can meausre it, or collect data from running the system and do some calculation.

5 Apply LQR

With 7 and 8, we have A and B already. Then we need to choose Q and R. Here we use following Q and R [3]

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \quad R = \rho$$

ρ is how much we want to penalize the input; The smaller it is, the faster the convergence, but it will put more stress on the motor that provides the force. K is something we can figure out. for this case, it is [3]

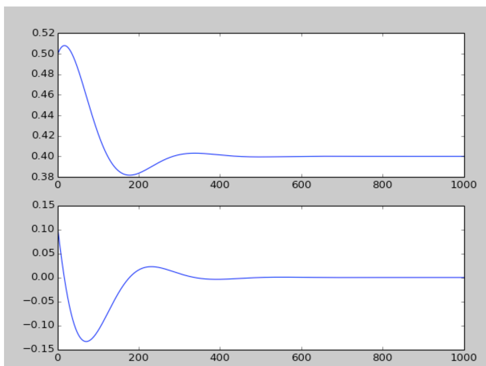
$$K = \begin{bmatrix} -\rho^{-\frac{1}{2}} & -\sqrt{2}\rho^{-\frac{1}{4}} \end{bmatrix}$$

With equation 5, We can drive state to the state we want. And equation 6 will be the system behavior over time.

6 LQR With Integral Action

We can also introduce an integral term on the controller, further detail consult [2].

7 Simulation



I will omit explanations and details of the simulation for now.

References

[1] Shayegan Omidshafiei. *Reinforcement Learning-based Quadcopter Control*

[2] R. M. Murray. *CD 2 Note LQR Control* 2006

[3] Jongeun Choi,. *ME/ECE 851-Linear Systems and Control Lecture 14 Note*

[4] S. L. Waslander†, G. M. Hoffmann, et. al. *Multi-Agent Quadrotor Testbed Control Design: Integral Sliding Mode vs. Reinforcement Learning*

[5] C. J. Wang. *State Estimation of a Quadcopter Using Kalman Filter*