LQR control of a Nonlinear Quadcopter System

Project Group F

Keith Chester, Bob DeMont, Sean Hart

A paper presented for the class of
Robot Motion - RBE 502

Robotics Engineering
Worcester Polytechnic Institute
United States of America
November 9, 2021

Abstract

The purpose of the project is to simulate the control and stable flight of an unmanned four rotor flying vehicle known as a quadcopter. The problem is set into three goals. The first is to have the quadcopter hover in a stable configuration. The second goal is to introduce a drone in flight and to have the quadcopter intercept that flight without knowing it's path. The third goal is to simulate the capture of the drone with a randomized disturbance force and maintain stable flight while returning to base.

Approaching the General Problem

To simulate the quadcopter's flight, we have to simulate its physical flight dynamics and place them into a state equation. Table 1 introduces the variables, or states, we used for basic flight dynamics, as well as constants used in computations:

$\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}$ *earth position in the x, y, z plane*	$\mathbf{c_1}, \mathbf{c_2}, \mathbf{c_3}$ *relative position of the x, y, z plane*
$\boldsymbol{\phi}, \boldsymbol{\theta}, \boldsymbol{\psi}$ *angles in the x, y, and z plane*	$\mathbf{e_1}, \mathbf{e_2}, \mathbf{e_3}$ *relative position of the x, y, z plane*
$\mathbf{v_1}, \mathbf{v_2}, \mathbf{v_3}$ *velocities in the x, y, z plane*	$\mathbf{u_1}, \mathbf{u_2}, \mathbf{u_3}, \mathbf{u_4}$ *thrust generated by quadcopter rotors*
$\boldsymbol{\omega_1}, \boldsymbol{\omega_2}, \boldsymbol{\omega_3}$ *accelerations in the x, y, z plane*	$\mathbf{g}$ *9.81 meters per second squared*
$\mathbf{m}$*mass, 0.5 kilograms*	$\mathbf{l}$ *length: 0.2 meters for each rotor arm*
$\mathbf{I_1}.\mathbf{I_2}, \mathbf{I_3}$*1.24, 1.24& 2.48 kilograms per square meter of inertia*	$\boldsymbol{\sigma}$*0.01, proportionality constant for thrust*
$\boldsymbol{\mu}$ *newtons, maximum thrust of each rotor*	

Table 1: Basic Flight Dynamic Variables and Constants

The first four variables, and states representing their derivatives, were placed into a vector named z. The purpose of the z vector is to be able to hold states which, when fed into a solver for ordinary differential equations, would generate solutions that would feed back into the controller and modify the behavior of the plant towards its desired goal. In order to do this, we have to feed back corrected values. If a controller can be placed in an invariant, state space form simply as:

$\dot{x} = Ax + Bu$

Where x are the states, and u are the inputs of the system. The thrusts are initially each set to the weight of the quadcopter divided by four, so that the system starts in equilibrium, and not in free fall. We can design a control, a -Kx to affect closed loop change. The formula is then:

$\dot{x} = (A - BK)x$

Linear Quadratic Regulation to Generate K Controlled Feedback

We used the Linear-Quadratic Regulator (LQR) approach as an optimal control technique as a manner of generating K values dynamically. LQR operates on both the A matrix (states) and B matrix (inputs), but are given two matrices, a Q and and R, whose purpose is to weigh cost so the algorithm may calculate between two factors, the state and the actuators. Since we are not asked to weigh the cost of using the actuators (as if we were conserving fuel) , we have left the R matrix as a diagonal matrix of ones. We have also not yet weighed the Q matrix, leaving it too as a matrix of diagonal ones, so that we may experiment in the future what is the best way to prioritize states in the system. LQR acts on a dynamic system of the form:

$\dot{x} = Ax + Bu$

It then calculates a cost function based on optimization. This integrates the costs of both Q and R for comparison:

$J = x_0^T F(0) x(t_0) + \int_{t_0}^{t_f} x^T Q x + u^T R u + 2x^T N u \, dt$

The control input u that minimizes this cost function is

$u = -Kx$

The K value is given by:

$K = R^{-1}(B^T P(t) + N^T)$

and P is created by solving Ricatti's continuous time differential equation:

$A^T P(t) + P(t) A - (P(t) B + N) R^{-1} (B^T P(t) + N^T) + Q = \dot{P(t)}$

P(t) is bounded by $P(0) = F(0)$

Quadcopter Hovering

We take the $\mathbf{z}$ vector and differentiate it to create a $\dot{\boldsymbol{z}}$ vector. A and B matrices are created by taking the differentiated state vector and calculating the Jacobian by z for A and by u for B. In order to make certain that our continuous linear time-invariant system is *controllable*, we must ensure that the combinations of our state and input matrices are linearly independent to the degree that we have an independent column for each state. This will ensure a state of global asymptotic equilibrium, that is, a tendency for the differential equations to move towards zero as an equilibrium point. To do so, we check rank of the combined matrix:

$C = [BABA^2BA^3BA^4BA^5BA^6BA^7BA^8BA^9BA^{10}BA^{11}B]$

and assure that we have rank at least equal to the number of states in our z vector. Once we assign the quadcopter its initial states and the amount of time we wish to span, we can also give it a desired state and calculate the trajectory of the quadcopter using ODE45 in Matlab. Since we wish to test the quadcopter to act in a hover position, we feed it a desired state which is the same as it's initial state. This will start the error ($\mathbf{e}$), and the derivation of ($\mathbf{e}$), $\dot{\boldsymbol{e}} tozero$. We code quadcopter drawing functions and plotting functions, and we determine that the quadcopter stays still (that is, it seeks out a state of global asymptotic equilibrium).

Quadcopter Interception of Target

Quadcopter Return to Base with a Disturbance Force

References

[1] Faraz Ahmad, Pushpendra Kumar, Anamika Bhandari, Pravin P. Patil. Simulation of the Quadcopter Dynamics with LQR based Control. Materials Today: Proceedings,Volume 24, Part 2, 2020, Pages 326-332, ISSN 2214-7853. https://doi.org/10.1016/j.matpr.2020.04.282.

[2] Jinho Kim, S. Andrew Gadsden, Stephen A . Wilkerson. "A Comprehensive Survey of Control Strategies for Autonomus Quadrotors". arXiv:2005.09858v1. 20 May 2020.

[3] Madani, T, and A Benallegue. "Backstepping Control for a Quadrotor Helicopter." 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2006. 3255–3260. Web.

[4] Jing Qiao, Zhixiang Liu, Youmin Zhang. "Payload Dropping Control of an Unmanned Quadrotor Helicopter Based on Backstepping Controller". MATEC Web Conf., 277 (2019) 01004. DOI: https://doi.org/10.1051/matecconf/201927701004.

[5] Jia, Zhenyue and Yu, Jianqiao and Ai, Xiaolin. "Integral Backstepping Control for Quadrotor Helicopters". Association for Computing Machinery. 2017. DOI:https://doi-org./10.1145/3057039.3057052

[6] Daewon Lee, H JIN Kim, Shankar Sastry. Feedback Linearization vs Adaptive Sliding Mode Control for a Quadrotor Helicopter. International Journal of Control, Automation, and Systems (2009). DOI 10.1007/s 12555-009-0311-8.