Inner-Outer loop control for Quadrotor UAV with input and state constraints

Project report for AE700 : Guidance & Control of Unmanned Autonomous Vehicles

Adityaya Dhande
210070005
Dept. of Electrical Engineering
Indian Institute of Technology, Bombay
Mumbai, Maharashtra
210070005@iitb.ac.in

Priyanshu Gupta
210070064
Dept. of Electrical Engineering
Indian Institute of Technology, Bombay
Mumbai, Maharashtra
210070064@iitb.ac.in

Madharapu Srisai
22M0061
Dept. of Aerospace Engineering
Indian Institute of Technology, Bombay
Mumbai, Maharashtra
22m0061@iitb.ac.in

Yerramilli Akhilesh
22M0016
Dept. of Aerospace Engineering
Indian Institute of Technology, Bombay
Mumbai, Maharashtra
22m0016@iitb.ac.in

I. Overview

The paper that we have chosen is titled **Inner-Outer Loop Control for Quadrotor UAVs With Input and State Constraints** by *Ning Cao* and *Alan F. Lynch*. It involves designing inner loop and a saturated outer loop controller for a Quadrotor. The outer loop takes in the desired position and heading angle and calculates the saturated thrust as well as the desired roll and pitch angles. These are then given as input to the inner loop which uses a dynamic inversion model to generate the required forces and torques. We then simulated the controller using Python, assuming ideal conditions like absence of air friction, absence of gyroscopic moments, absence of blade flapping, ideal propulsors, diagonal inertia matrix, availability of accurate state estimates of the quadrotor. We applied this controller to get the quadrotor to rise to a set height and to follow a triangular trajectory at a fixed height as in the paper.

II. Quadrotor Model

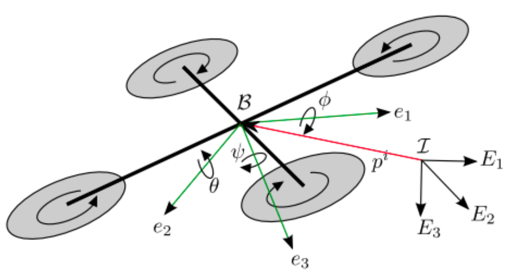

Fig. 1: Model of the quadrotor

Writing the kinematic and dynamic equations in the body frame and choosing the state of the quadrotor as $\begin{bmatrix} p_x^b & p_y^b & p_z^b & u & v & w & \phi & \theta & \psi & p & q & r \end{bmatrix}^T$. The kinematic and dynamic equations are :

$$\dot{p}^b = -\mathrm{sk}(\omega^b)p^b + v^b \tag{1}$$

$$\dot{v}^b = -\mathrm{sk}(\omega^b)v^b + gR_i^b E_3 - \frac{T}{m}e_3 \tag{2}$$

$$\dot{\eta} = W(\eta)\omega^b \tag{3}$$

$$\dot{\omega}^b = J^{-1}\{\tau^b - \omega^b \times J\omega\} \tag{4}$$

where the position, velocity and angular velocity is expressed in the body frame, $\eta = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T$, R_i^b is a rotation matrix from the inertial to body frame according to the sequence $Z - Y' - X''$. ϕ, θ, ψ denote the roll, pitch and yaw respectively. J is the inertia tensor.

$$W(\eta) = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)\sec(\theta) & \cos(\phi)\sec(\theta) \end{bmatrix}$$

III. Controller design

The desired position of the quadrotor in the inertial frame is given as p_d^i. The desired position in the body frame is given by $p_d^b = R_i^b p_d^i$. The outer loop takes the desired position as the input and calculates the required thrust and the desired roll and pitch angles given by θ_d and ϕ_d. The inner loop generates torques to track the attitude commanded by the outer loop. The thrust and torques are fed as inputs to the quadrotor model. The loop structure is as shown in the figure.

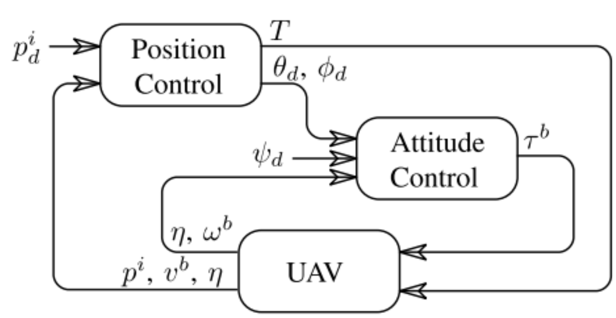

Fig. 2: Inner-outer loop controller

δ_1 is defined as $\delta_1 = p^b - p_d^b$. We employ a virtual controller to get the accelarations along the x,y and z directions as $u = \begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}^T$. The virtual control is supposed to be similar to a proportional controller, but has to be saturated due to input constraints. Defining $y_1 = \begin{bmatrix} y_{1,1} & y_{1,2} & y_{1,3} \end{bmatrix}^T = k_2\delta_1 + v^b$ and $y_2 = \begin{bmatrix} y_{2,1} & y_{2,2} & y_{2,3} \end{bmatrix}^T = v^b$, we can calculate u as $u = -\Sigma_2(k_2 y_2 + \Sigma_1(k_1 y_1))$ where k_1 and k_2 are control gains and Σ_i are saturation functions given by $\Sigma_i([s_1, s_2, s_3]^T) = [\sigma_{i,1}(s_1), \sigma_{i,2}(s_2), \sigma_{i,3}(s_3)]$ We used the same $\sigma_{i,j}(s)$ as in the paper, it is as follows,

$$\sigma_{i,j}(x) = \begin{cases} x & \text{if } |x| \leq L_{i,j} \\ \text{sign}(x)\left(L_{i,j} + \frac{\beta_{i,j} - \beta_{i,j} e^{-2(|x| - L_{i,j})}}{1 + (2\beta_{i,j} - 1)e^{-2(|x| - L_{i,j})}}\right) & \text{otherwise} \end{cases}$$

$\beta_{i,j} = M_{i,j} - L_{i,j}$. We have to choose $M_{i,j}$ and $L_{i,j}$. As in the paper we choose $M_{i,j} = 0.95 L_{i,j}$, $M_{1,1} = M_{1,2} = M_{1,3}$ and $M_{2,1} = M_{2,2} = M_{2,3}$ The control gains that we have chosen are tabulated below :

TABLE I: Chosen gain values

Gain	Value
k_1	1.0
k_2	0.5
k_3	3.0
$M_{1,i}$	0.58
$M_{2,i}$	3.8

The values of k_1, k_2 and k_3 are slightly different from those given in the paper. The values of $M_{i,j}$ are not mentioned in the paper but we have chosen them such that they satisfy the inequalities described by the paper. Now we have all the parameters for the saturation function and that completes the virtual controller, which gives us u. We now have to generate θ_d and ϕ_d using u and the dynamic inversion model. We choose $\psi_d = 0$ to simplify the equations, which then gives us,

$$\theta_d = -\arcsin(\frac{u_1}{g}) \tag{5}$$

$$\phi_d = \arcsin\left(\frac{u_2}{g\cos(\theta_d)}\right) \tag{6}$$

$$T = m(g\cos(\theta_d)\cos(\phi_d) - u_3) \tag{7}$$

θ_d and ϕ_d are inputs to the inner loop. For the inner loop, we define e_η as $e_\eta = \eta - \eta_d$ where $\eta_d = [\phi_d, \theta_d, \psi_d]^T$ and e_ω as $e_\omega = W(\eta)\omega^b$, by neglecting $\dot{\eta}_d$. We use a PD controller (paper uses PID) for the inner loop such that,

$$\ddot{e}_\eta = \bar{\tau}^b = -k_p^a e_\eta - k_d^a e_\omega$$

The actual torques are calculated as

$$\tau^b = \omega^b \times J\omega^b + J\{W^{-1}\bar{\tau}^b\}$$

We have chosen $k_p^a = \text{diag}([7, 7, 7])$ and $k_d^a = \text{diag}([2, 2, 2])$ This completes the inner loop, and hence the controller.

IV. Simulation results

We have simulated the controller and quadrotor using Python and used Euler's method to solve the set of coupled differential equations of the quadrotor dynamics and kinematics, taking even time steps of 0.01s duration. The physical parameters that we chose are tabulated below.

TABLE II: Chosen parameters

Parameter	Value (S.I. units)
Mass	1.0
$J_x x$	0.0820
$J_y y$	0.0845
$J_z z$	0.1377
$J_x z$	0

We have simulated a setpoint reaching manoeuvre and a triangular trajectory tracking manoeuvre. The state of the quadrotor is intialised to $[0\ 0 \cdots 0]^T$ at the start of both the simulations.

A. Hovering

In this simulation the desired point is a constant vector, given by $p_d^i = [0, 0, -1.2\text{m}]^T$, for 20s. This is vertically above the starting point of the quadrotor. The thrust and altitude vary with time as in Fig. 3. The velocity along the z-axis of the inertial frame varies with time as in Fig. 4.

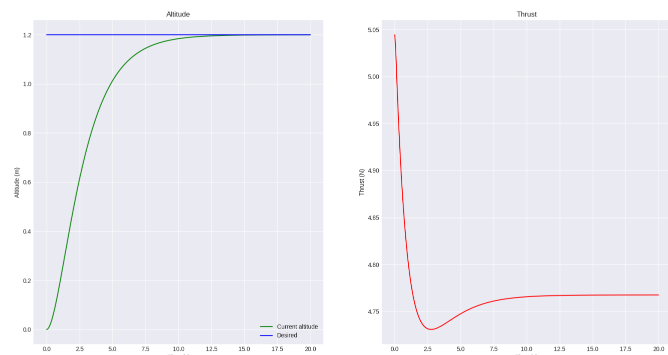

Fig. 3: Plot of altitude and thrust vs time

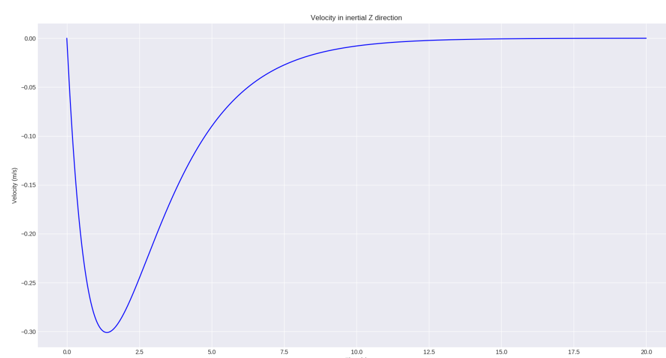

Fig. 4: Plot of v_z^i vs time

B. Tracking linear trajectories

The simulation time is 300s and the desired point changes with each time step forming a triangle.

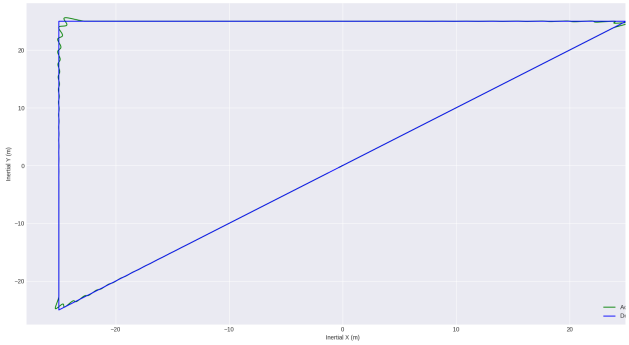

Fig. 5: Actual and desired triangular trajectories

Fig. 5 shows the plot of the actual and desired trajectories of the quadrotor in the x-y plane The triangular trajectory is at height of 20m, and is formed by the points $[25, 25, -20]^T$m, $[-25, 25, -20]^T$m and $[-25, -25, -20]^T$m. The position of the quadrotor in the inertial frame is plotted against time in Fig. 6 along with the desired values of the position in the inertial frame. In Fig. 7 the angles ϕ, θ and ψ are plotted against time along with ϕ_d, θ_d and ψ_d. Fig. 8 has the velocities of the quadrotor along the axes of the inertial frame against time. Fig. 9 has the thrust and torques which are given as inputs to the dynamic model of the quadrotor plotted against time.

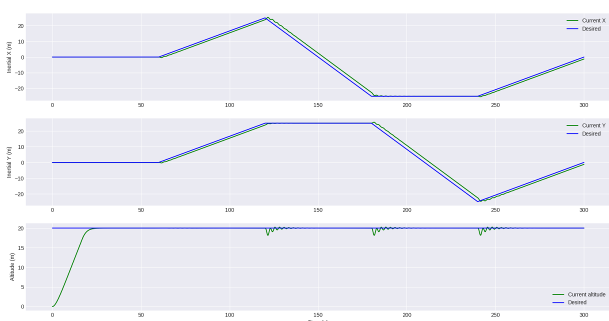

Fig. 6: Position in inertial frame vs time

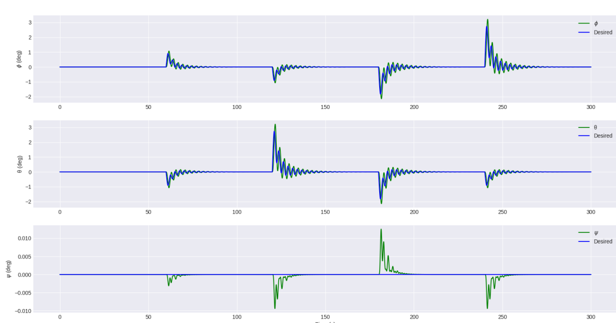

Fig. 7: Euler angles of body frame vs time

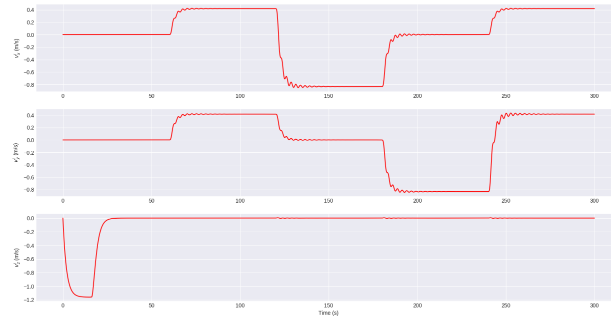

Fig. 8: Velocities in inertial frame vs time

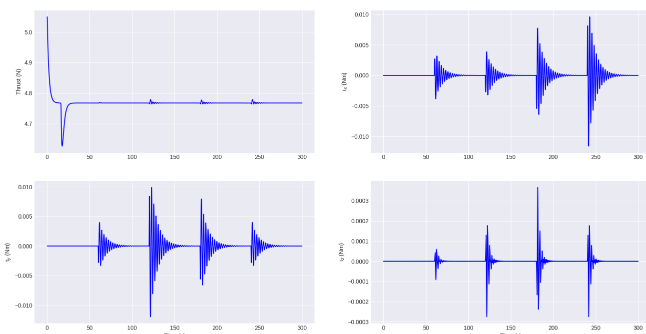

Fig. 9: Velocities in inertial frame vs time

V. Conclusion

The simulation results are matching with the experimental results summarised in the paper. The major difference is the absence of noise and disturbances as we have not accounted for them in the simulation and assumed availability of perfect state estimates. There is a slight drop in the altitude when a setpoint is about to be reached, when the quadrotor changes orientation. The graphs are attached as `.png` files in the submission folder. The `README.txt` file describes the Python files needed for running the simulation.

VI. References

[1] N. Cao and A. F. Lynch, "Inner–Outer Loop Control for Quadrotor UAVs With Input and State Constraints," in IEEE Transactions on Control Systems Technology, vol. 24, no. 5, pp. 1797-1804, Sept. 2016, doi: 10.1109/TCST.2015.2505642.

[2] Quan Quan, "Introduction to Multicopter Design and Control", Springer, 2017