Model Predictive Control for Quadrotor under the Influence of Turbulence Flow

Xueyang Qi
University of Pennsylvania
Philadelphia, Pennsylvania
xueyangq@seas.upenn.edu

Yingzhuo Wang
University of Pennsylvania
Philadelphia, Pennsylvania
yw28@seas.upenn.edu

Ruizhe Wang
University of Pennsylvania
Philadelphia, Pennsylvania
ruizhew@seas.upenn.edu

***Abstract*—Quadrotors can be used to deliver supplies in the event of a disaster, but may encounter turbulence during flight. In our project, we applied Model Predictive Control (MPC) to a quadrotor in 3D space. We explored the influence of different parameters of controller on the system response. In addition, we added payload to the quadrotor, and also included turbulence flow as the interference of the quadrotor movement. Our simulation results demonstrated that our MPC controller can enable the quadrotor to reach target positions and shew resistance to the effects of low speed turbulence flow.**

***Index Terms*—quadrotor, Model Predictive Control (MPC), turbulence flow, payload**

I. Introduction

In the wake of natural disasters, the rapid and efficient delivery of essential supplies to affected areas is a critical component of effective disaster response. Traditional methods of aid delivery often encounter significant challenges due to disrupted or inaccessible transportation routes. Addressing this vital need, our study introduces a groundbreaking development in disaster response technology: an advanced quadrotor equipped with MPC with an rejection to the turbulence flow on air. The innovative approach presents a transformable solution for delivering food and supply to those who are in the challenging environments.

With the MPC strategy, the system can make the next-stage estimation of thrust input at discrete steps, combining the thrust constraints and cost optimization. For a 3D quadrotor dynamics, a linearization is used to linearize a non-linearized system by taking the first order Taylor expansion, which simplifies the complex, non-linear system into a more manageable linear form. At this stage, turbulence flow disturbance randomly generated from Von Karman model is applied to the system. The turbulence flow separates into three different velocity with given intensity applied on the x,y,and z direction of the system. In addition, "wing" function from the Drake simulator allows the quadrotor to interact with the wind applied. By applying fluid dynamics on the rotors, the "wing" function allows the turbulence flow to exert elevation and drag forces on the quadrotor. This realistic interaction enhances the drone's ability to navigate in conditions closely mimicking real-world scenarios, particularly in the presence of strong and erratic wind patterns. Except from the different levels of disturbance, payload mass toleration is another crucial factor to take into account. A solid cube with a given mass is attached under the bottom of the quadrotor. The controller must adjust the input thrust to lift and stabilize the quadrotor with the payload. This adaptability is critical in disaster response scenarios, where the drone might need to carry various types of supplies, each with different weights and sizes

II. Methodology

The basic flow diagram of the program is shown below:

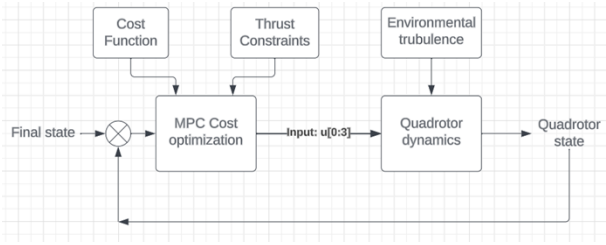


Fig. 1: Diagram flow chart

The system takes a final desired state as an input. In the MPC block, the cost optimization function minimizes the next-step cost under the given thrust limit. At this stage, a thrust input can be generated from MPC process. The thrust input. combining with the turbulence flow, can produce a new quadrotor state which takes back to the beginning of the system.

A. Quadrotor Dynamics

The full state of the quadrotor is shown below[1]:

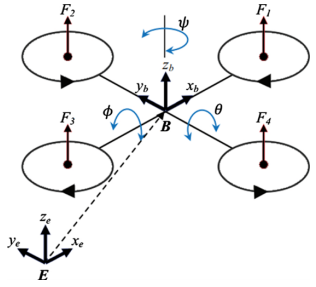


Fig. 2: Quadrotor 3D model

$$X = \left[x, y, z, \phi, \theta, \psi, \dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi}\right]^T \tag{1}$$

$$\dot{X} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \ddot{x} \\ \ddot{y} \\ \ddot{z} \\ \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \\ p + (\sin\phi\tan\theta)q + (\cos\phi\tan\theta)r \\ q\cos\phi - r\sin\phi \\ \frac{\sin\phi}{\cos\theta}q + \frac{\cos\phi}{\cos\theta}r \\ -(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)\frac{u_1}{m} \\ -(\cos\phi\sin\theta\cos\psi - \sin\phi\sin\psi)\frac{u_1}{m} \\ -(\cos\phi\cos\theta)\frac{u_1}{m} + g \\ \frac{(I_y - I_z)qr + u_2}{I_x} \\ \frac{(I_z - I_x)pr + u_3}{I_y} \\ \frac{(I_x - I_y)pq + u_4}{I_z} \end{bmatrix} \tag{2}$$

The non-linear dynamics can be linearized by taking first-order Taylor expansion:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ A &= \left.\frac{\partial f(x,u)}{\partial x}\right|_{x=x_{current}} \\ B &= \left.\frac{\partial f(x,u)}{\partial u}\right|_{u=u_{current}} \\ u_k &= \left[F_1, F_2, F_3, F_4\right]^T \end{aligned} \tag{3}$$

B. MPC cost optimization

The cost optimization:

$$\begin{aligned} \min_{x[k],u[k]} \sum_{k=0}^{N} \Big[& (x[k] - x_{final})^T Q (x[k] - x_{final}) \\ & + (u[k] - u_{final})^T R (u[k] - u_{final}) \Big] \\ subj.to \quad & x_{k+1} = Ax_k + Bu_k \\ & x[0] = x_{current} \\ & u_{final} = \left[\tfrac{1}{4}mg \quad \tfrac{1}{4}mg \quad \tfrac{1}{4}mg \quad \tfrac{1}{4}mg\right]^T \\ & u_{lowerlimit} \leq u \leq u_{upperlimit} \end{aligned} \tag{4}$$

C. Von Karman Turbulence Flow Model

The forming filters are approximation of the Von Karman velocity spectra which are valid in a range of normalized frequencies of less than 50 radians.[2]

Longitudinal:

$$H_u(s) = \frac{\sigma_u \sqrt{\frac{2}{\pi V}} L_u \left(1 + 0.25\left(\frac{L_u}{V}\right)\right)}{1 + 1.357\left(\frac{L_u}{V}s\right) + 0.1987\left(\frac{L_u}{V}s\right)^2} \tag{5}$$

Lateral:

$$H_v(s) = \frac{\sigma_v \sqrt{\frac{1}{\pi V}} L_v \left(1 + 2.7478\left(\frac{L_v}{V}s\right) + 0.3398\left(\frac{L_v}{V}s\right)^2\right)}{1 + 2.9958\left(\frac{L_v}{V}s\right) + 1.9754\left(\frac{L_v}{V}s\right)^2 + 0.1539\left(\frac{L_v}{V}s\right)^3} \tag{6}$$

Directional:

$$H_w(s) = \frac{\sigma_w \sqrt{\frac{1}{\pi V}} L_w \left(1 + 2.7478\left(\frac{L_w}{V}s\right) + 0.3398\left(\frac{L_w}{V}s\right)^2\right)}{1 + 2.9958\left(\frac{L_w}{V}s\right) + 1.9754\left(\frac{L_w}{V}s\right)^2 + 0.1539\left(\frac{L_w}{V}s\right)^3} \tag{7}$$

For low-altitude model (altitude < 1000 ft.):

$$\begin{aligned} L_w &= h \\ L_u &= L_v = \frac{h}{(0.177 + 0.000823h)^{1.2}} \\ \sigma_w &= 0.1 W_{20} \\ \frac{\sigma_u}{\sigma_w} &= \frac{\sigma_v}{\sigma_w} = \frac{1}{(0.177 + 0.000823h)^{0.4}} \end{aligned} \tag{8}$$

For medium/high altitudes (altitudes > 2000 ft.):

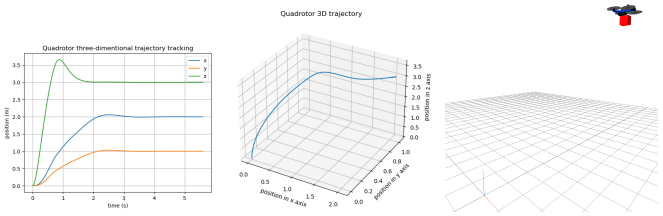
$$L_u = L_v = L_w = 2500ft \tag{9}$$

III. Results

For all the figures shown in the results, their desired position are all $[2, 1, 3]$.

A. Plain verification of MPC controller

Among all simulations, the primary experiment was to verify that whether the MPC controller could drive the quadrotor to the desired destination without any disturbances, namely testing the net capability of the controller.
Aiming to test as comprehensively as possible, several trials were conducted with both normal and some extreme positions. The results turned out that the MPC controller was capable of driving the quadrotor to the desired positions over all demonstrations. The below figure 3 illustrate an example plot of the simulations.



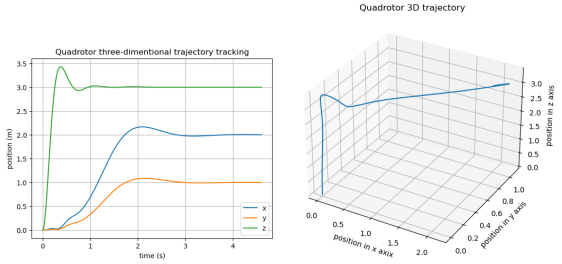
(a) Axial trace tracks. (b) 3D trajectory. (c) 3D viewer.

Fig. 3: Trajectory of quadrotor without any disturbances.

B. Parameter changes validation

Ensuring the capability of the MPC controller, we began to explore how parameter changes will influence the response of the quadrotor so that could further examine the ability of the controller.
Among all parameters utilized in the controller computation, we focused on Q and R, and conducted simulations correspondingly. The responses with different Q and R were in

accordance with the analysed results as expected. Specifically, larger magnitude in Q value increased the sensitivity to changes on the corresponding parameters while smaller magnitude in R brought about more delicate control via changes in u (since every unit changes in u will be counted in a moderate scale; consider if R were too large, a very small change in u would cause quite large costs). However, in practice, the chosen of R should be moderate in view of energy dissipation. The below figure 4 is an example plot of the case which increased the magnitude of Q value corresponded to z.

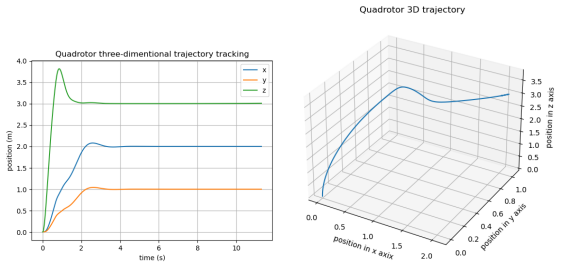


(a) Axial trace tracks. (b) 3D trajectory.

Fig. 4: Trajectory of quadrotor with lager magnitude Q value corresponded to z.

C. Turbulence flow anti-interference test

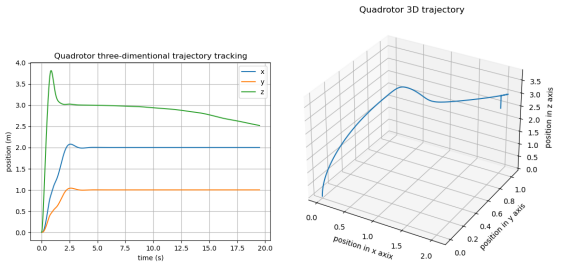
As introduced in the methodology section, we adapted Von Karman Wind Turbulence Model as the ambient interference to test the anti-jamming capability of our MPC controller. The experiments were conducted over a gradient scales of speeds turbulence flow. The results showed that the quadrotor system could only resist low speed turbulence flow and for those higher speed turbulence flow, the quadrotor could reach the desired position at the very initial and then deviated from that position, where the deviation speed was tend to be positively correlated with the turbulence flow speed. The below figures 5-7 illustrated the results.



(a) Axial trace tracks. (b) 3D trajectory.

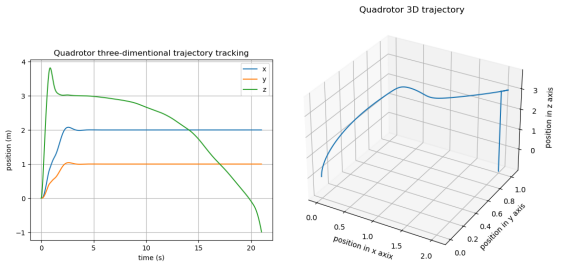
Fig. 5: Trajectory of quadrotor with low speed turbulence flow.

According to the three sets of figures, for the unknown turbulence disturbances, the influence was all largely exerted on z axis for all cases. From our perception and analysis, we deemed that this phenomenon was mainly caused by the plug in of "Wing" class in Drake[3]. Referring to the



(a) Axial trace tracks. (b) 3D trajectory.

Fig. 6: Trajectory of quadrotor with high speed turbulence flow.



(a) Axial trace tracks. (b) 3D trajectory.

Fig. 7: Trajectory of quadrotor with extremely high speed turbulence flow.

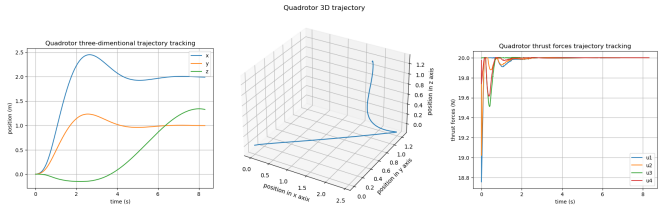
"Wing" class documentation, it indicates that currently it only supports a particular model of flat-plate aerodynamics, which inferred that the spatial force generated by the turbulence flow was large for lift (vertical dimension) and small for drag (horizontal), verifying that the turbulence influence was mainly applied on z axis. Further, since the turbulence flow was unknown to the MPC controller so that its computation goal was still to maintain the state when at the desired position, therefore could not resist the continuously increasing spatial force by the increasing turbulence flow speed and the result is that the quadrotor gradually flew down with an increasing speed.

D. Payload tolerance test

Under the low speed turbulence flow environment, we further conducted a series of experiment on payload to test the payload tolerance of the system with known payload mass. The results showed that as long as the maximally required thrust within the thrust limit, the quadrotor could reach the desired position. Otherwise, the quadrotor could only reach partial height or even be blown down to negative height (in practical, it could not take off), with thrust at the thrust limit. The below figures 8-9 showed some large payload cases.

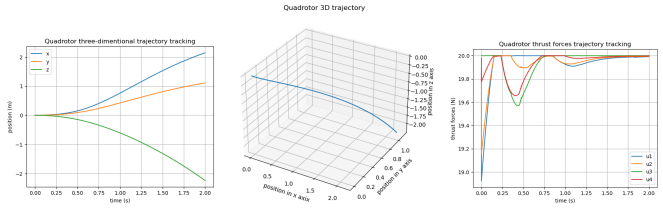
Conclusion

Our simulation results proved that our MPC controller can make the quadrotor reach the position we desired and had certain anti-interference. In some cases, the MPC controller



(a) Axial trace tracks. (b) 3D trajectory. (c) Thrust trace track.

Fig. 8: Trajectory of quadrotor with large payload under low speed turbulence flow environment.



(a) Axial trace tracks. (b) 3D trajectory. (c) Thrust trace track.

Fig. 9: Trajectory of quadrotor with extremely large payload under low speed turbulence flow environment.

may fail. We have found two ways to fail. The first was that if the quadrotor can't reach the target point. This happened when the payload was too large. This happened because thrust limits were fixed and the propeller cannot provide enough thrust to lift the quadrotor. The second way was that even though our quadrotor can reach the target point, it then flew down and away from the target point, which happened when the turbulence flow velocity was very high. We checked the thrust and found it did not reach the limit throughout the flying-down move, which it should did. Similar to the analysis in the results, the desired final state for the controller was the state when at the desired point. With the gradually increasing turbulence disturbance, the thrust increased as well to maintain the preset final state until reaching the thrust limit and then flew down. However, in our simulation, the quadrotor began to fly down when the thrust had not reaches its limit. The cause of the problem may be the "Wing" Class in Drake[3] that we used to apply disturbing forces to the quadrotor.

In our work, we attached the payload to the quadrotor and we calculated them as a whole. In the future, we could try to separate the payload with the quadrotor so that it can be dropped when it reaches the desired point. In real life, the payload may be attached to the quadrotor by a rope. In addition, our work was not accurate about how the quadrotor with a payload was affected by turbulence flow. We just used the Wing Class in Drake and that is not suitable for a quadrotor with four propellers, so we needed to improve aerodynamics in our system.

REFERENCES

[1] F. Sabatino, "Quadrotor control: modeling, nonlinear control design, and simulation," Master's thesis, School of Electrical Engineering (EES), Automatic Control, KTH Royal Institute of Technology, Stockholm, Sweden, 2015.

[2] "Von Karman Wind Turbulence Model (Continuous)," MathWorks - Makers of MATLAB and Simulink. [Online]. Available: https://www.mathworks.com/help/aeroblks/vonkarmanwindturbulence-modelcontinuous.html. [Accessed: 12/11/2023].

[3] "Drake Robotics Toolkit, 'pydrake.multibody.plant module,' [Online]. Available: https://drake.mit.edu/pydrake/pydrake.multibody.plant.html. [Accessed: 12/11/2023]."