Safe MPC for Quadrotors using Control Barrier Functions

Eric Xu, Maitham AL-Sunni

***Abstract*—Autonomy of quadrotors has been a hot topic in the past decade. A main challenge of quadrotors' autonomy is the safety concerns and how a quadrotor can deal with obstacles while flying. In this work we present a control policy that utilizes model predictive control and control barrier functions to achieve safety. An example with a quadrotor avoiding multiple obstacles is presented to demonstrate the efficiency of the control policy.**

I. Introduction

Quadrotors are of great interest nowadays in multiple areas of life. A couple of these areas are agriculture, delivery services, surveillance, etc. In fact, we have recently seen many applications in load delivery for quadrotors [3], homeland security and other general surveillance applications (where remote surveillance would provide a much safer alternative to in person) have arose from the further development in quadrotors technology [4]. Because these quadrotors are often used for observation work, we must ensure that they can both safely and efficiently complete their tasks. Hence the motivation for this work. In this work, we will be combining two control methodologies to produce a method in which we can guarantee the safety of a quadrotor while ensuring that it follows its desired trajectory as closely as possible. The particular methods that we will be discussing in this paper are Model Predictive Control (MPC) and Control Barrier Functions (CBFs) for which the former is already an extensively researched technique and the latter being a more recently developed control strategy by Ames [1]. In particular, the most noteworthy thing about CBFs are the fact that we can transform non-convex state constraints into convex control constraints and solve our optimal control problem as a standard quadratic program (this will be further developed in Section IV).

II. Quadrotor's Model

We develop our methods based on the well-known quadrotor model that utilizes quaternions to model states that are related to attitude. The model has 4 main parts, position (${}^{N}r$) that has three associated states (in 3D), a 4D quaternion that represents angles of rotations (i.e., yaw, pitch, and roll), linear velocity (${}^{B}v$) that has three associated states (in 3D), and angular velocity (${}^{B}\omega$) that has three associated states (in 3D) as well.

$$x = \begin{bmatrix} {}^{N}r \in \mathbb{R}^3 \textit{ Position in N frame} \\ q \in \mathbb{H} \textit{ Attitude } (B \to N) \\ {}^{B}v \in \mathbb{R}^3 \textit{ Linear velocity in B frame} \\ {}^{B}\omega \in \mathbb{R}^3 \textit{ Angular velocity in B frame} \end{bmatrix} \tag{1}$$

And the dynamics of the quadrotor is given by the following:

$$\dot{x} = \begin{bmatrix} Q^{B}v \\ \frac{1}{2}L(q)H^{B}\omega \\ \frac{1}{m}{}^{B}F -{}^{B}\omega \times {}^{B}v \\ J^{-1}({}^{B}\tau -{}^{B}\omega \times J^{B}\omega) \end{bmatrix} \tag{2}$$

Where:

$${}^{B}F = Q^T \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ k_T & k_T & k_T & k_T \end{bmatrix} \tag{3}$$

$${}^{B}\tau = \begin{bmatrix} 0 & sk_T & 0 & -sk_T \\ -sk_T & 0 & sk_T & 0 \\ k_m & -k_m & k_m & -k_m \end{bmatrix} \tag{4}$$

The rotational matrix Q, can be computed for a quaternion q using:

$$Q = R^T(q)L(q) \tag{5}$$

And the transformation from the body frame to the world frame is computed using:

$${}^{N}x = Q^{B}x \tag{6}$$

III. MPC for Tracking

Although there are many ways to complete a tracking task, we know upfront that MPC would most likely outperform other methods especially with the presence of constraints. Accordingly, in order to solve the tracking problem, we used the following model predictive control formulation:

$$\begin{aligned} \min_{X,U} \quad & (x_N - x_r)^T Q_f (x_N - x_r) + \sum_{k=0}^{N-1} (x_k - x_r)^T Q (x_k - x_r) + u_k^T R u_k \\ \text{s.t.} \quad & x_{k+1} = Ax_k + Bu_k \\ & u_{min} \leq u_k \leq u_{max} \end{aligned} \tag{7}$$

And to solve this MPC problem we formulate it as a quadratic program as follows:

Code: github.com/Maladyne/Optimal-Control-Project

Both authors are with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, USA, {ex, malsunni}@andrew.cmu.edu

$$
\begin{aligned}
\text{minimize}_z \quad & \frac{1}{2} z^T P z \\
\text{subject to} \quad & Dz = d \\
& Cz \leq c
\end{aligned} \tag{8}
$$

Where z is the vector that contains all inputs and states of the system: $\begin{bmatrix} u_0 & x_1 & u_1 & \dots & x_{N-1} & u_{N-1} & x_N \end{bmatrix}^T$, P is the matrix that has the costs of each input and state, D is the dynamics constraints matrix, and C is a matrix that selects the states or the inputs we want to limit by the entries of the vector c.

IV. Control Barrier Functions

CBFs are a relatively new control method generally applied to safety applications [1]. As mentioned in the introduction, one of the most noteworthy things about CBFs is that we can rewrite nonconvex state constraints as convex input constraints; here we will delve a little bit into the theory of CBFs. We assume we have dynamics of the following form

$$\dot{x} = f(x) + g(x)u, \tag{9}$$

note that this formulation is affine in our control input u. Then we define a safety set:

$$\mathcal{C} = \{x | h(x) \geq 0\}, \tag{10}$$

which one could think about as a superlevel set of the control barrier function $h(x)$ which encodes our safety requirement. As can be seen, the main idea behind CBFs is to ensure that this CBF h remains nonnegative, i.e. we have that, for all time, $\dot{h} \geq 0$, which we can rewrite in terms of the dynamics (Eq. 9) of our system

$$\frac{\partial h}{\partial x}\frac{\partial x}{\partial t} = \frac{\partial h}{\partial x}\left(f(x) + g(x)u\right) \geq -\alpha\left(h(x)\right), \tag{11}$$

for which we can see that this is an affine constraint with respect to the control input u and we define α as a class κ function which ensures that we don't get too far from the boundary of the safety set 10. Thus, assuming we already have some controller u_{nom} that does reference tracking for us, we may form the following QP that guarantees safety:

$$
\begin{aligned}
&\min_u \frac{1}{2}\|u - u_{nom}\|_2^2 \\
&\text{s.t. } \frac{\partial h}{\partial x}\left(f(x) + g(x)u\right) + \alpha\left(h(x)\right) \geq 0.
\end{aligned}
$$

A. Exponential Control Barrier Functions

One critical issue with the regular formulation of CBFs are the fact that we require the CBF, h, to have relative degree 1 wrt u (meaning that the first time derivative of h depends on u, the control input). In our case, and many others, this is not true. So we in order to generalize CBFs we have Exponential CBFs (ECBFs) [5]. In particular, we define the CBF $h(x)$ in the same way ,but this time it has relative degree r meaning that

$$h^r(x,u) = L_f^r h(x) + L_g L_f^{r-1} h(x) u$$

Where we've used the lie derivative notation for time derivatives for convenience and we further define the "state" variable

$$\eta(x) = \begin{bmatrix} h(x) \\ \dot{h}(x) \\ \vdots \\ h^{r-1}(x) \end{bmatrix} = \begin{bmatrix} h(x) \\ L_f h(x) \\ \vdots \\ L_f^{r-1} h(x), \end{bmatrix} \tag{12}$$

noting that this variable is only a function of x because of the relative degree of the CBF. Then we define the following linear system

$$
\begin{aligned}
\dot{\eta}(x) &= \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \eta(x) + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \mu \\
h(x) &= \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \eta(x)
\end{aligned}
$$

Where we choose u such that $h^r(x,u) = \mu$: Then we apply ordinary linear feedback control (i.e. pole placement) $\mu \geq -K\eta(x)$ where the inequality renders $h(x)$ positive for all time under the comparison lemma [5].

V. Results

To test our methods, we implement a tracking problem; our goal is to track a circular path of a specific radius but also to consider the presence of some obstacles in that path so that we can avoid them. Our methodology is to first solve the tracking problem in Section III as if there are no obstacles in the path, and from that we get the nominal states and inputs, X_{nom} and U_{nom}. After that we use U_{nom} to solve the other quadratic program given in Section IV to get U that will ensure that the states X are inside the safe set we defined (i.e., the distance to any present obstacle ≥ 0)

We implemented the method on a quadrotor with assumed parameters as follows: m (mass) = 0.5 kg, l (length to CoM) = 0.175, k_T = 1, k_m = 0.0245, J = $\text{diag}(0.0023, 0.0023, 0.004)$. The setting of the tracking problem we consider is: T_{final} = 50 seconds, frequency (h) = 20 Hz.

For the safety of our quadrotor, we use a QP in conjunction with the output from the MPC with ECBF constraints. In particular the CBF we choose is

$$h(x) = x^2 + (y-1)^2 + (z-2)^2 - 0.5^2 \tag{13}$$

which encodes the obstacle (a sphere with radius 0.5 centered at the coordinates $(x,y,z) = (0,1,2)$), then we proceed as in Section IV-A and develop the QP

$$
\begin{aligned}
&\min_u \frac{1}{2}\|u - u_{nom}\|_2^2 \\
&\text{s.t. } L_f^2 h(x) + L_g L_f h(x) u \geq -K\eta(x)
\end{aligned}
$$

Where

$$\eta(x) = \begin{bmatrix} h(x) \\ L_f h(x) \end{bmatrix}$$

and we use the matrix $K = [1.65, 1.65]$.

So, after solving the tracking problem (for a circular path of radius of 1 meter), we get an unsafe trajectory that we called ”nominal”.

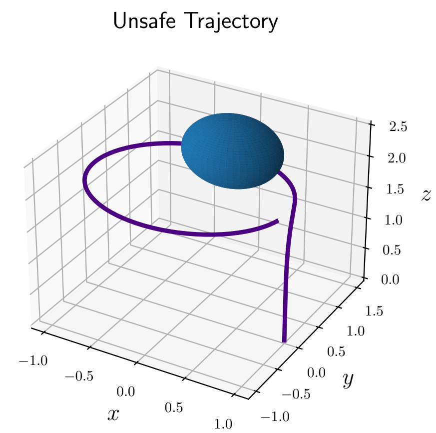

Fig. 1. The nominal trajectory (before applying CBFs).

After that, we use the nominal control inputs to solve the safety problem in Section IV, to get the following safe trajectory:

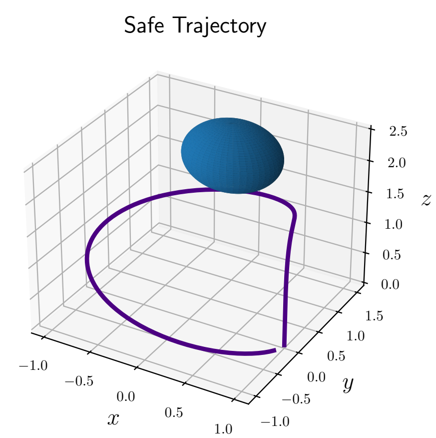

Fig. 2. The safe trajectory.

It is clear from the results in the previous figures that the resulting MPC trajectory is violating safety as the quadrotor crashes into the obstacle (the sphere). However, as the method in Section IV is applied, the quadrotor avoids the obstacle.

VI. Conclusion and Future Work

In this work we presented a way of applying control barrier functions in parallel with a model predictive controller to achieve safety for a quadrotor that is performing a tracking path. Our results have currently achieved the main goal which

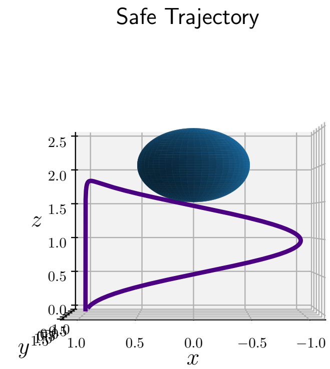

Fig. 3. The safe trajectory from a different angle

is avoiding obstacles successfully; however, as a next step, we plan to enhance the tracking results to make the objective costs as minimal as possible while achieving safety as well. Also we plan to make the obstacle avoidance work for higher speeds, and to add multiple obstacles.

References

[1] Ames, A. D., Xu, X., Grizzle, J. W., & Tabuada, P. (2016). Control barrier function based quadratic programs for safety critical systems. IEEE Transactions on Automatic Control, 62(8), 3861-3876.

[2] Zhou, K. (2021). Safety Control in Quadrotors Based on Control Barrier Function [Master’s Thesis, Eindhoven University of Technology].

[3] I. H. B. Pizetta, A. S. Brandão and M. Sarcinelli-Filho, ”Load Transportation by Quadrotors in Crowded Workspaces,” in IEEE Access, vol. 8, pp. 223941-223951, 2020, doi: 10.1109/ACCESS.2020.3043719.

[4] S. Gupte, Paul Infant Teenu Mohandas and J. M. Conrad, ”A survey of quadrotor Unmanned Aerial Vehicles,” 2012 Proceedings of IEEE Southeastcon, 2012, pp. 1-6, doi: 10.1109/SECon.2012.6196930.

[5] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath and P. Tabuada, ”Control Barrier Functions: Theory and Applications,” 2019 18th European Control Conference (ECC), 2019, pp. 3420-3431, doi: 10.23919/ECC.2019.8796030.