Collision-Free Trajectory Tracking for Quadrotor UAVs: Closed-Form versus Optimization-Based Controllers

Aiza Batool
Department of Computer, Control and Management Engineering
Sapienza University of Rome
Rome, Italy
batool@diag.uniroma1.it

Abstract—This paper presents a comparative study of closed-form and optimization-based controllers for safe trajectory tracking in quadrotor UAVs. The proposed approach integrates control barrier functions (CBF) with an elliptical safety boundary to ensure obstacle avoidance. A closed-form controller is developed using CBFs and validated with elliptical and circular safety boundaries. This controller is then compared against optimization-based methods, specifically MPC with CBF constraints with an elliptical boundary. Two MPC variants are considered: one with distance constraints (MPC-DC) and another incorporating CBF constraints (MPC-CBF). The simulation results show the trade-offs between trajectory accuracy, computational efficiency, and adaptability in dynamic environments. The findings show that the CBF-based controller with an elliptical boundary achieves reliable and safe real-time tracking, while MPC-based methods offer predictive benefits at a higher computational cost.

Index Terms—Quadrotors, Control Barrier Functions, Trajectory Tracking, Model Predictive Control, Collision Avoidance, Elliptical Boundary.

I. Introduction

Aerial robotics has seen substantial growth, with Unmanned Aerial Vehicles (UAVs) now essential in sectors such as precision farming, surveillance, disaster response, and environmental observation. Among the different UAV platforms, quadrotors are widely used due to their mechanical simplicity, maneuverability, and effectiveness in navigating and controlling motion in three dimensional spaces [1]. Despite these advantages, the quadrotor's nonlinear, underactuated dynamics and limited onboard computation pose challenges for autonomous navigation. A key task is to design a controller that enables collision-free trajectory tracking—ensuring accurate trajectory following while maintaining safety around environmental obstacles.

This paper addresses this problem by comparing two prominent control approaches:

- A closed-form controller using Control Barrier Functions (CBFs), incorporating an Elliptical Boundary (EB) around the UAV to filter unsafe tracking commands.
- An optimization-based controller, specifically Model Predictive Control (MPC), implemented with distance constraints (MPC-DC) and CBF constraints (MPC-CBF) [2], [3].

CBFs offer a formal framework for enforcing safety through forward set invariance, enabling the system to remain within a safe set by constraining the control input [4]. In the closed-form controller, CBFs filter the nominal tracking command to prevent unsafe maneuvers. The elliptical boundary further enhances safety by modeling the UAV's footprint more accurately, facilitating navigation through cluttered and narrow environments.

In contrast, MPC optimizes control actions over a finite prediction horizon, balancing tracking objectives with obstacle avoidance. MPC-DC incorporates geometric safety margins, while MPC-CBF embeds safety constraints directly using barrier functions [2], [5]. Although effective, these methods increase computational load and may challenge real-time feasibility in embedded UAV systems. CBF-based controllers offer efficient safety enforcement via constraint filtering over nominal laws like PD [2], but they lack predictive planning and may be limited in dynamic scenarios, where feasible control inputs are not always easy to find. In contrast, MPC enables future-aware control with constraint handling [6], but it is computationally demanding for UAVs and often prone to numerical problems such as initialization sensitivity and reduced explicability. Prior studies often investigate these approaches independently, without clearly quantifying the trade-offs Closed-form controllers, while not always available, offer fast, numerically robust, and predictable performance. Optimization-based methods, though easier to formulate for complex scenarios, can be less transparent and computationally intensive. This study fills this gap by comparing a trajectory tracking controller with CBF and an elliptical boundary filter against two predictive MPC variants—MPC-DC and MPC-CBF—under shared UAV scenarios. This paper presents simulation results in static and dynamic environments to compare the two control strategies. The metrics used to quantify performance include mean absolute error (MAE), mean squared error (MSE), decision-making time, execution time, and minimum obstacle clearance.

Ultimately, this work contributes to a comparative analysis of closed-form and optimization-based control strategies for safe quadrotor trajectory tracking, highlighting the benefits of integrating geometry-aware safety constraints into real-time UAV control. The remainder of the paper is organized as follows: Preliminaries are introduced in Section II, Proposed Methods are discussed in Section III, Results and Discussion are provided in Section IV, and Conclusion is presented in Section V.

II. Preliminaries

A. Quadrotor Dynamics

A quadrotor is an underactuated aerial vehicle characterized by six degrees of freedom (DOF), controlled using four independent inputs: total thrust T and the torques $\tau_\phi, \tau_\theta, \tau_\psi$ along roll, pitch, and yaw, respectively. The state vector of the system is defined as:

$$\xi = [x, y, z, \dot{x}, \dot{y}, \dot{z}, \phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}]^T$$

with corresponding control input:

$$u = [T, \tau_\phi, \tau_\theta, \tau_\psi]^T$$

Under small-angle and quasi-hovering assumptions, the translational dynamics simplify to:

$$\ddot{x} = -\frac{T}{m}(\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi) \quad (1)$$

$$\ddot{y} = -\frac{T}{m}(\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi) \quad (2)$$

$$\ddot{z} = g - \frac{T}{m}\cos\phi \cos\theta \quad (3)$$

The angular dynamics can be expressed as:

$$\ddot{\phi} = \frac{\tau_\phi}{I_x}, \quad \ddot{\theta} = \frac{\tau_\theta}{I_y}, \quad \ddot{\psi} = \frac{\tau_\psi}{I_z} \quad (4)$$

This simplified formulation is commonly used in UAV control studies for system design and testing [7], [8]. A typical approach for generating tracking input is the Proportional-Derivative (PD) controller. In the x-axis, the control input is given by:

$$u^* = \ddot{x}_d + k_d(\dot{x}_d - \dot{x}) + k_p(x_d - x) \quad (5)$$

where x_d, $\dot{x}_d$, and $\ddot{x}_d$ represent the target position, velocity, and acceleration, respectively, and k_p, $k_d > 0$ denote controller gains.

B. Control Barrier Functions (CBFs)

Control Barrier Functions (CBFs) provide a formal framework to ensure system safety by preserving the forward invariance of a predefined safe set $\mathcal{C}$:

$$\mathcal{C} = \{x \in \mathbb{R}^n \mid h(x) \geq 0\} \quad (6)$$

Given a control-affine dynamic system:

$$\dot{x} = f(x) + g(x)u \quad (7)$$

the derivative of the barrier function $h(x)$ is computed as:

$$\dot{h}(x) = L_f h(x) + L_g h(x) u \quad (8)$$

A function $h(x)$ qualifies as a valid CBF if there exists an extended class-$\mathcal{K}$ function $\alpha(\cdot)$ such that:

$$L_f h(x) + L_g h(x) u + \alpha(h(x)) \geq 0 \quad (9)$$

The above condition defines the set of inputs that keep the system's evolution within the boundaries of the safe set $\mathcal{C}$ [4], [9]. CBFs have been successfully applied in robotics and UAV safety applications [10].

In this work, CBFs are integrated in two ways:

- As a safety layer over the nominal PD controller in the closed-form approach.
- As inequality constraints in the MPC-CBF formulation to ensure real-time safe trajectory tracking.

III. Proposed Methods

This section presents the proposed control strategies for collision-free trajectory tracking. It includes the design of a closed-form CBF-based controller with elliptical and circular safety boundaries, and two optimization-based controllers: MPC with Distance Constraints and MPC with embedded Control Barrier Functions.

A. Trajectory Tracking with CBF: Elliptical vs Circular Boundaries

The goal of trajectory tracking with Control Barrier Functions (CBFs) is to ensure safety while following a reference trajectory. This section presents a geometry-aware CBF-based tracking controller that compares the traditional circular boundary (CB) with a elliptical boundary (EB). The EB captures the actual footprint of the quadrotor, reducing unnecessary safety margins and improving navigation through narrow passages [11].

1) Geometric Modeling with Elliptical Boundary: To represent the quadrotor's body shape in the 2D x–z plane, we define an elliptical region centered at its center of mass:

$$({}^Bx, {}^Bz)\, M \begin{pmatrix} {}^Bx \\ {}^Bz \end{pmatrix} = 1, \quad M = \begin{pmatrix} \frac{1}{a^2} & 0 \\ 0 & \frac{1}{c^2} \end{pmatrix} \quad (10)$$

Using a rotation matrix $R(\theta)$ and translation to inertial coordinates, the ellipse rotates with the UAV's heading:

$$(O - \bar{P})^T R(\theta) M R^T(\theta)(O - \bar{P}) = 1$$

This transformation provides tighter obstacle negotiations in cluttered environments. In this work, we consider quasi-hovering flight conditions with small pitch angles,i.e., $\theta \approx 0$, so the ellipse remains aligned with the inertial frame.

2) Barrier Function and Safety Set: To enforce collision avoidance, we construct a CBF that integrates both position and velocity terms:

$$h_1 = (O - \bar{P})^T M (O - \bar{P}) + \mu (O - \bar{P})^T M \dot{O} > \delta \quad (11)$$

In this formulation, $O \in \mathbb{R}^2$ is the quadrotor's position in the inertial frame, $\bar{P}$ is the obstacle center, $M \in \mathbb{R}^{2\times 2}$ defines the shape of the elliptical boundary, $\dot{O}$ is the velocity of the quadrotor, μ is a tuning parameter that balances position and velocity influence, and δ is the minimum safety margin from the obstacle.

Its time derivative is:

$$\dot{h}_1 = 2(O-\bar{P})^T M\dot{O} + \mu\dot{O}^T M\dot{O} + \mu(O-\bar{P})^T Mu \quad (12)$$

The system is considered safe if:

$$\dot{h}_1 + \alpha h_1 \geq 0$$

This constraint ensures that the control input u maintains the state inside the safe set $\mathcal{C} = \{x : h_1(x) \geq 0\}$, thus preserving forward invariance.

3) Control Law with Projection Operators: The nominal tracking input is defined as:

$$u^\star = \ddot{O}_d + k_d(\dot{O}_d - \dot{O}) + k_p(O_d - O)$$

To ensure safety during operation, we employ a projection-based filtering mechanism. The control input is adjusted depending on the current system state: The projection-based filtering mechanism is inspired by the approach proposed in [12].

$$u = \begin{cases} u^\star, & \text{if } (O,\dot{O}) \in \mathcal{D}_{\text{track}}(t) \\ -\frac{2}{\mu}\Pi_M\dot{O} + \Pi_M^\perp u^\star, & \text{if } (O,\dot{O}) \in \mathcal{D}_\perp(t) \end{cases} \quad (13)$$

The projection matrices are:

$$\Pi_M = M(O-\bar{P})[(O-\bar{P})^T M^2(O-\bar{P})]^{-1}(O-\bar{P})^T M \quad (14)$$

$$\Pi_M^\perp = I - \Pi_M \quad (15)$$

Here, the state space is partitioned into two regions:

- $\mathcal{D}_{\text{track}}$: where $h_1 \geq \delta$, and nominal control is safe.
- $\mathcal{D}_\perp$: where $h_1 < \delta$, and filtering is required to maintain safety.

Substituting the filtered control input into the expression for $\dot{h}_1$ and simplifying, we get:

$$\dot{h}_1 = 2(O-\bar{P})^T M\dot{O} + \mu\dot{O}^T M\dot{O} + \mu(O-\bar{P})^T Mu \quad (16)$$

$$= \mu\dot{O}^T M\dot{O} + 2(O-\bar{P})^T M\dot{O} - 2(O-\bar{P})^T M\Pi_M\dot{O} \quad (17)$$

Since projection ensures $2(O-\bar{P})^T M\Pi_M\dot{O} = 2(O-\bar{P})^T M\dot{O}$, this simplifies to:

$$\dot{\bar{h}}_1 = \mu\dot{O}^T M\dot{O} \geq 0 \quad (18)$$

This shows that $\bar{h}_1$ is monotonically non decreasing, meaning that when the system enters the filtering region $\mathcal{D}_\perp(t)$, the quadrotor is prevented from approaching the obstacle further. Therefore, the projection-based safety controller guarantees collision avoidance while maintaining motion along the path.

B. Transition to Optimization-Based Controllers

While the trajectory tracking controller with CBF incorporating an Elliptical Boundary (EB) is computationally efficient and effective, the most popular approach for handling complex and dynamic scenarios relies on solving constrained optimization problems. Furthermore, closed-form solutions are not always easy to obtain, though in this work a comparison is possible. This subsection introduces optimization-based control strategies to ensure collision-free trajectory tracking in UAVs. We compare two methods: Model Predictive Control with Distance Constraints (MPC-DC) and Model Predictive Control with embedded Control Barrier Functions (MPC-CBF). These approaches compute control actions by solving constrained optimization problems over a finite horizon, while incorporating system dynamics, control limits, and safety constraints [6], [13].

1) NMPC Formulation and Discretized Dynamics: The Nonlinear Model Predictive Control (NMPC) approach aims to follow a desired trajectory while satisfying the nonlinear dynamics of the quadrotor and maintaining safety. At each time step, it solves a constrained nonlinear optimization problem over a prediction horizon N with sampling time T.

The system dynamics are described in a nonlinear state-space format:

$$x_{k+1} = f(x_k, u_k) \quad (19)$$

where $x_k = [x, \dot{x}, z, \dot{z}, \theta, \dot{\theta}]^T$ denotes the system state and $u_k = [U_1, U_2]^T$ is the input vector consisting of torque and thrust. The system is discretized via Euler integration:

$$x_{k+1} = x_k + T \cdot f(x_k, u_k) \quad (20)$$

The NMPC optimization minimizes a weighted cost function

$$\min_{U_k, X_k} J_{\text{mpc}} = \sum_{i=1}^{N} \left(\|r_{i|k} - x_{i|k}\|_Q^2 + \|u_{i|k}\|_{R_1}^2 + \|\Delta u_{i|k}\|_{R_2}^2 \right), \quad (21)$$

where $r_{i|k}$ is the target reference state at step i, $x_{i|k}$ is the predicted state, and $\Delta u_{i|k} = u_{i|k} - u_{i-1|k}$ quantifies the control rate change.

2) MPC with Distance Constraints (MPC-DC): MPC-DC adapts the nonlinear MPC formulation by directly enforcing geometric safety margins through Euclidean distance constraints. Unlike the CBF-based MPC, which incorporates mathematical safety functions, MPC-DC uses explicit distance checks between the UAV and obstacles, reducing computational overhead while maintaining obstacle avoidance. The same NMPC formulation (20) and cost function (21) are used. To ensure safety, a distance constraint is added to maintain a minimum clearance $\delta = 0.4\,\text{m}$ between the quadrotor and obstacles:

$$g_k = \sqrt{(x_k - x_{\text{obs}})^2 + (z_k - z_{\text{obs}})^2} > \delta \quad (22)$$

where x_k, z_k denote the quadrotor position and $x_{\text{obs}}, z_{\text{obs}}$ denote the obstacle center. This constraint is applied for each predicted state over the horizon.

3) MPC with Control Barrier Functions (MPC-CBF): MPC-CBF extends the NMPC formulation by directly integrating Control Barrier Functions into the optimization problem to ensure real-time safety during trajectory tracking. This approach allows the quadrotor to avoid obstacles while still tracking a desired reference, even during aggressive maneuvers.

The safety constraint is imposed using a velocity-aware CBF condition formulated as:

$$g(x_k, p_{\text{obs}}) = -2(O - P_{\text{obs}})^T M \dot{O} - \mu \|\dot{O}\|^2 + \alpha(h_1) > 0 \tag{23}$$

where $O = [x_o, z_o]^T$ is the quadrotor position, $\dot{O}$ its velocity, and P_{obs} is the position of the obstacle. The matrix M defines the elliptical boundary, and μ scales the velocity influence.

This condition is evaluated at each predicted state over the horizon to dynamically enforce obstacle avoidance. It ensures that the quadrotor maintains a safety margin $\delta = 0.4\,\text{m}$, accounting for both current velocity and future trajectory alignment.

The same NMPC cost function (21) and discretized dynamics (20) are used.

This formulation enables smoother adaptive obstacle avoidance, especially effective in cluttered or dynamic environments, without compromising trajectory tracking accuracy.

IV. Results and Discussion

This section presents the simulation results evaluating the performance of the proposed control strategies. Comparisons are drawn between the closed-form CBF controller with elliptical and circular bounding boxes, and the optimization-based controllers (MPC-DC and MPC-CBF), under both static and dynamic obstacle scenarios. The key performance indicators include trajectory tracking accuracy, obstacle clearance, and computational efficiency.

A. Elliptical vs Circular Bounding Box

The closed-form CBF controller is evaluated using elliptical and circular safety boundaries under both static and dynamic obstacle conditions. Controller parameters were: safety margin $\delta = 0.2$, projection gain $\mu = 0.1$, position gains $k_p = 60$, $k_d = 10$, pitch gains $k_{\theta p} = 35$, $k_{\theta d} = 2$, altitude gains $k_{zp} = 5$, $k_{zd} = 5$, inertia $I_y = 1 \times 10^{-4}$, mass $m = 0.3$, and bounding box semi-axes (a, c): $(5, 3)$ for EB, $(5, 5)$ for CB.

Table I presents performance metrics. Figure 1 shows that EB enables smoother and safer trajectory tracking compared to CB. Figure 2 confirms EB consistently maintains tighter minimum clearance. Evaluation is based on MAE (Mean Absolute Error), MSE (Mean Squared Error), MSM (Minimum Safety Margin), and OS (Overshoot).

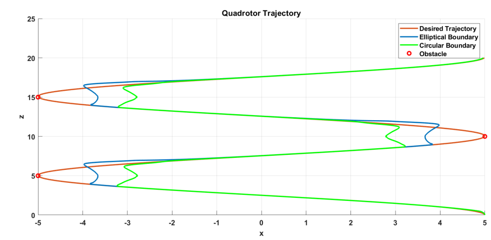


Fig. 1: Quadrotor trajectory (EB and CB) vs desired path. EB enables safer and smoother navigation.

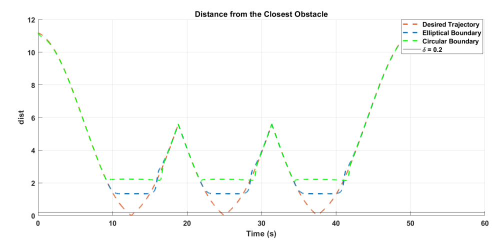


Fig. 2: Minimum Euclidean clearance from obstacles. EB maintains tighter safety margins.

Furthermore, we extend the EB model to 3D space by defining the UAV's position in x–y–z space using the matrix $M = \text{diag}(1/a^2, 1/b^2, 1/c^2)$, as shown in Figure 3. Key parameters: safety margin $\delta = 0.15$, pitch/roll gains $k_{\theta p} = k_{\phi p} = 100$, $k_{\theta d} = k_{\phi d} = 30$, and ellipsoid axes $a = 1$, $b = 2$, $c = 3$.

Figure 4 confirms that the controller preserves clearance even in 3D cluttered spaces. The ellipsoid defines the UAV's safety region used to assess tracking and obstacle avoidance.

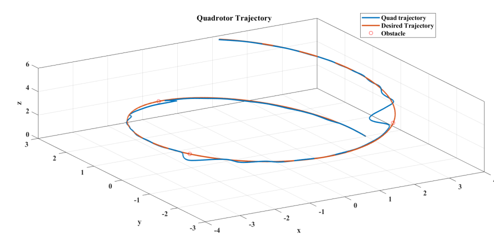


Fig. 3: 3D trajectory tracking with EB. The UAV avoids static obstacles and returns smoothly to the desired reference trajectory
.

B. MPC-DC vs MPC-CBF (Static Obstacles)

1) Simulation Setup: The MPC controllers (MPC-DC and MPC-CBF) were configured with the following parameters: safety margin $\delta = 0.4$, projection gain $\mu = 0.6$, prediction horizon $N = 15$, and sampling time $T_s = 0.1\,\text{s}$. State limits were: $\dot{x}, \dot{z} \in [-1.5, 1.5]\,\text{m/s}$, pitch angle $\theta \in [-0.5, 0.5]\,\text{rad}$, and pitch rate $\dot{\theta} \in [-0.1, 0.1]\,\text{rad/s}$. Control

TABLE I: Tracking Metrics Comparison: Elliptical vs Circular Boundary

Metric	EB	CB
MAE (m)	0.49	0.95
MSE (m^2)	0.40	1.39
MSM (m)	1.34	2.08
OS (m)	1.34	2.23

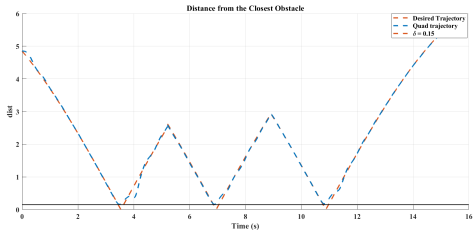


Fig. 4: Minimum Euclidean distance from obstacles. The quadrotor maintains a safe clearance above the threshold $\delta = 0.15$, ensuring successful obstacle avoidance.

inputs were constrained as $U_1 \in [-0.5, 0.5]$, $U_2 \in [-4, 20]$, control rate changes as $\Delta u \in [-0.1, 0.1]$.

The weight matrices are tuned as: $Q = \text{diag}(q_w \cdot [x_w, z_w, \theta_w])$, $R = (1 - q_w - r_{1w}) \cdot [0.4, 0.6]$, and $R_1 = r_{1w} \cdot [0.4, 0.6]$, with $q_w = 0.8024$, $r_{1w} = 0.1813$, $x_w = 0.9997$, $z_w = 0.8708$, $\theta_w = 0.2295$.

All NMPC-based controllers (MPC-DC and MPC-CBF) are implemented in MATLAB using CasADi with the IPOPT solver [14]. The solver operated in real-time with a sampling period of $T_s = 0.1\,\text{s}$ a maximum of 20 iterations, and a convergence tolerance of 10^{-8}. Obstacle positions were passed as dynamic parameters to the optimizer and updated at each control step. At every iteration, only the first control input of the computed optimal sequence was applied, before the horizon shifts forward for the next step.

Table II summarizes the performance metrics comparing the closed-form CBF controller, MPC-DC, and MPC-CBF. While CBF-Tracking achieves the fastest execution and decision times, it shows higher tracking errors. In contrast, MPC-DC achieves the lowest tracking error by explicitly enforcing distance constraints, though at the cost of sharp trajectory deviations.

TABLE II: Controller Comparison: CBF vs MPC Variants

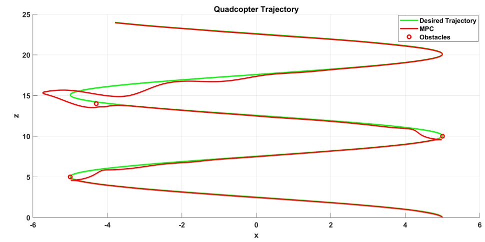
Metric	CBF (Closed-Form)	MPC-DC	MPC-CBF
MAE (m)	0.49	0.26	0.42
MSE (m^2)	0.40	0.13	0.34
Decision Time (s)	0.0003	0.019	0.053
Execution Time (s)	35.504	37.948	42.876

Figures 5a and 5b present the trajectory tracking results under static obstacles. MPC-DC produces sharp deviations to avoid obstacles, compromising smoothness, while MPC-CBF ensures smoother avoidance and better alignment with the desired trajectory.

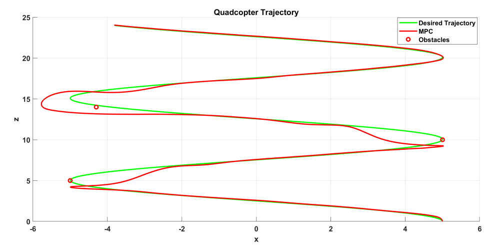
Figures 6a and 6b show the minimum Euclidean distance from obstacles. MPC-DC occasionally drops near the clearance threshold, while MPC-CBF consistently satisfies safety margins, highlighting its conservative and robust behavior.

C. Trajectory Tracking in Dynamic Obstacle Environments

This section evaluates controller performance with multiple moving obstacles placed along the reference trajectory. The same parameters from the static case were used for both MPC and closed-form controllers. The quadrotor adjusts its motion in real time to maintain trajectory accuracy while ensuring safe avoidance.

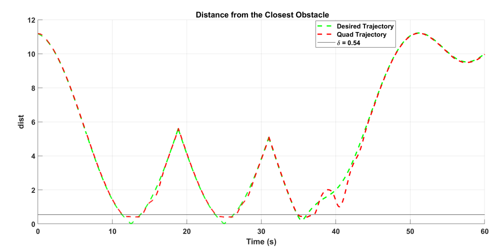


(a) Tracking with MPC-DC.



(b) Tracking with MPC-CBF.

Fig. 5: Trajectory tracking under static obstacles. (a) MPC-DC navigates obstacles via direct deviations. (b) MPC-CBF achieves smoother avoidance.



(a) MPC-DC: Minimum Euclidean distance from obstacles. While the clearance threshold $\delta = 0.4\,\text{m}$ is mostly maintained, violations occur at certain time steps, indicating suboptimal avoidance.

(b) MPC-CBF: Obstacle clearance remains consistently above $\delta = 0.4\,\text{m}$, even during dynamic maneuvers.

Fig. 6: Minimum Euclidean distance from the closest obstacle. MPC-CBF outperforms MPC-DC in maintaining consistent safety margins.

Figure 7 shows that the closed-form CBF controller with an Elliptical Boundary (EB) provides smoother navigation and reduced overshoot compared to the Circular Boundary (CB). Among the optimization-based controllers, MPC-DC (Figure 8) struggles with obstacle motion, risking collisions. In contrast, MPC-CBF (Figure 9) adapts effectively to moving obstacles, maintaining safe separation and accurate tracking.

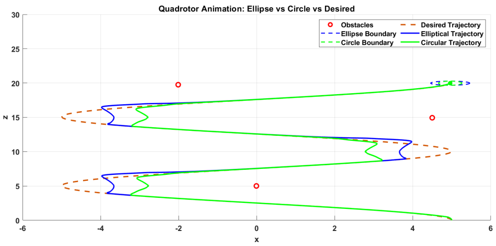


Fig. 7: Trajectory tracking with CBFs under Elliptical and Circular Boundaries amid dynamic obstacles. EB provides more efficient and safer avoidance compared to CB.

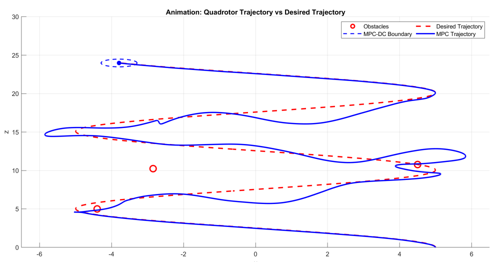


Fig. 8: Trajectory tracking with MPC-DC. The trajectory intersects with obstacles, indicating limited safety performance.

V. Conclusion

Safety-critical control in robotic systems remains a fundamental challenge, particularly for real-time quadrotor navigation in cluttered environments. This work addressed trajectory tracking with obstacle avoidance using Control Barrier Functions (CBFs), comparing two main approaches: a closed-form trajectory tracking controller with CBFs and optimization-based controllers using Model Predictive Control (MPC).

The closed-form controller used an elliptical boundary (EB) to represent the UAV's shape, providing a tighter, more efficient safety region than conventional circular boundary (CB).

In contrast, optimization-based controllers, such as MPC-DC and MPC-CBF, demonstrated stronger predictive abilities. MPC-DC achieved the lowest tracking error using distance constraints, while MPC-CBF offered smoother, safer obstacle avoidance at the cost of increased computational load.

In execution and decision time, the closed-form CBF controller outperformed both MPC variants, making it highly suitable for resource-constrained platforms.

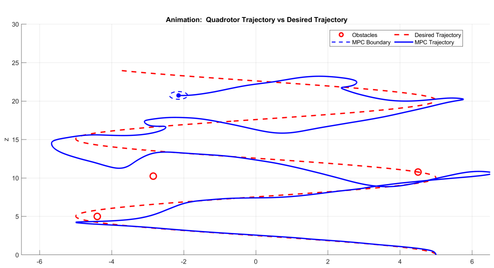


Fig. 9: Trajectory tracking with MPC-CBF. The controller adapts to obstacle movement while maintaining safe and accurate tracking.

This comparative study highlights that each controller has distinct strengths: the trajectory tracking controller with CBF-EB offers a simple yet robust solution, while MPC-CBF is preferable for dynamic, high-risk environments with onboard computing.

Future research may relax the small-angle assumption by incorporating full orientation dynamics into the safety model. Another direction is to integrate learning-based strategies, such as Reinforcement Learning (RL) or adaptive CBFs, to enhance adaptability in uncertain environments. Finally, real-world UAV testing and multi-agent deployments will be key to evaluating practical performance.

References

[1] K. Cook, "The silent force multiplier: The history and role of uavs in warfare," 2007 IEEE Aerospace Conference, pp. 1–7, 2007.

[2] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," IEEE Transactions on Automatic Control, vol. 62, no. 8, pp. 3861–3876, 2017.

[3] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in 53rd IEEE Conference on Decision and Control, 2014, pp. 6271–6278.

[4] P. Wieland, "Constructive safety using control barrier functions," in Proceedings of the IFAC Conference on Analysis and Design of Hybrid Systems, vol. 7, 08 2007, pp. 462–467.

[5] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in 2019 18th European Control Conference (ECC), 2019, pp. 3420–3431.

[6] J. Richalet, A. Rault, J. Testud, and J. Papon, "Model predictive heuristic control," Automatica, vol. 14, no. 5, pp. 413–428, 1978.

[7] A. Ailon, "Closed-form feedback controllers for set-point and trajectory tracking for the nonlinear model of quadrotor helicopters," IFAC Proceedings Volumes, vol. 45, no. 13, pp. 375–380, 2012, 7th IFAC Symposium on Robust Control Design.

[8] A. Ailon and S. Arogeti, "Closed-form nonlinear tracking controllers for quadrotors with model and input generator uncertainties," Automatica, vol. 54, pp. 317–324, 2015.

[9] S. Prajna and A. Jadbabaie, "Safety verification of hybrid systems using barrier certificates," in International Workshop on Hybrid Systems: Computation and Control, 2004, pp. 477–492.

[10] F. Ferraguti, C. Talignani Landi, S. Costi, M. Bonfè, S. Farsoni, C. Secchi, and C. Fantuzzi, "Safety barrier functions and multi-camera tracking for human–robot shared environment," Robotics and Autonomous Systems, vol. 124, p. 103388, 2020.

[11] J. Park and Y. Kim, "Real-time collision avoidance control of quadrotor using ellipsoid as a bounding box," in 2016 European Control Conference (ECC), 2016, pp. 2152–2157.

[12] A. Cristofaro, M. Ferro, and M. Vendittelli, "Safe trajectory tracking using closed-form controllers based on control barrier functions," in 2022 IEEE 61st Conference on Decision and Control (CDC), 2022, pp. 3329–3334.

[13] B. Kouvaritakis and M. Cannon, Model Predictive Control. Switzerland: Springer International Publishing, 2016, vol. 38.

[14] J. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi – a software framework for nonlinear optimization and optimal control," Mathematical Programming Computation, vol. 11, no. 1, pp. 1–36, 2019.