

# Three Dimensional Optimal Path Generation and Tracking for Quadrotors in Dynamic Environment

Hariharan R.<sup>1</sup>, Siddarth Yadla<sup>2</sup> and Sikha Hota<sup>3</sup>

**Abstract**—This paper presents a collision avoidance algorithm for quadrotors in a three-dimensional (3D) cluttered environment filled with dynamic obstacles, including a path re-planning algorithm to rejoin with its original trajectory in optimal time. The algorithms are based on a geometrical approach, accomplished by changing the heading and the pitch angles of the quadrotor simultaneously while keeping its speed constant. The novelty of the proposed work is that the algorithm ensures a smooth path generation for both the collision avoidance and re-planning maneuvers for the quadrotor with very low tracking error, including optimality in path length. To validate the proposed algorithm, the generated path is tracked using a dynamic model of a quadrotor using a Lyapunov-based backstepping controller in the presence of external disturbances.

## I. INTRODUCTION

Quadrotors have been gaining a lot of momentum due to their applications in a large spectrum of human lives. They are popularly used because of ease of manufacturing, ability to lift off heavy payload, and inherent capability of automation.

For tasks to be completed successfully in a cluttered environment, it is necessary for quadrotors to plan their path, avoiding obstacles, both static and dynamic, on the way. Path planning algorithms are generally classified into two categories, global and local. The global planning algorithms provide solutions considering the information about the whole environment beforehand. So this requires a considerable amount of memory for storage and computation. The local planning algorithms, on the other hand, are reactive. They only provide solutions reacting to local information acquired from the sensors of quadrotors, which significantly reduces the computational requirements and makes it efficient to implement in real-time. But the drawback of local planning algorithms is that they may not provide global optimal solutions. Hence, a hybrid approach that plans the path before the task using a global planner and avoids obstacles on the way using a local planner is preferable [1].

There have been a variety of approaches proposed over the years for collision avoidance. Some of the famous approaches include A\* [2], Delaunay Triangulation [3], Rapidly-exploring Random Trees (RRT) [4] and Potential

Field approaches [5]. A comprehensive survey of these methods is presented in [6]. These methods, even though they provide optimal solutions, require high computational memory and time. Geometric-based algorithms have been widely used as they require low memory and use low computation time, making them simple, efficient, and easy to implement. These algorithms use the geometric relation between an unmanned aerial vehicle (UAV) and an obstacle to calculate the optimal trajectory to evade obstacles. Many geometric-based collision avoidance algorithms exist in the existing literature, such as [7], [8], [9], [10], [11], [12], [13], [14]. Each of these methods has assumed that the change in velocity of the UAV is instantaneous, which demands very high lateral acceleration and hence, impractical. A novel practical collision avoidance approach called 3-dimensional velocity obstacle (3DVO) method, achieved by using a pure turning maneuver of the UAV in a chosen avoidance plane, was proposed in [15]. Since the avoidance plane is selected from a finite set, the proposed solution does not guarantee global optimality.

In this paper, a collision avoidance strategy and a path re-planning algorithm for quadrotors are proposed to avoid dynamic obstacles and then return to its original trajectory. The collision avoidance algorithm is based on the geometric method that provides an optimal solution by minimizing the change in the velocity vector. In addition, a novel path-replanning algorithm which makes the quadrotor maneuver back to its original course after successful collision avoidance has been presented. The algorithm presented in this paper changes the heading and elevation of the quadrotor simultaneously for maneuvers, keeping its speed constant. In addition, the proposed work ensures a smooth transition of the velocity vector by maneuvering the quadrotor along circular curves in three-dimensional (3D) space, making it implementable in real-world applications with very low tracking error.

Along with the path planning, a path tracking algorithm [16], [17] that guarantees asymptotic convergence of the UAV with the desired path is also presented here. In order to solve the trajectory tracking problem, many nonlinear controllers like Lyapunov-based control, Sliding mode control [18], [19], backstepping control [20], [21], feedback linearization control [22], and model predictive control [23], [24] exist in the literature. These control techniques take into account the nonlinearities of the dynamics and assure the stability of the nonlinear system [25]. In this work, a Lyapunov-based backstepping control with integral action and Saturation function [26] has been used to track the path

<sup>1</sup>Hariharan R. is with the Department of Aerospace Engineering, Indian Institute of Technology Kharagpur, Kharagpur, West Bengal 721302, India rharil999@iitkgp.ac.in

<sup>2</sup>Siddarth Yadla is with the Department of Aerospace Engineering, Indian Institute of Technology Kharagpur, Kharagpur, West Bengal 721302, India siddarthiyadla@iitkgp.ac.in

<sup>3</sup>Sikha Hota is with the Department of Aerospace Engineering, Indian Institute of Technology Kharagpur, Kharagpur, West Bengal 721302, India sikhahota@aero.iitkgp.ac.in

derived from the collision avoidance algorithm efficiently with very low tracking error.

The rest of the paper is organized as follows: Section II describes the problem formulation. Section III-A deals with the collision detection algorithm. In section III-B, the collision avoidance approach is presented. Section III-C discusses the novel path replanning approach for the quadrotor to reach the original trajectory. Section IV presents the dynamic model of the quadrotor and the controller design to track the generated path. Section V demonstrates the simulation results. Finally, section VI concludes the paper.

## II. PROBLEM FORMULATION

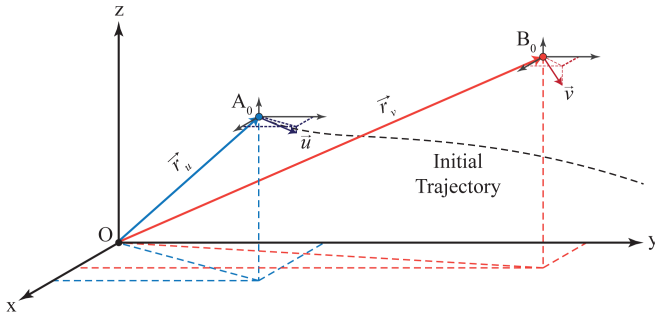


Fig. 1. Position vectors of quadrotor  $A_0$ , obstacle  $B_0$  and the goal  $G$ .

Consider that a quadrotor is moving with constant speed along a trajectory generated using a global path planner. The quadrotor detects an intruder at position  $B_0$  moving with a constant velocity  $\vec{v}$ . Let the position and velocity of the quadrotor at this instant be  $A_0$  and  $\vec{u}$  respectively. All the vectors are measured in the inertial frame of reference with  $x, y, z$  coordinate axes (refer Fig. 1).

The quadrotor calculates that it would collide with the intruder if it advanced in the same direction [9]; hence, it becomes necessary to change its trajectory course.

We focus mainly on two objectives in this paper:

- 1) To develop a local path planning algorithm that not only avoids the collision with the intruder but also enables the quadrotor to rejoin with its initial trajectory.
- 2) To design a trajectory tracking algorithm that will ensure the convergence of the quadrotor with the generated path.

## III. PATH PLANNING

### A. Collision Detection

The collision detection technique is excerpted from [9]. The relative velocity of the quadrotor with respect to the intruder is given by  $\vec{v}_{rel} = \vec{u} - \vec{v}$  (see Fig. 2). The *minimum separation distance* vector,  $\vec{d}_{rel}$  between the UAV and the obstacle as defined in [9] can be calculated as

$$\vec{d}_{rel} = \frac{\vec{r} \cdot \vec{v}_{rel}}{|\vec{v}_{rel}|^2} \vec{v}_{rel} - \vec{r} \quad (1)$$

where,  $\vec{r} = \vec{r}_v - \vec{r}_u$ . It is proved in [9] that the collision between the UAV and the obstacle is certain if and only if

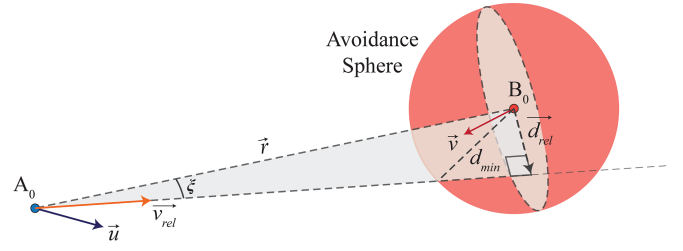


Fig. 2. Definition of *minimum separation distance* vector

the following conditions are satisfied:

$$|\vec{d}_{rel}| < d_{min} \quad \text{and} \quad \dot{r} < 0 \quad (2)$$

where  $d_{min}$  is the *minimum distance radius*, i.e., the radius of the avoidance sphere.

### B. Collision Avoidance

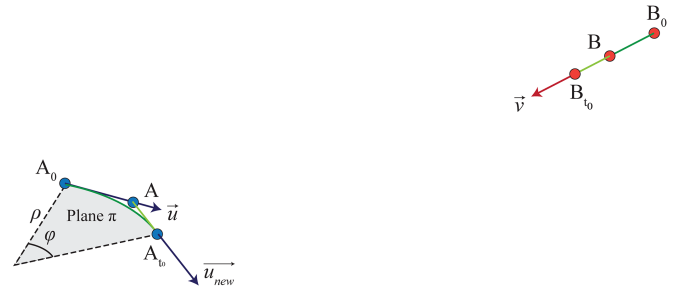


Fig. 3. Quadrotor turning maneuver along a circular path

Consider that the collision avoidance trajectory includes a circular turn with radius  $\rho$  for  $t_0$  seconds on an arbitrary plane,  $\pi$  followed by a straight line till the obstacle is evaded.  $\rho$  can be chosen according to the dynamics of the quadrotor. The quadrotor changes its velocity from  $\vec{u}$  to  $\vec{u}_{new}$  after the travel along the circular path (see Fig. 3(a)), and this change avoids the collision with the intruder. Let the angle subtended by the circular path towards the center be  $\varphi$ . The relation between  $\varphi$  and  $t_0$  is given as:

$$\varphi = \frac{|\vec{u}|t_0}{\rho} \quad (3)$$

Let the positions of the quadrotor and the intruder, after the quadrotor makes the circular turn, be  $A_{t_0}$  and  $B_{t_0}$ ,

respectively. Let us consider two arbitrary points,  $A$  on the initial trajectory of the quadrotor and  $B$  on the trajectory of the intruder. These points are chosen such that if the quadrotor moves from the point  $A$  with the velocity  $\vec{u}_{new}$  and the intruder moves from the point  $B$  with the velocity  $\vec{v}$ , they reach the points  $A_0$  and  $B_0$ , respectively after  $t$  seconds. It is assumed that the initial trajectory is a straight line in the vicinity of the current position of the quadrotor, i.e.,  $\vec{A_0A}$  has the same direction as  $\vec{u}$ . Using geometry, we can show that (Fig. 3(b))

$$t = \frac{AA_0}{|\vec{u}|} = \frac{A_0A}{|\vec{u}|} = \frac{\rho \tan(\varphi/2)}{|\vec{u}|} \quad (4)$$

$$\vec{r}_A = \vec{r}_u + \vec{u}.t \quad (5)$$

$$\vec{r}_B = \vec{r}_v + \vec{v}.(t_0 - t) \quad (6)$$

where  $\vec{r}_A$  and  $\vec{r}_B$  are the position vectors of the points  $A$  and  $B$ , respectively.

Now, we can redefine the problem as: "Given that the quadrotor and the obstacle are at virtual positions,  $A$  and  $B$ , with the velocities  $\vec{u}$  and  $\vec{v}$ , respectively, find the optimal velocity vector for the quadrotor, which will change instantaneously from its current direction to avoid collision and make an angle  $\varphi$  with  $\vec{u}$ ."

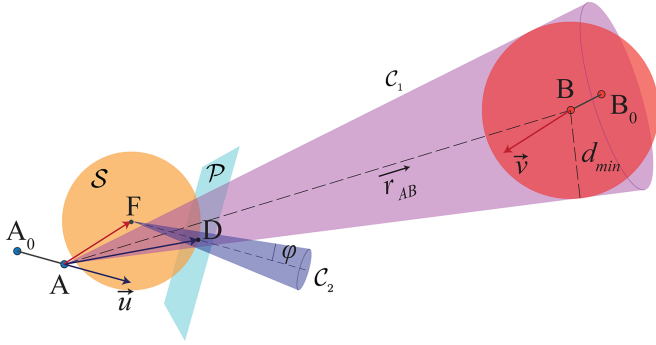


Fig. 4. Geometric visualization of collision avoidance approach

Consider the points  $D$  and  $F$  such that  $\vec{AD}$  has the same magnitude and direction of the relative velocity vector of the quadrotor,  $\vec{v}_{rel}$  and  $\vec{AF}$  has the same magnitude and direction of the vector  $-\vec{v}$  (refer fig. 4). The position vectors of the points,  $D$  and  $F$  are given by

$$\vec{r}_D = \vec{r}_A + \vec{u} - \vec{v} \quad (7)$$

$$\vec{r}_F = \vec{r}_A - \vec{v} = \vec{r}_u + \vec{u}.t - \vec{v} \quad (8)$$

Note that the vectors  $\vec{AD}$  and  $\vec{AF}$  are in meters whereas  $\vec{v}_{rel}$  and  $-\vec{v}$  are in meters per second.

The cone,  $C_1$  represents the collision cone. To avoid the collision,  $\vec{AD}$  should lie on or outside the collision cone. The center of the sphere  $S$  lies at  $F$ , and since  $\vec{FD} = \vec{u}$ , the radius is equal to  $|\vec{u}|$ . The significance of the sphere,  $S$  is that the vector joining the point,  $A$  to any point on the sphere  $S$ , represents a possible relative velocity vector of the quadrotor keeping its speed constant. The cone,  $C_2$  has its vertex fixed at  $F$  and  $\vec{FD}$  as its axis. The semi-vertical angle of the cone,  $C_2$  is equal to  $\varphi$ , and hence any vector

joining the point,  $F$  to any point on the cone will make an angle,  $\varphi$  with  $\vec{u}$ . Since the vertex of the cone,  $C_2$  intersects with the center of the sphere,  $S$ , their intersection will be a circle. Hence, to reduce the system further, the circle can be interpreted as an intersection between the sphere,  $S$ , and a plane,  $P$ . Let the center of the circle be the point,  $C$  and its position vector,  $\vec{r}_C$  can be calculated as:

$$\vec{r}_C = \vec{r}_F + (1 - \cos \varphi) \vec{u} \quad (9)$$

For simplifying the notations, we define two nomenclatures here:

1)  $v_i \equiv i^{th}$  component of any vector  $\vec{v}$

2)  $i_P \equiv i^{th}$  component of any position vector  $\vec{r}_P$

The equations of the plane,  $P$ , the sphere,  $S$  and the cone,  $C_1$  can be given as

$$P : u_x(x - x_C) + u_y(y - y_C) + u_z(z - z_C) = 0 \quad (10)$$

$$S : (x - x_F)^2 + (y - y_F)^2 + (z - z_F)^2 - |\vec{u}|^2 = 0 \quad (11)$$

$$C_1 : (|\vec{r}_{AB}|^2 - d_{min}^2)[(x - x_A)^2 + (y - y_A)^2 + (z - z_A)^2] - [x_{AB}(x - x_A) + y_{AB}(y - y_A) + z_{AB}(z - z_A)]^2 = 0 \quad (12)$$

where,  $\vec{r}_{AB} = \vec{r}_B - \vec{r}_A$  and  $x_{AB}, y_{AB}, z_{AB}$  are the  $x, y, z$  components of  $\vec{r}_{AB}$  respectively. Any point on the intersection of the sphere,  $S$ , the cone,  $C_1$ , and the plane,  $P$  is a possible solution and let us consider one such point be  $Q(x, y, z)$ . Since we are finding an optimal solution, to minimize the change in velocity vector, the point  $Q$  should lie close to the point,  $D$ . Let the function,  $f$  represent the Euclidean distance between the points  $Q$  and  $D$ .

$$f = \sqrt{(x - x_D)^2 + (y - y_D)^2 + (z - z_D)^2} \quad (13)$$

The problem is now a constrained nonlinear multi-variable optimization problem, described as

$$\min_{x, y, z, \varphi} f \text{ s.t. } S = 0, C_1 = 0, P = 0, \text{ and } \varphi > 0.$$

This problem can be solved numerically using the **fmincon** function or **MultiStart** function (for multiple local minimum solutions) in MATLAB. Once the solution is found, we can calculate  $t_0$  from (3) and  $\vec{u}_{new}$  can be calculated as

$$\vec{u}_{new} = [x - x_F \quad y - y_F \quad z - z_F]^T \quad (14)$$

where,  $x_F, y_F, z_F$  can be calculated from (4) and (8). The equation of the plane,  $\pi$  can be calculated as

$$\text{Plane } \pi : n_x(x - x_u) + n_y(y - y_u) + n_z(z - z_u) = 0 \quad (15)$$

where,  $n_x, n_y, n_z$  are the  $x, y, z$  components of  $\vec{n} = \vec{u} \times \vec{u}_{new}$ . The collision avoidance trajectory will be given by the circular path in the plane,  $\pi$  for  $t_0$  second, and then a straight path. The maneuver will end when the quadrotor reaches the point of the closest approach with the intruder (see Fig. 5) as the distance between them increases after this moment.

A solution cannot be found if the sphere,  $S$  lies entirely inside the cone,  $C_1$  and so, the proposed approach can be used if and only if the following condition holds:

$$\exists(x, y, z) \in C_1 : (x - x_F)^2 + (y - y_F)^2 + (z - z_F)^2 \leq |\vec{u}|^2$$

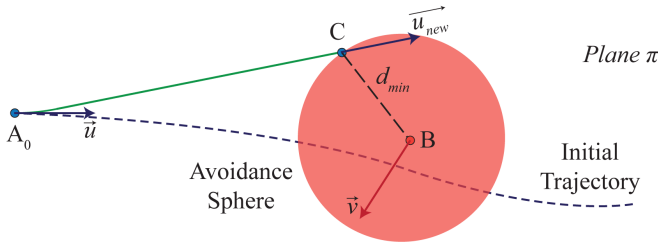


Fig. 5. Quadrotor at Point of Closest Approach (PCA)

In rare cases where the condition does not hold, a solution is possible if an intersection exists between the cone and the sphere, which is possible only with an increase in the speed of the quadrotor, which is beyond the scope of the present work. The method for this particular case was presented in [10] to provide a solution for heading, elevation, and speed change, but the change of velocity vector was instantaneous and hence impractical.

### C. Path Replanning

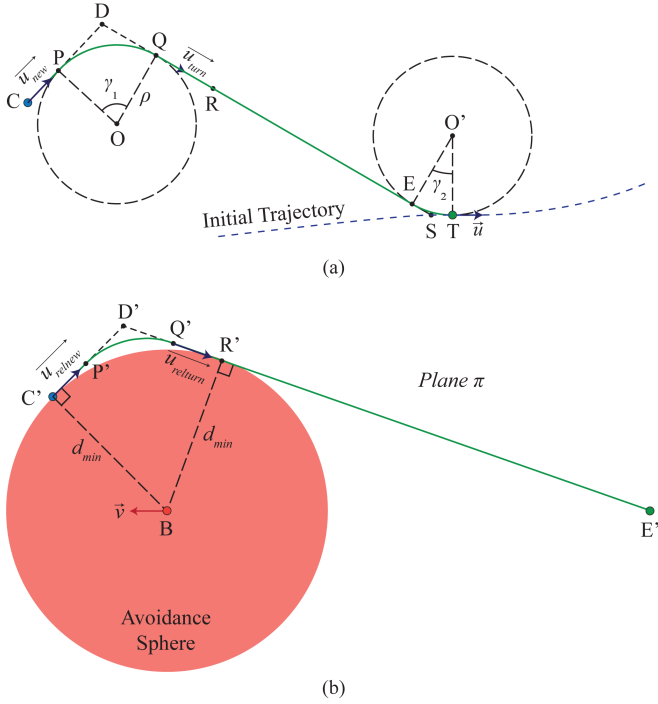


Fig. 6. (a) Global Frame (b) Relative Frame

The replanning maneuver includes two circular turns with a straight path connecting those two turns. Let the position of the quadrotor and the intruder at the point of the closest approach be  $C$  and  $B$ , respectively. Consider a point,  $T(x_T, y_T, z_T)$ , where the quadrotor will rejoin its initial trajectory, and the desired velocity of the quadrotor at  $T$  be  $\vec{u}$ . Let the time taken for the collision avoidance maneuver be  $t_C$  seconds. The replanned maneuver is given by the path  $CPQET$  in the plane  $\pi$ , which is the same plane described in the collision avoidance trajectory (see Fig. 6(a)).  $PQ$  and  $ET$  are circular arcs of radius,  $\rho$  and subtend angles,  $\gamma_1$ , and

$\gamma_2$ , with their centers,  $O$  and  $O'$ , respectively. The point,  $D$  is the intersection of the extended portions of the line segments,  $CP$ , and  $QE$ .

Let the time taken to reach  $P$  and  $Q$  be  $t_1$  and  $t_2$ , respectively. The position vectors for points,  $P$ ,  $Q$  and  $D$  can be calculated as

$$\vec{r}_P = \vec{r}_H + \vec{u}_{new} \cdot (t_1 - t_C) \quad (16)$$

$$\vec{r}_D = \vec{r}_P + \frac{\rho \tan(\gamma_1/2)}{|\vec{u}_{new}|} \vec{u}_{new} \quad (17)$$

$$\vec{r}_Q = \vec{r}_D + \frac{\rho \tan(\gamma_1/2)}{|\vec{u}_{turn}|} \vec{u}_{turn} \quad (18)$$

where  $\vec{u}_{new}$  is the velocity of the quadrotor after collision avoidance,  $\vec{u}_{turn}$  is the velocity of the quadrotor after the first circular turn.  $\vec{u}_{turn}$  is calculated as

$$\vec{u}_{turn} = |\vec{u}_{new}| \frac{\vec{r}_E - \vec{r}_D}{|\vec{r}_E - \vec{r}_D|} \quad (19)$$

Let the point of the intersection between the extended portions of the line segment,  $QE$ , and the initial trajectory be  $S$ . We assume that the initial trajectory is straight in the vicinity of point,  $T$ , i.e.,  $\vec{ST}$  has the same direction as  $\vec{u}$ . The position vectors of points,  $E$ , and  $S$  are calculated as

$$\vec{r}_S = \vec{r}_T - \frac{\rho \tan(\gamma_2/2)}{|\vec{u}|} \vec{u} \quad (20)$$

$$\vec{r}_E = \vec{r}_S - \frac{\rho \tan(\gamma_2/2)}{|\vec{u}_{turn}|} \vec{u}_{turn} \quad (21)$$

The angles,  $\gamma_1$ , and  $\gamma_2$  are given by

$$\gamma_1 = \cos^{-1} \left( \frac{\vec{u}_{new} \cdot \vec{u}_{turn}}{|\vec{u}_{new}| |\vec{u}_{turn}|} \right) \quad (22)$$

$$\gamma_2 = \cos^{-1} \left( \frac{\vec{u} \cdot \vec{u}_{turn}}{|\vec{u}| |\vec{u}_{turn}|} \right) \quad (23)$$

To simplify further, let us consider the point,  $E$ , to be the local goal of the quadrotor. The path is given by  $C'P'Q'E'$  transforming it to the relative frame of the intruder. In this frame of reference, the circular path  $PQ$  will be transformed into a trochoid,  $P'Q'$  (see Fig. 6(b)). The position vectors of the points  $Q'$  and  $E'$  can be calculated as

$$\vec{r}_{Q'} = \vec{r}_Q - \vec{v} \cdot t_2 \quad (24)$$

$$\vec{r}_{E'} = \vec{r}_E - \vec{v} \cdot t_f \quad (25)$$

Let the total time taken to reach the goal be  $t_f$ . The relations between  $t_1, t_2$  and  $t_f$  are given as

$$t_2 = t_1 + \frac{\rho \gamma_1}{|\vec{u}_{new}|} \quad (26)$$

$$t_f = t_2 + \frac{|\vec{r}_E - \vec{r}_Q|}{|\vec{u}_{turn}|} \quad (27)$$

For an optimal maneuver, a path is considered such that  $Q'E'$  is tangential to the avoidance sphere at the point,

$R'$ . Hence, using trigonometry in  $\triangle BR'E'$ , the following equation should hold:

$$|\vec{r}_B - \vec{r}_{E'}| \sin \left( \cos^{-1} \left( \frac{(\vec{r}_B - \vec{r}_{E'}) \cdot (\vec{r}_{Q'} - \vec{r}_{E'})}{|\vec{r}_B - \vec{r}_{E'}| |\vec{r}_{Q'} - \vec{r}_{E'}|} \right) \right) = d_{min} \quad (28)$$

Consider a function,  $f_r$  that represents the distance traveled by the quadrotor for the replanned maneuver, given as

$$f_r = \rho\gamma_1 + |\vec{r}_E - \vec{r}_Q| + \rho\gamma_2 \quad (29)$$

For optimal solution, we minimize the function  $f_r$ , i.e.,

$$\min_{x_T, y_T, z_T} f_r \quad (30)$$

s.t. (20) – (28), and  $\gamma_1, \gamma_2 > 0$ .

This optimization problem can be solved numerically using the **fmincon** function or **MultiStart** function (for multiple local minimum solutions) in MATLAB. Once the solution is found, we can get the points in the path  $CPQET$ .

It is possible for the optimization problem to have no solution, and hence, we also propose an alternative approach. For this method, the constraint that line segment  $Q'E'$  should be tangential to the avoidance sphere is removed. However, there is a possibility for a previously evaded intruder to appear in this replanned maneuver. To tackle this issue, a parameter, *safe replanning radius* ( $d_{safe}$ ), is proposed for the quadrotor, which is defined as the minimum distance between the quadrotor and the intruder after which the quadrotor starts the circular turn, as shown in Fig. 7.

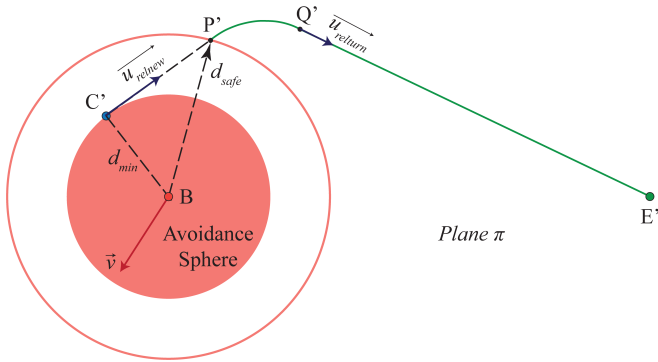


Fig. 7. Replanning maneuver - Approach II (Relative Frame)

With this parameter, the position vector of the point,  $P$  can be calculated as

$$\vec{r}_P = \vec{r}_C + \frac{\sqrt{d_{safe}^2 - d_{min}^2}}{|\vec{u}_{renew}|} \vec{u}_{renew} \quad (31)$$

Once  $\vec{r}_P$  is calculated, the other variables of the problem (30) can be solved to get the replanned path. Even in this approach, there are very few cases where an intruder, previously evaded, may appear on the way. This issue can be prevented either by turning along the path with a radius of curvature more than the *minimum turning radius* of the quadrotor or increasing the value of the *safe replanning radius*.

The combination of the collision avoidance and the replanning algorithms will provide the desired smooth path starting from and ending to the initial trajectory optimally. The generated path for the quadrotor will be a collection of circle-straight-circle (CSC) paths. The path tracking algorithm is discussed in the next section.

#### IV. TRAJECTORY TRACKING

This section aims at designing a controller that ensures the convergence of the quadrotor with the desired path generated by the path planning algorithm. The problem is described as follows,

- The distance between the center of mass of the quadrotor and the desired position on the generated path should be zero.
- The angle between the velocity vector of the quadrotor and the vector tangent to the desired trajectory should be zero.

##### A. Dynamic modeling of Quadrotor

In order to understand the position and attitude of the quadrotor in space, following coordinate systems are defined:

- Body-fixed frame,  $F^B (X_B, Y_B, Z_B)$  that is fixed to the quadrotor
- Earth-fixed frame,  $F^E (X_E, Y_E, Z_E)$  that is fixed to the earth

To transform the linear velocities and the angular velocities from the Body-fixed Frame to the Earth-fixed frame, the relationship is as follows:

$$\dot{\epsilon} = \begin{bmatrix} R & 0_{3 \times 3} \\ 0_{3 \times 3} & T \end{bmatrix} \begin{bmatrix} \mathcal{V}^B \\ \omega^B \end{bmatrix} \quad (32)$$

where,  $\epsilon$  gives the quadrotor position ( $\epsilon = [x \ y \ z]$ ) and attitude ( $\eta = [\phi \ \theta \ \psi]$ ), i.e.,

$$\epsilon = [x \ y \ z \ \phi \ \theta \ \psi] \quad (33)$$

The linear velocity,  $\mathcal{V}^B$  and angular velocity,  $\omega^B$ , in the body frame is given as

$$\mathcal{V}^B = [u \ v \ w] \quad (34)$$

$$\omega^B = [p \ q \ r]$$

The rotation matrix  $R$  is,

$$R = \begin{bmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi - s_\psi c_\phi & c_\psi s_\theta c_\phi + s_\psi s_\phi \\ s_\psi c_\theta & s_\psi s_\theta s_\phi + c_\psi c_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} \quad (35)$$

where  $c_x \equiv \cos x$  and  $s_x \equiv \sin x$ .

The transformation matrix is given as,

$$T = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi / \cos \theta & \cos \phi / \cos \theta \end{bmatrix} \quad (36)$$

The quadrotor is a nonlinear, under-actuated system as six states are controlled by four input variables. The control of the states is accomplished by having an outer loop and an

inner loop. The outer loop stabilizes the translational states and computes the desired roll and pitch angles which are to be fed to the inner loop to stabilize the attitude. The dynamic equations of the quadrotor are derived using the Newton-Euler method with the assumptions that the quadrotor is rigid and the structure is symmetric, aerodynamic effects are neglected at low speeds. Also the center of mass of the quadrotor is assumed to coincide with the origin of the body-fixed frame.

The Inertial matrix,  $I$  is given as

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (37)$$

The dynamical equations using Newton-Euler's method can be written as [27], [28],

$$\begin{bmatrix} mI_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I \end{bmatrix} \begin{bmatrix} \dot{\mathcal{V}}^B \\ \dot{\omega}^B \end{bmatrix} + \begin{bmatrix} \mathcal{V}^B \times (m\mathcal{V}^B) \\ \omega^B \times (I\omega^B) \end{bmatrix} = \begin{bmatrix} f^B \\ \tau^B \end{bmatrix} = \begin{bmatrix} f_{rotor}^B - f_{gravity}^B \\ \tau_{rotor}^B \end{bmatrix} \quad (38)$$

where  $f_B$  and  $\tau_B$  are the forces and the moments acting on the quadrotor. The forces include the rotor forces ( $f_{rotor}^B$ ) and gravitational force ( $f_{gravity}^B$ ). The moments include the rotor moments ( $\tau_{rotor}^B$ ).

The gravitational forces, rotor forces, and rotor moments are given by

$$\begin{aligned} f_{gravity} &= [0 \quad 0 \quad -mg]^T \\ f_{rotor} &= R [f_{rotor_x} \quad f_{rotor_y} \quad f_{rotor_z}]^T \\ \tau_{rotor}^B &= [\tau_\phi \quad \tau_\theta \quad \tau_\psi] \end{aligned}$$

The dynamic equations of the quadrotor will become

$$\begin{aligned} \ddot{x} &= (\cos \phi \cos \psi \sin \theta + \sin \theta \sin \psi) \frac{1}{m} b(\Sigma \omega_i^2) \\ \ddot{y} &= (\cos \phi \sin \psi \sin \theta - \sin \theta \cos \psi) \frac{1}{m} b(\Sigma \omega_i^2) \\ \ddot{z} &= \cos \theta \cos \phi \frac{1}{m} b(\omega_i^2) - g \\ \ddot{\phi} &= \frac{l}{I_x} b(\omega_4^2 - \omega_2^2) + \dot{\theta} \dot{\psi} \frac{I_y - I_z}{I_x} - \frac{J_r}{I_x} \dot{\theta} \omega_r \\ \ddot{\theta} &= \frac{l}{I_y} b(\omega_3^2 - \omega_1^2) + \dot{\phi} \dot{\psi} \frac{I_z - I_x}{I_y} - \frac{J_r}{I_y} \dot{\phi} \omega_r \\ \ddot{\psi} &= \frac{1}{I_z} b(\omega_1^2 + \omega_3^2 - \omega_2^2 - \omega_4^2) + \dot{\phi} \dot{\theta} \frac{I_x - I_y}{I_z} \end{aligned} \quad (39)$$

where,  $\omega_r = \omega_2 + \omega_4 - \omega_1 - \omega_3$ . The control inputs to the quadrotor are given as,

$$U = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \\ b(\omega_4^2 - \omega_2^2) \\ b(\omega_3^2 - \omega_1^2) \\ d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} \quad (40)$$

The external disturbances can be added to the dynamical equations, which can be rewritten in the following form [29].

$$\ddot{X} = f(X) + g(X)U + \delta \quad (41)$$

where

$$X = [z \quad \phi \quad \theta \quad \psi \quad x \quad y]^T,$$

$$f(X) = \begin{bmatrix} -g \\ \frac{I_y - I_z}{I_x} \dot{\theta} \dot{\psi} + \frac{J_r}{I_x} \omega_r \dot{\theta} \\ \frac{I_z - I_x}{I_y} \dot{\phi} \dot{\psi} + \frac{J_r}{I_y} \omega_r \dot{\phi} \\ \frac{I_x - I_y}{I_z} \dot{\phi} \dot{\theta} \\ 0 \\ 0 \end{bmatrix} \text{ and}$$

$$g(X) = \begin{bmatrix} (1/m) \cos \theta \cos \phi & 0 & 0 & 0 \\ 0 & \frac{l}{I_x} & 0 & 0 \\ 0 & 0 & \frac{l}{I_y} & 0 \\ 0 & 0 & 0 & \frac{1}{I_z} \\ (1/m)u_x & 0 & 0 & 0 \\ (1/m)u_y & 0 & 0 & 0 \end{bmatrix}.$$

$$\delta = [\delta_1 \quad \delta_2 \quad \delta_3 \quad \delta_4 \quad \delta_5 \quad \delta_6]^T$$

where  $u_x$  and  $u_y$  are the virtual control inputs and  $\delta$  is the bounded external disturbance vector ( $|\delta_i| \leq \beta$ ,  $\beta$  is a positive constant). The virtual control inputs are given by

$$\begin{aligned} u_x &= \cos \phi \cos \psi \sin \theta + \sin \phi \sin \psi \\ u_y &= \cos \phi \sin \psi \sin \theta - \cos \psi \sin \phi \end{aligned} \quad (42)$$

## B. Controller Design

Backstepping control ensures global asymptotic stability for the given choice of Lyapunov function. The classical backstepping technique is not robust to model uncertainties and external disturbances. Hence, an integral action, which eliminates the steady-state error due to the modeling and system uncertainties [30], and a saturation function are added to the control law. The error terms, and the control inputs, for the following system dynamics [29]

$$\ddot{X} = f(X) + g(X)U + \delta \quad (43)$$

are defined as follows [26]:

The first tracking error is defined as

$$e_1 = x_{1d} - x_1 \quad (44)$$

The integral of the error term is,

$$p_1 = \int_0^\tau e_1(\tau) d\tau \quad (45)$$

A stabilizing function is defined for the purpose to stabilize the second error term, which is given as

$$\alpha_1 = \dot{x}_{1d} + k_1 e_1 + \lambda_1 p_1 \quad (46)$$

where,  $k_1$  is a positive constant and  $\lambda_1$  is the integral term parameter. Now, the second error term is,

$$e_2 = \alpha_1 - \dot{x}_1 = \dot{x}_{1d} - \dot{x}_1 + \lambda_1 p_1 + k_1 e_1 \quad (47)$$

The Lyapunov function for the first and the second error terms can be defined as,

$$V_1(p_1, e_1) = \frac{1}{2}\lambda_1 p_1^2 + \frac{1}{2}e_1^2 \quad (48)$$

$$V_2(p_1, e_1, e_2) = \frac{1}{2}\lambda_1 p_1^2 + \frac{1}{2}e_1^2 + \frac{1}{2}e_2^2 \quad (49)$$

The derivatives for the Lyapunov functions are given as,

$$\dot{V}_1(p_1, e_1) = e_1(\dot{x}_{1d} - \dot{x}_1 + \lambda_1 p_1) = -k_1 e_1^2 + e_1 e_2 \quad (50)$$

$$\begin{aligned} \dot{V}_2(p_1, e_1, e_2) &= \dot{V}_1(p_1, e_1) + e_2 \dot{e}_2 \\ &= -k_1 e_1^2 + e_2(e_1 + \ddot{x}_{1d} \\ &\quad - (\ddot{x}_1 + k_1 \dot{e}_1 + \lambda_1 \dot{p}_1)) \end{aligned} \quad (51)$$

From the quadrotor dynamic equations, we get

$$\begin{aligned} \dot{V}_2 &= -k_1 e_1^2 + e_2((1 + \lambda_1 - k_1^2)e_1 + k_1 e_2 - k_1 \lambda_1 p_1 \\ &\quad + \ddot{x}_{1d} - f(x_1) - g(x_{11})U_1 - \delta_1) \end{aligned} \quad (52)$$

The control law to stabilize the error terms is,

$$\begin{aligned} U_1 &= \frac{1}{g(x_{11})}[(1 + \lambda_1 - k_1^2)e_1 + (k_1 + k_2)e_2 \\ &\quad - k_1 \lambda_1 p_1 - \epsilon_1 \text{sat}\left(\frac{e_2}{\mu_1}\right) + \ddot{x}_{1d} - f(x_1)] \end{aligned} \quad (53)$$

The derivative of the second Lyapunov function ( $\dot{V}_2$ ) can be proved to be negative definite using the non-vanishing perturbation theorem [26], [31]. Hence, according to Lyapunov stability theory, the controller ensures the asymptotic stability of the states.

The virtual position control inputs  $u_x$  and  $u_y$  and the attitude control inputs can be derived in the same way as  $U_1$ . The desired pitch angle,  $\phi_d$ , can be derived using (42).

## V. SIMULATION RESULTS

TABLE I  
SIMULATION PARAMETERS

Initial Position of Quadrotor	(-45,150,350) m
Initial Velocity of Quadrotor, $\vec{u}$	15 $\hat{i}$ m/s
Turning Radius of Quadrotor, $\rho$	130 m
Minimum Distance Radius, $d_{min}$	152.4 m (500 ft)
Safe Replanning Radius, $d_{safe}$	190.5 m ( $1.25 \times d_{min}$ )
Initial Position of Intruder 1 (Dynamic)	(1036.8, 418.4, 518.4) m
Velocity of Intruder 1, $\vec{v}$	$-(15/\sqrt{6}) [2\hat{i} + \hat{j} + \hat{k}]$ m/s
Initial Position of Intruder 2 (Static)	(1000, 100, 250) m

For the simulation, the quadrotor is considered moving on a straight-line path initially with a constant speed of 15 m/s. The quadrotor detects a dynamic intruder at  $t = 3$  s and hence calculates the collision avoidance and replanning trajectory. At  $t = 35.8$  s, the quadrotor attempts to rejoin with the initial trajectory after successfully avoiding the dynamic obstacle. But, at  $t = 41.4$  s, the quadrotor detects a static obstacle and hence starts the collision avoidance maneuver once again. Finally, at  $t = 67.7$  s, the quadrotor starts its course to join with the initial trajectory and successfully joins at  $t = 84.5$  s. The parameters for the simulation are tabulated in Table

TABLE II  
QUADROTOR PARAMETERS

Mass of the quadrotor, $m$	0.65 kg
Length, $l$	0.27 m
Acceleration due to gravity, $g$	9.81 m/s <sup>2</sup>
Roll Inertia, $I_{xx}$	$5.1 \times 10^{-3}$ kgm <sup>2</sup>
Pitch Inertia, $I_{yy}$	$5.1 \times 10^{-3}$ kgm <sup>2</sup>
Yaw Inertia, $I_{zz}$	$7.6 \times 10^{-3}$ kgm <sup>2</sup>
Thrust factor, $b$	$2.98 \times 10^{-6}$ Ns <sup>2</sup>
Drag factor, $d$	$3.2 \times 10^{-7}$ Ns <sup>2</sup>

I. The external periodic disturbances are given as forces and moments, as  $0.5\sin(t)$  N and  $0.1\sin(t)$  Nm, respectively.

The path tracking algorithm using a 6-degree-of-freedom model of a quadrotor is demonstrated to track the path. The parameters of the quadrotor are tabulated in Table II, and the control parameters used in the simulation are given as [26]

$$\begin{aligned} k_{1-12} &= [5 \ 5 \ 2 \ 2 \ 2 \ 2 \ 10 \ 10 \ 12 \ 12 \ 12 \ 5] \\ \lambda_{1-6} &= [5 \ 3 \ 3 \ 6 \ 6 \ 5] \\ \mu_{1-6} &= [1 \ 1 \ 0.4 \ 0.3 \ 0.3 \ 0.3] \\ \epsilon_{1-6} &= [4 \ 6 \ 5 \ 1 \ 1 \ 3] \end{aligned}$$

The simulation is performed using MATLAB 2019b on an Intel Core i5-7200U processor and 8 GB RAM system.

Fig. 8 shows the desired trajectory and the actual trajectory of the quadrotor tracked by the controller. The Saturation Integral Backstepping control (SIBC) is compared with the Classical Backstepping Control (CBC) approach in the presence of external periodic disturbances. It can be seen that the SIBC approach is robust to the external disturbances compared to the CBC approach, as discussed in IV. The plots of the control inputs and errors are shown in Fig. 9 and 10 respectively. It can be seen that the path planning algorithm has provided an optimal path avoiding obstacles, and the backstepping controller has tracked the path with an acceptable error.

## VI. CONCLUSION

In this paper, a 3-dimensional geometric path planning algorithm that successfully avoids collision with static as well as dynamic obstacles and facilitates the quadrotor to come back to its initial trajectory has been proposed. The algorithms provide an optimal solution by changing the heading angle and the pitch angle of the quadrotor simultaneously while keeping the speed constant. The novelty of the presented approach is that the change in the velocity for collision avoidance is made smooth, making it flyable. A Lyapunov-based backstepping controller has also been designed using a six-degree-of-freedom model of a Quadrotor for tracking the path. The controller is proved to be stable even in the presence of external disturbances such as periodic disturbances in this work. This demonstrates that the proposed approach could be implemented in quadrotors to fly in real-world scenarios.



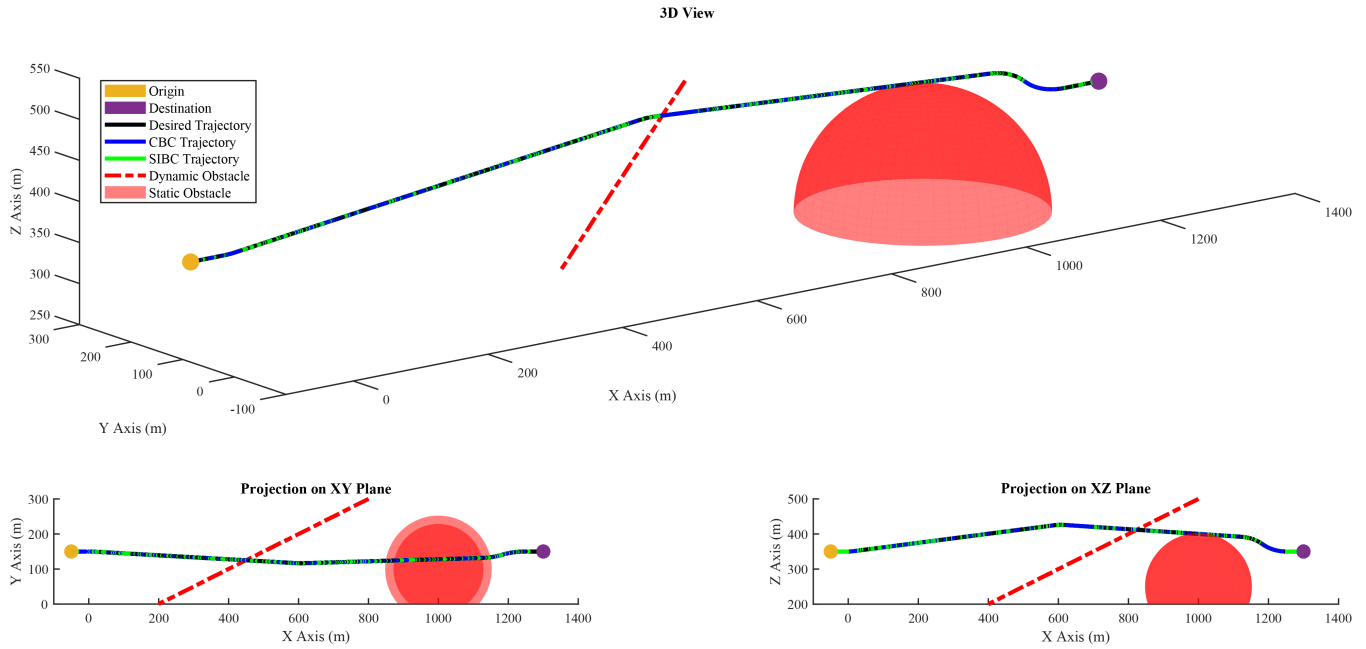


Fig. 8. Comparison of Desired Path and Actual Path of the Quadrotor under periodic disturbances

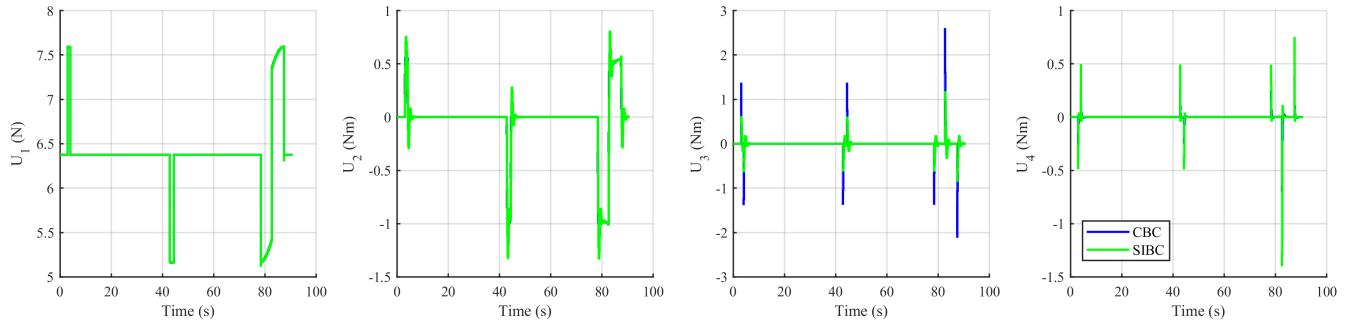


Fig. 9. Control inputs for the quadrotor under periodic disturbances

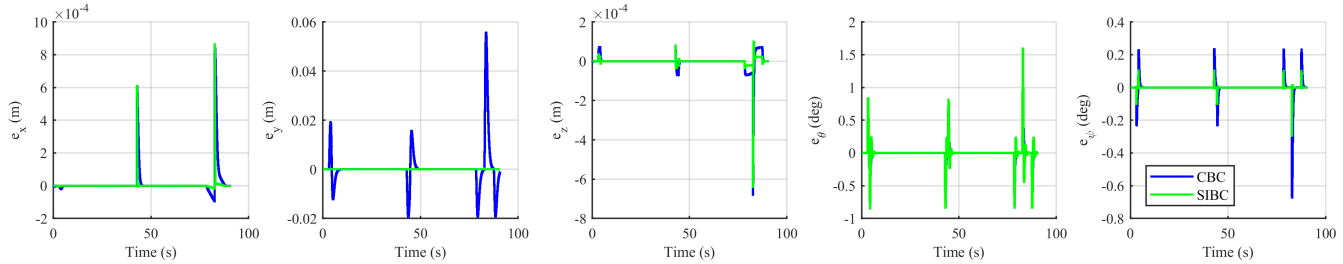


Fig. 10. Tracking errors from the replanned path under periodic disturbances

## REFERENCES

- [1] I. Mahjri, A. Dhraief, and A. Belghith, "A review on collision avoidance systems for unmanned aerial vehicles," in *Communication Technologies for Vehicles*. Cham: Springer International Publishing, 2015, pp. 203–214.
- [2] P. Ferbach, "A method of progressive constraints for nonholonomic motion planning," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 1, pp. 172–179, 1998.
- [3] M. Shanmugavel, A. Tsourdos, and B. A. White, "Collision avoidance and path planning of multiple uavs using flyable paths in 3d," in *2010 15th International Conference on Methods and Models in Automation and Robotics*, 2010, pp. 218–222.
- [4] Y. Lin and S. Saripalli, "Path planning using 3d dubins curve for unmanned aerial vehicles," in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2014, pp. 296–304.
- [5] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, 1985, pp. 500–505.
- [6] J. N. Yasin, S. A. Mohamed, M.-H. Haghbayan, J. Heikkonen, H. Tenhunen, and J. Plosila, "Unmanned aerial vehicles (uavs): Collision avoidance systems and approaches," *IEEE Access*, vol. 8, pp. 105 139–105 155, 2020.
- [7] A. Chakravarthy and D. Ghose, "Obstacle avoidance in a dynamic environment: a collision cone approach," *IEEE Transactions on Systems*,



*Man, and Cybernetics - Part A: Systems and Humans*, vol. 28, no. 5, pp. 562–574, 1998.

- [8] J. Goss, R. Rajvanshi, and K. Subbarao, "Aircraft conflict detection and resolution using mixed geometric and collision cone approaches," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2004, p. 4879.
- [9] C. Carbone, U. Ciniglio, F. Corrado, and S. Luongo, "A novel 3d geometric algorithm for aircraft autonomous collision avoidance," in *Proceedings of the 45th IEEE Conference on Decision and Control*, 2006, pp. 1580–1585.
- [10] S. Luongo, C. Carbone, F. Corrado, and U. Ciniglio, "An optimal 3d analytical solution for collision avoidance between aircraft," in *2009 IEEE Aerospace conference*, 2009, pp. 1–9.
- [11] S. Luongo, F. Corrado, U. Ciniglio, V. Di Vito, and A. Moccia, "A novel 3d analytical algorithm for autonomous collision avoidance considering cylindrical safety bubble," in *2010 IEEE Aerospace Conference*, 2010, pp. 1–13.
- [12] J.-W. Park, H. Oh, and M.-J. Tahk, "Uav conflict detection and resolution based on geometric approach," *International Journal of Aeronautical and Space Sciences*, vol. 10, pp. 37–45, 05 2009.
- [13] C. Lai, M. Lone, P. Thomas, J. Whidborne, and A. Cooke, "On-board trajectory generation for collision avoidance in unmanned aerial vehicles," in *2011 IEEE Aerospace Conference*, 2011, pp. 1–14.
- [14] X. Yang, L. M. Alvarez, and T. Bruggemann, "A 3d collision avoidance strategy for uavs in a non-cooperative environment," *Journal of Intelligent & Robotic Systems*, vol. 70, no. 1, pp. 315–327, Apr 2013.
- [15] Y. I. Jenie, E.-J. van Kampen, C. C. de Visser, J. Ellerbroek, and J. M. Hoekstra, "Three-dimensional velocity obstacle method for uncoordinated avoidance maneuvers of unmanned aerial vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 10, pp. 2312–2323, 2016.
- [16] R. W. Beard and T. W. McLain, *Small unmanned aircraft: Theory and practice*. Princeton University Press, 2012.
- [17] S. Hota and D. Ghose, "Optimal trajectory planning for path convergence in three-dimensional space," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 228, no. 5, pp. 766–780, 2013.
- [18] M. Herrera, W. Chamorro, A. P. Gómez, and O. Camacho, "Sliding mode control: An approach to control a quadrotor," in *2015 Asia-Pacific Conference on Computer Aided System Engineering*. IEEE, 2015, pp. 314–319.
- [19] K. Runcharoon and V. Srichatrapimuk, "Sliding mode control of quadrotor," in *2013 The International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAECE)*. IEEE, 2013, pp. 552–557.
- [20] W. Jasim and D. Gu, "Integral backstepping controller for quadrotor path tracking," in *2015 International Conference on Advanced Robotics (ICAR)*. IEEE, 2015, pp. 593–598.
- [21] S. Bouabdallah and R. Siegwart, "Backstepping and sliding-mode techniques applied to an indoor micro quadrotor," in *Proceedings of the 2005 IEEE international conference on robotics and automation*. IEEE, 2005, pp. 2247–2252.
- [22] A. Akhtar, S. L. Waslander, and C. Nielsen, "Fault tolerant path following for a quadrotor," in *52nd IEEE Conference on Decision and Control*. IEEE, 2013, pp. 847–852.
- [23] C. E. Garcia, D. M. Prete, and M. Morari, "Model predictive control: Theory and practice—a survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.
- [24] M. Bangura and R. Mahony, "Real-time model predictive control for quadrotors," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 11 773–11 780, 2014.
- [25] F. Alyoussef and I. Kaya, "A review on nonlinear control approaches: sliding mode control back-stepping control and feedback linearization control," in *International Engineering and Natural Sciences Conference (IENSC 2019)*, vol. 2019, 2019, pp. 608–619.
- [26] L. Zhou, J. Zhang, H. She, and H. Jin, "Quadrotor uav flight control via a novel saturation integral backstepping controller," *Automatika*, vol. 60, no. 2, pp. 193–206, 2019.
- [27] Z. Benić, P. Piljek, and D. Kotarski, "Mathematical modelling of unmanned aerial vehicles with four rotors," *Interdisciplinary Description of Complex Systems: INDECS*, vol. 14, no. 1, pp. 88–100, 2016.
- [28] D. Kotarski, Z. Benić, and M. Krznar, "Control design for unmanned aerial vehicles with four rotors," *Interdisciplinary Description of Complex Systems: INDECS*, vol. 14, no. 2, pp. 236–245, 2016.
- [29] M. A. M. Basri, A. R. Husain, and K. A. Danapalasingam, "Enhanced backstepping controller design with application to autonomous quadrotor unmanned aerial vehicle," *Journal of Intelligent & Robotic Systems*, vol. 79, no. 2, pp. 295–321, 2015.
- [30] Y. Al Younes, A. Drak, H. Noura, A. Rabhi, and A. El Hajjaji, "Non-linear integral backstepping model-free control applied to a quadrotor system," in *Proceedings of the 10th International Conference on Intelligent Unmanned Systems, Montreal, Canada*, vol. 29, 2014.
- [31] H. Khalil, *Nonlinear Systems*, ser. Pearson Education. Prentice Hall, 2013.