Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Generalised Mathematical Model for the Kinodynamic Planning of Multirotor UAVs and Aerial Robots using Topological Manifolds

Yuvaraj George

p20210046@hyderabad.bits-pilani.ac.in

Birla Institute of Technology and Science - Hyderabad Campus

Venkata Daseswara Rao Yendluri

Birla Institute of Technology and Science - Hyderabad Campus

Research Article

Keywords: Kinodynamic planning, Multirotor UAV, Topological manifold, Holonomic constraints, Actuator-level feasibility, Configuration-aware planning

Posted Date: October 9th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10761790/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Generalised Mathematical Model for the Kinodynamic Planning of Multirotor UAVs and Aerial Robots using Topological Manifolds

Yuvaraj George[1] and Venkata Daseswara Rao Yendluri[1]

Department of Mechanical Engineering, Birla Institute of Technology and Science (BITS), Pilani – Hyderabad Campus, Hyderabad, India
p20210046@hyderabad.bits-pilani.ac.in, yvdrao@hyderabad.bits-pilani.ac.in,
Corresponding Author:

Yuvaraj George
p20210046@hyderabad.bits-pilani.ac.in

Abstract. A generalised mathematical model is introduced for trajectory planning of multi-rotor unmanned aerial vehicles (MUAVs) and aerial robots using topological manifolds. Unlike conventional point-mass planning approaches, the proposed framework models a MUAV as a rigid planar point-set embedded in $\mathbb{R}^3$, with fixed inter-point distances representing rotor geometry under a Euclidean metric. The framework enables systematic propagation of center-of-gravity (CG) trajectories to actuator-level trajectories through direct kinematic mapping, allowing vertex-level curvature, acceleration, and jerk to be evaluated alongside conventional centre-based metrics. Forward and inverse kinematic relations for all vertices of the MUAV are derived, and parametric time-continuous trajectories are constructed to compare with point-mass planning approaches. Simulation studies across different multirotor configurations demonstrate that geometrically distinct platforms sharing identical CG motion can exhibit substantially different actuator-level kinematic profiles, highlighting feasibility and admissibility limitations inherent in point-mass planning models. By integrating physical dimensions directly into the planning formulation, the proposed approach provides a configuration-aware extension to conventional MUAV trajectory planning applicable to both single-vehicle and coordinated multi-vehicle operations.

Key words: Kinodynamic planning, Multirotor UAV, Topological manifold, Holonomic constraints, Actuator-level feasibility, Configuration-aware planning

1 Introduction

A Drone is an unmanned aerial vehicle (UAV). The term “drone" is commonly used to refer to UAVs, particularly rotorcraft or multirotor UAVs. These rotorcraft have propellers powered by brushless direct current (BLDC) motors,

which generate lift to counteract gravitational acceleration. Adjusting the motor speeds changes the position and orientation of a drone in the atmosphere [46]. An aerial robot is a MUAV capable of perceiving its environment, performing motion planning, and executing the plan by commanding motors with collision avoidance [12, 55]. Commercial UAVs often integrate autonomous subsystems for control, battery management, and safety monitoring, operating either through radio-controlled (RC) transmitters or predefined programming [5, 32]. These systems optimize the input provided by the operator or preprogrammed commands. However, the classification of UAVs as aerial robots depends on the required degree of autonomy, as there is no universal definition in UAV classifications. In the case of aerial robots, motion planning, and control are considered optimization problems. In the extant literature, factors such as connectivity [72], battery life [41, 64], and collision avoidance [21] are critical for optimizing configuration and functionality. Autonomy in MUAVs is generally categorized into three functional levels— [1]: path and motion planning [29, 62, 7, 74, 6, 66, 33, 31, 35, 30], optimization [17, 58, 47, 15], and control [36, 9, 61, 3, 54, 14, 20] —each requiring unique algorithm based on mission requirements. Although the fundamental operating principle remains the same, the geometric designs of drones vary [3, 69]. As a result, different configurations require distinct mathematical models and control systems specific to the application and design [40, 34, 2, 70, 4, 18, 23, 24, 51, 53, 56, 75, 77]. Integrating control, optimization, and planning across various MUAV configurations for different missions requires a generalized framework. The available literature highlights the need for a mathematical model that supports various MUAV configurations, this area is largely undeveloped.

MUAVs, as dynamic systems, are commonly modeled using Newton–Euler or Lagrangian formulations for trajectory planning and control [68]. To represent rigid-body motion, two coordinate systems are typically employed: a global reference frame, which is time-invariant, and a body-fixed reference frame, which evolves with the vehicle [19]. These formulations form the standard foundation for modeling and control of aerial robots.

In most UAV mathematical models, the vehicle is represented using a point-mass abstraction, wherein the system state is defined relative to its CG, with or without differential constraints [1, 43]. Historically, this modeling choice was motivated by limited onboard computational capacity, which encouraged minimizing the number of parameters used to describe the system [49, 16, 67, 58, 26]. Consequently, the control and planning frameworks for MUAVs often prioritize compact state representations by treating the vehicle as a point mass.

Within this abstraction, motion planning is typically formulated as a nonholonomic planning problem [16, 34, 38, 40, 45, 76]. The point-mass representation offers a generalized and computationally efficient approach to solving the path planning decision problem, i.e., determining the existence of an admissible path between two configurations [42]. However, by construction, this abstraction does not explicitly account for the physical dimensions, geometry, or fixed inter-rotor distances of the vehicle [48, 63, 10].

A free-flying aerial vehicle possesses six degrees of freedom, comprising three translational and three rotational components. Orientation is commonly parameterized using yaw–pitch–roll Euler angles [25, 44], rotation matrices, or quaternions. These representations are mathematically equivalent and widely used in practice, each with well-understood numerical and implementation trade-offs [68]. Euler angle parameterizations may encounter kinematic singularities under certain configurations [65], motivating the use of quaternions or matrix-based representations in many control and planning frameworks [39, 50, 22]. Regardless of the chosen representation, trajectory solutions obtained at the CG level must ultimately be mapped to realizable velocity or acceleration fields to generate physical motion [11]. Approximations such as skew-symmetric matrix representations under small angular deviations are commonly employed to facilitate this mapping [52].

Despite these well-established modeling tools, challenges persist when kinematic planning abstractions are applied to dynamic systems. In particular, while mass is a scalar quantity, rotational dynamics are governed by a symmetric inertia matrix defined in the body frame, whose properties depend on the vehicle geometry and mass distribution [25]. These characteristics are inherently tied to the physical configuration of the vehicle rather than solely its CG trajectory.

Several studies have extended classical formulations to improve modeling fidelity. Benic et al. [8] derived dynamic mappings between configuration space and applied forces using Newton–Euler equations. Bouzid et al. [13] incorporated atmospheric disturbances and demonstrated robust closed-loop control using sliding-mode techniques. Md. Tolba and Bijan [69] proposed a generic model for unbalanced MUAVs by relaxing the assumption that the point mass coincides with the geometric center of the vehicle.

While these approaches improve dynamic accuracy, they largely retain a CG-centric point-mass planning. As a result, holonomic constraints arising from fixed geometric relationships—such as constant inter-rotor distances—are not explicitly represented during trajectory planning [52, 43] . Consequently, different MUAV configurations may share identical CG trajectories while exhibiting distinct curvature, acceleration, and jerk profiles at the actuator or vertex level. This gap motivates the need for a planning and analysis framework that preserves the advantages of point-mass planning while explicitly capturing configuration-dependent effects induced by the vehicle's geometry.

The main contributions of this work are as follows:

1. A generalized mathematical model for kinodynamic planning of multirotor UAVs is proposed, in which the vehicle is represented as a rigid planar point-set embedded in $\mathbb{R}^3$, preserving standard rigid-body degrees of freedom while explicitly enforcing fixed inter-point distance constraints. Here, kinodynamic planning refers to time-parameterized nonholonomic motion planning subject to system-inherent bounds on velocity, acceleration, and jerk, without invoking full dynamic or control-input optimization.
2. A unified kinematic formulation that enables direct propagation of CG trajectories to actuator-level trajectories, allowing the motion of individual ro-

tors to be expressed within a common Cartesian framework. Unlike conventional geometry-aware planning approaches—where vehicle geometry is primarily used to identify collision-free paths, the proposed mathematical model employs vehicle geometry to evaluate time-parameterised trajectories at the actuator level.

3. A planning and analysis framework that extends conventional point-mass trajectory planning by incorporating physical dimensions into feasibility and admissibility evaluation, without altering the underlying rigid-body motion representation. The proposed model does not replace existing path or trajectory planning algorithms; rather, it enables the assessment of kinematic feasibility metrics (velocity, acceleration, and jerk) for individual rotors without altering the underlying planning algorithm.
4. An analytical and simulation-based study showing that multirotor platforms with identical CG motion can exhibit distinct actuator-level curvature, acceleration, and jerk profiles depending on configuration.

2 Methodology

Existing mathematical models for MUAVs are often developed for specific vehicle configurations or under simplifying assumptions that limit their applicability across different multirotor geometries [8]. While a generalized mathematical model is essential for achieving autonomy and systematic control in MUAVs, most existing approaches that aim at generalization restrict their applicability to specific configurations or vehicle classes. As a result, current models either sacrifice generality or rely on assumptions that limit their extensibility across different multirotor geometries.

To remain broadly applicable, a generalized MUAV model must necessarily rely on structured approximations. Such generalization is more naturally introduced at the kinematic level, where configuration-independent representations can be formulated without loss of topological consistency. In contrast, dynamic characteristics—such as inertia, actuator limits, and thrust distributions—can be incorporated through boundary conditions imposed on the kinematic constraints. This separation allows trajectory planning to proceed even in the absence of a fully specified dynamic model, while still yielding executable motion plans once configuration-dependent parameters are applied. Consequently, a generalized kinodynamic framework remains feasible for MUAV trajectory planning and coordination in shared operational airspace.

From this perspective, the principal shortcoming lies in the inadequacy of mathematical models to consistently represent MUAV geometry and constraints across configurations. In particular, point-mass representations commonly employed in kinodynamic planning do not explicitly enforce holonomic constraints associated with fixed geometric relationships, such as constant inter-rotor distances. In this work, the proposed formulation is therefore examined in comparison with a point-mass representation to evaluate its implications for planning and feasibility analysis.

The point-mass CG position and orientation are first obtained in the form of a homogeneous transformation matrix. The MUAV arms are then extended relative to this CG transformation to compute the position of each vertex in the body frame. Consequently, the proposed mathematical model consolidates multiple coordinate systems into a unified Euclidean representation, enabling kinematic descriptions to be expressed entirely in real-valued Cartesian coordinates.

The formulation represents kinematic quantities within a common Cartesian framework while remaining fully consistent with standard rigid-body motion representations. This unified approach simplifies the interpretation of kinematic solutions and enables direct mapping between configuration space and workspace. Forward and inverse kinematic solutions are derived within this framework and subsequently analysed to characterise the structure, continuity, and admissible ranges of motion.

The model is further validated through kinodynamic planning simulations of a multirotor UAV in Euclidean space, where holonomic constraints corresponding to vehicle geometry are explicitly incorporated. By enforcing these constraints, the proposed formulation preserves continuity of the generated motion and enables systematic comparison of admissible trajectories. The generated trajectories are then compared with those obtained using a conventional point-mass representation, highlighting differences in admissibility and configuration-dependent behavior.

To assess practical feasibility, simulations were conducted for both forward and inverse kinematics, and the resulting trajectories and state estimates were compared against those obtained using a point-mass MUAV model. The results, presented in subsequent sections, demonstrate that while the CG trajectory remains invariant, configuration-dependent effects emerge at the vertex level, influencing curvature, acceleration, and jerk. These observations support the use of the proposed model as a generalized extension to conventional point-mass planning, particularly for addressing admissible path decision problems in aerial robots.

3 Results & Discussions

3.1 Mathematical Modeling

The concept of the topological manifold is utilized. The configuration of a MUAV was mapped to a 2-D manifold, explicitly representing the geometric constraints of the system. It is a symmetric closed convex polygon similar to that of a quadcopter with four propellers, forming a planar quadrilateral in three-dimensional space. The simulations and validations for quadcopter are presented in this article as it is a widely used MUAV configuration. Nevertheless, the formulations are more generic and applicable to other configurations.

Consider a symmetric quadcopter with four propellers forming the rigid planar polygon (rotor polygon) $V_1V_2V_3V_4$ in three-dimensional space. V_1, V_2, V_3, and

V_4 are the positions of the motors of the MUAV. V_1, V_2, V_3, and V_4 are the rigid planar polygon (rotor polygon)'s vertices representing the planar quadrilateral. Following the standard convention in aeronautics, rotation matrices are multiplied to obtain the 3×3 matrix as shown in (1). (1) represents the orientation of the MUAV.

$$\begin{aligned} Rot_{3\times3}(ori) &= Rot_{z,ori}(\Theta)\, Rot_{y,ori}(\Psi)\, Rot_{x,ori}(\Phi) \\ &= \overset{\text{pitch}}{\begin{bmatrix} \cos\Theta & -\sin\Theta & 0 \\ \sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 1 \end{bmatrix}} \overset{\text{yaw}}{\begin{bmatrix} \cos\Psi & 0 & \sin\Psi \\ 0 & 1 & 0 \\ -\sin\Psi & 0 & \cos\Psi \end{bmatrix}} \times \\ &\qquad \overset{\text{roll}}{\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\Phi & -\sin\Phi \\ 0 & \sin\Phi & \cos\Phi \end{bmatrix}} \end{aligned} \tag{1}$$

The transformation matrix for the position and orientation of CG is given in (2).

$$Trans_{Homo} = \left[\begin{array}{ccc|c} & Rot_{3\times3}(ori) & & \begin{matrix} x_{cg} \\ y_{cg} \\ z_{cg} \end{matrix} \\ \hline P_x & P_y & P_z & S.F \end{array}\right] \tag{2}$$

The transformation matrix is formulated for the MUAV, and the order of operation follows conventional SRT (Scale, Rotate & Translate). The point mass of the quadcopter is attached to the center of the rigid planar polygon (rotor polygon) as it is the origin of the local reference frame. (2) gives the position and orientation of the local reference frame $X'Y'Z'$, which is shown in Figure.1.

$$\vec{V_i} = \begin{bmatrix} 1 & 0 & 0 & r\cos\left(\frac{360}{n}i + \gamma^\circ\right) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r\sin\left(\frac{360}{n}i + \gamma^\circ\right) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

$$\vec{V_i}(Trans) = Trans_{Homo} \times \vec{V_i} \tag{4}$$

In the Figure.1, the rigid planar polygon (rotor polygon) is shown on the plane $X'Z'$. The distance from CG to the vector 'r' is the distance between the BLDC motor's shaft and the CG of the MUAV. The CG is the center of the rigid planar polygon (rotor polygon), and the 'n' vertices are equidistant from the center for symmetric MUAV configuration. Thus, the vertices from CG shall have a 3×3 Identity matrix as a rotational matrix, a perspective matrix with a scaling factor, and $\pm\vec{r}$ on X'-axis and Z'-axis. The rigid planar polygon

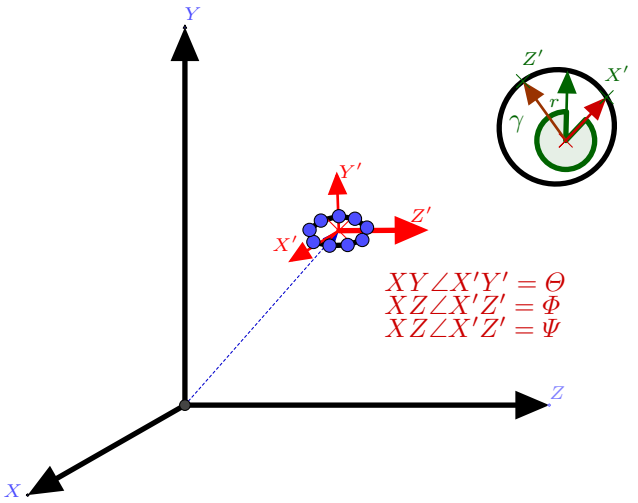


Fig. 1: Topological mapping

(rotor polygon) shall have only two coordinates, and the Y' component in the matrix is zero. The equation (3) is the generalized form of 4×4 transformation matrices for any velocity vector of MUAV (vertex of the rigid planar polygon (rotor polygon)) in the local reference frame. Here, i ranges from 0 to $n-1$, where n is the number of motors for the MUAV. γ° is the offset angle of the motor about the local body frame. Therefore post-multiplying each vertex matrix of the body frame in (3) with (2) gives the global representation of the MUAV's vertex. Therefore, a MUAV with 'n' motors shall have 'n' vertices forming a closed convex polygon. (5) gives the generalized transformation matrix of the vector in the global reference frame. Similarly, a MUAV with n motors can be represented as equation (6).

$$\vec{V_i}(Trans) = \begin{bmatrix} a_{00} & a_{01} & a_{02} & x_{cg} + r\cos\left(\frac{360}{n}i + \gamma^\circ\right) \\ a_{10} & a_{11} & a_{12} & y_{cg} \\ a_{20} & a_{21} & a_{22} & z_{cg} + r\sin\left(\frac{360}{n}i + \gamma^\circ\right) \\ P_x & P_y & P_z & S.F \end{bmatrix} \tag{5}$$

$$V_{plane} = \begin{bmatrix} V_{0x} & .... & V_{nx} \\ V_{0y} & .... & V_{ny} \\ V_{0z} & .... & V_{nz} \\ 1 & .... & 1 \end{bmatrix} = Trans_{Homo} \times \begin{bmatrix} r\cos\left(\frac{360}{n}0 + \gamma^\circ\right) & .... & r\cos\left(\frac{360}{n}i + \gamma^\circ\right) \\ y_{cg} & .... & y_{cg} \\ r\sin\left(\frac{360}{n}0 + \gamma^\circ\right) & .... & r\sin\left(\frac{360}{n}i + \gamma^\circ\right) \\ 1 & .... & 1 \end{bmatrix} \tag{6}$$

Thus, (6) gives the coordinates of the rigid planar polygon (rotor polygon) as multiplied by the transformation matrix. It also gives the plane's orientation with respect to the global reference frame. The rigid planar polygons (rotor polygons) vertices have an identity matrix as a rotational matrix, as shown in (3). So, its product with CG's rotational matrix will yield the same orientation as the CG. Therefore, the orientation of the CG is the orientation of the rigid planar polygon (rotor polygon). (6) inherits a holonomic constraint by the MUAV's geometry when $r\cos(\frac{360}{n}i+\gamma^\circ)$ & $r\sin(\frac{360}{n}i+\gamma^\circ)$ is multiplied with the homogeneous transformation matrix -$Trans_{Homo}$. The geometric constraint is that the distance between the V_i and CG is always constant *'r'* for all translations and rotations of the MUAV. The constraint can be written as (7). From (5), the global coordinates of any vertex in Euclidean space are V_{ix} is $x_{cg}+r\cos(\frac{360}{n}i+\gamma^\circ) \equiv x_{cg}+r\cos(L)$, V_{iy} is y_{cg} and V_{iz} is $z_{cg}+r\sin(\frac{360}{n}i+\gamma^\circ) \equiv z_{cg}+r\sin(L)$. Substitution of these global coordinates of any vertex in (7) will prove that r is constant.

$$r=\sqrt{(V_{ix}-x_{cg})^2+(V_{iy}-y_{cg})^2+(V_{iz}-z_{cg})^2} \tag{7}$$

Apart from point mass representation being absent with holonomic constraints, three are translational, and three are rotational components that require two sets of coordinates, i.e., Cartesian and spherical, about three axes. Representing the MUAV as a rigid planar polygon (rotor polygon) and considering only Cartesian coordinates to present 'n+1' coordinates deduces the model to one coordinate system and, to an extent, avoids the singularities due to trigonometric relations. Nevertheless, the rotational components are implicit in the formulation. For the spatial motion of the MUAV, the holonomic constraint shall be satisfied for all spatial translations. Therefore, representing a MUAV as a rigid planar polygon (rotor polygon) with 'n' vertices yields 'n' sets of state variables. This solves the problem of control singularity as the number of state variables is equal to the number of control variables. Each state variable is a set of three Cartesian coordinates, and the system's state in 3D space can be defined by the 'n+1' sets of Cartesian coordinates ('n' vertices & CG).

The formulations used map the point mass as a rigid planar polygon (rotor polygon) to incorporate geometric constraints. Thus, the rigid planar polygon (rotor polygon) is represented as a set of 3 independent parameters in three-dimensional space $\mathbb{R}^3$, which is a special Euclidean group $SE(3)$. Thus, the global reference frame will have three axes representing a rigid planar polygon (rotor polygon) without trigonometric relations in path planning $f\{x,y,z,\theta,\phi,\psi\} \mapsto f'\{n,r,\gamma^\circ,f(x_{cg},y_{cg},z_{cg})\} \mapsto f''\{[x:y:z]_{3\times n+1}\}$. Therefore, this mapping function serves as a topological manifold for the path planning of MUAV. Thus, geometric constraints are introduced into formalism with homomorphism. The MUAV structure is assumed to be rigid; structural deformations are negligible. The topological manifold preserves the system's degrees of freedom in three dimensions such that $D(D+1)/2 = 3(3+1)/2 = 6$, where D is the dimensionality of SE(3) group. In other words, this topological manifold increases the number

of elements in the set of state variables where the parameters are essentially Cartesian coordinates.

3.2 Dynamic Modelling

As discussed in the Introduction, a significant body of existing literature emphasises generalised dynamic modelling of UAV systems, including motor dynamics, thrust generation, and nonlinear rigid-body equations. They often limit the applicability in real-time trajectory planning and generalised path generation problems. However, the approach is to generalise kinematics and formulate dynamics implicitly as bounding conditions.

Consider Newton's second law applied to the UAV (8)

$$F_{\text{thrust}} = \frac{d(m\mathbf{v})}{dt} \tag{8}$$

where m is the mass of the MUAV and $\mathbf{v}$ is its velocity vector. For battery-powered UAVs, the mass remains constant during operation, as no fuel is consumed. Therefore, (8) simplifies to:

$$F_{\text{thrust}} = m\frac{d\mathbf{v}}{dt} = m\mathbf{a} \tag{9}$$

where $\mathbf{a}$ denotes the acceleration of the UAV.

Under steady flight conditions, the thrust generated by the propulsion system balances the gravitational force (10).

$$F_{\text{thrust}} = mg \tag{10}$$

Therefore, the net force acting on the UAV governs its acceleration as (11).

$$\sum \mathbf{F} = m\mathbf{a} = F_{\text{thrust}} - mg \tag{11}$$

As observed from (11), the excess thrust over gravitational force determines the acceleration capability of the UAV.

Let $F_{\max}$ and $F_{\min}$ denote the maximum and minimum thrust generated by the propulsion system. Using (11), the corresponding acceleration limits can be expressed as (13) and (12) for minimum and maximum accelerations respectively.

$$a_{\max} = \frac{F_{\max} - mg}{m} \tag{12}$$

$$a_{\min} = \frac{F_{\min} - mg}{m} \tag{13}$$

These bounds represent the dynamic capabilities of the UAV imposed by actuator and propulsion constraints. Since velocity is obtained by integrating acceleration over time (14)

$$\mathbf{v}(t) = \int \mathbf{a}(t)\, dt \tag{14}$$

The bounded acceleration conditions for minimum acceleration (13) and maximum acceleration (12) result in bounded velocity (15).

$$v_{\min} \leq \|\mathbf{v}(t)\| \leq v_{\max} \tag{15}$$

In the present work, the maximum velocity is selected as $v_{\max} = 10$ m/s, which lies within the operational envelope of commercially available off-the-shelf quadrotor UAVs.

From (12)–(15), it follows that:

1. Thrust limitations constrain acceleration,
2. Acceleration bounds constrain velocity,
3. Velocity bounds constrain feasible trajectories.

Thus, instead of explicitly modelling the complete nonlinear UAV dynamics, the proposed approach incorporates these effects implicitly through bounded velocity and acceleration constraints. This ensures that the generated trajectories remain dynamically feasible while being simple for real-time implementation.

3.3 Trajectory Analysis

By representing the MUAV as a rigid planar polygon (rotor polygon), a topological manifold, geometric constraints were inherently incorporated into the model. This ensured that the distance between each motor and the CG remained constant, reflecting the rigid body structure of the MUAV. This approach explicitly enforces fixed inter-point distance constraints and enables rotor-level trajectories to be computed directly from the CG trajectory and orientation, allowing curvature, acceleration, and jerk to be evaluated at both the CG and rotor points. By addressing the limitations of traditional modeling techniques, this approach provided a robust and efficient framework for MUAV trajectory planning and control. This section outlines the mathematical procedures for solving the forward and inverse kinematics of any planar MUAV. The forward kinematics is focused on Position and Orientation Tracking, while the inverse kinematics deals with Parametric Motion.

A typical mission profile of the MUAV is analyzed, as shown in Figure 2. It is noted that the MUAV takes off from the ground at $(0, 0, 0)$ and climbs to an altitude of 50 m, reaching the coordinates $(0, 50, 0)$, labeled as $\mathbf{0} - \mathbf{1}$. The MUAV is then reported to tilt in the X-Z plane to move along a straight path to a distance of 200 m at coordinates $(200, 50, 200)$, with angular displacements of $\theta = 10^\circ$ and $\phi = 10^\circ$ at $\mathbf{1}$. It is observed that this transition is followed by $\mathbf{1} - \mathbf{1'}$ and $\mathbf{1'} - \mathbf{2}$, which represent the linear translation. Before descending, it is ensured that the MUAV reorients to 0° for both θ and ϕ at $\mathbf{2} - \mathbf{2'}$, after which it begins its descent of 50 m, as shown by $\mathbf{2'} - \mathbf{3}$.

Table 1: Position Estimates of CG and Vertices Satisfying Geometric Constraints

Pos	**CG**	V_0	V_1	V_2	V_3	**d(CG,V**$_i$**)± 0.01** $\forall\ i \in [0,3]$
0	(0,0,0)	(0.1,0,0)	(-0.029,0,-0.0954)	(-0.082,0,0.0571)	(0.079,0,0.061)	[0.1, 0.1,0.1,0.09]
1	(0,50,0)	(0.1,50,0)	(-2.9e-2,50,-9.5e-2)	(-0.08,50,0.05)	(0.07,50,0.06)	[0.1, 0.09,0.1,0.09]
1'	(0,50,0)	(1e-2,49.9,-3e-2)	(-9e-2,50,-2e-2)	(4e-2,50,4e-2)	(6e-2,49.9,-4e-3)	[0.09,0.09,0.09,0.09]
2	(200,50,200)	(2e2,50,2e2)	(1.9e2,50,1.9e2)	(1.9e2,50,2e2)	(2e2,50,2e2)	[0.1,0.1,0.09,0.09]
2'	(200,50,200)	(2e2,49.9,1.9e2)	(1.9e2,50,1.9e2)	(1.9e2,49.9,1.9e2)	(2e2,49.9,1.9e2)	[0.09,0.1,0.1,0.09]
3	(200,50,200)	(200.1,0,200)	(1.9e2,0,1.9e2)	(1.9e2,0,2e2)	(2e2,0,2e2)	[0.1,0.1,0.1,0.09]

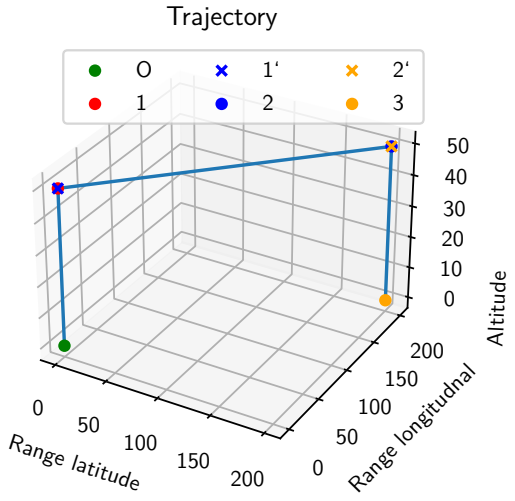
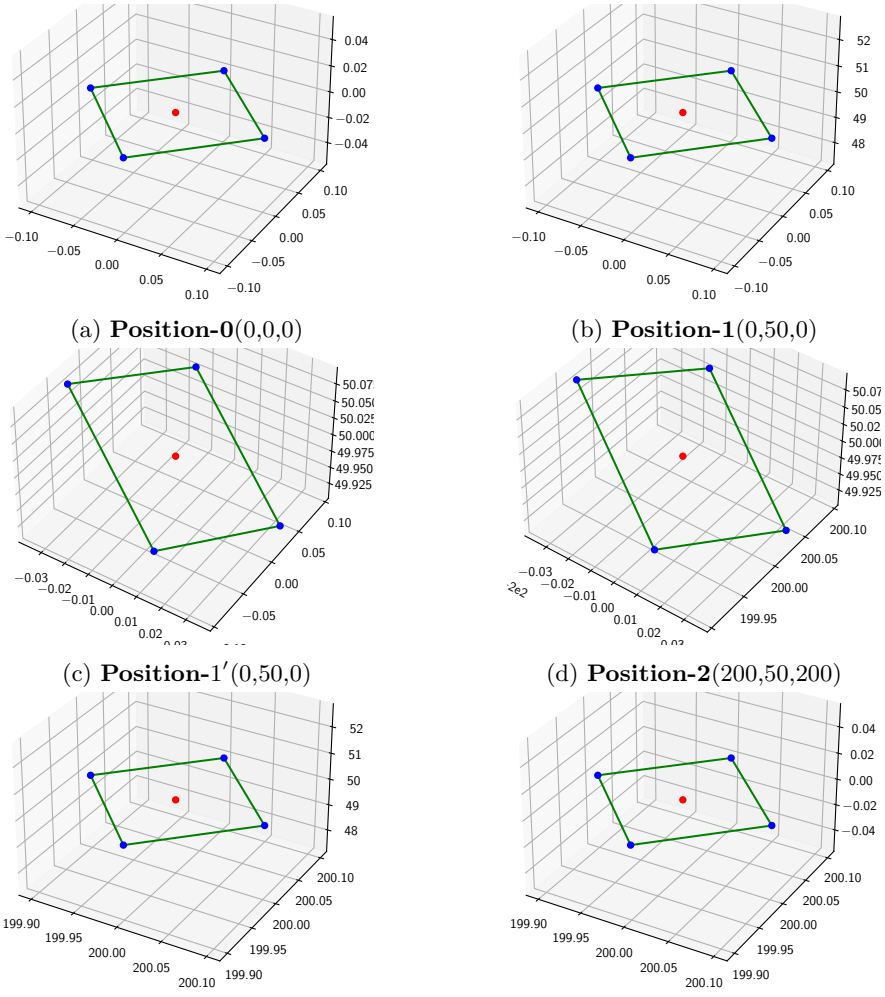


Fig. 2: Mission profile

Position and Orientation Tracking: Forward Kinematics The position and orientation at each stage of the mission profile are shown in Figure 3. From the figure, it is derived that the MUAV's representation as a rigid planar polygon (rotor polygon) in (6) implicitly includes both position and rotation coordinates for all vertices in 3D Euclidean space. The coordinates are generated as shown in Table 1, ensuring that the distance $d(CG, V_i)$ between the CG and any vertex V_i on the quadcopter's rigid planar polygon (rotor polygon) (with $n = 4$ and $r = 0.1$) remains constant and equals r. Owing to finite numerical precision and discretisation in the simulation environment, the rigid-distance constraint $d(CG, V_i) = \|V_i - \mathrm{CG}\| = r$ is satisfied up to a numerical tolerance of $\varepsilon = 10^{-2}$ m. Accordingly, the distances reported in Table 1 deviate from the nominal radius only within this tolerance. Additionally, it is confirmed that the condition in (7) is satisfied. Vertices exhibit symmetry about the CG along all three axes, which is akin to the physical behaviour of the MUAV. As shown in Table 1, there is symmetry about the CG, indicating that the behaviour closely resembles the actual response of the MUAV. The transition of vertices in the pitch motion displays symmetry about opposite motors, while in yaw, symmetry is observed about adjacent motors. The solutions of forward kinematics for estimating the position of 'n' vertices are found to have real solutions, ensuring the physical realizability of the model. Thereby, each vertex corresponds to each actuator and with a unique coordinate in 3D Euclidean space. Literature suggests that unique control signals for each actuator, based on their unique states, are beneficial for achieving robust and efficient control system performance [59, 60]. This approach is said to allow for precise adjustments, better handling of uncertainties, and improved overall system stability and efficiency.



(a) **Position-0**(0,0,0)

(b) **Position-1**(0,50,0)

(c) **Position-**$1'$(0,50,0)

(d) **Position-2**(200,50,200)

(e) **Position-**$2'$(200,50,200)

(f) **Position-3**(200,0,200)

Fig. 3: Position and orientation of the rigid planar polygon (rotor polygon) at various segments of the mission profile

Thus, the proposed mapping function for MUAV state estimation is hence validated. It is ensured that the homomorphism between the transformed rigid planar polygon (rotor polygon) and the angular orientation of the point-mass is preserved. This facilitates the computation of motor positions and control signals. This has practical application for MUAV when the model is integrated with path-following algorithms. It is assumed that the state of the MUAV is obtained from IMU data, which is transformed into a state manifold, and the waypoint is transformed into a target manifold. With this model, the controller can estimate the error for each motor. The proposed mapping is said to convert a 6-DOF

system into n Cartesian points, where n is the number of actuators, thereby enabling rotor-level state trajectories and error signals without additional post-processing in the kinematic mapping stage. This section establishes the admissibility of the proposed kinematic formulation in reconstructing actuator (vertex) states in Cartesian coordinates from the CG position and orientation.

Parametric Motion: Inverse Kinematics This section constructs a cubic parametric kinodynamic model for MUAV motion, first at the CG level and subsequently extended to vertex-level manifolds. The objective is to generate executable control trajectories without additional actuator mapping while preserving geometric constraints. The trajectory in the present work is represented using cubic polynomials. For a trajectory segment defined by boundary conditions, the formulation requires satisfying two initial conditions and two final conditions. This results in a system of four equations with four unknown coefficients. Hence, third-order (cubic) polynomial is the minimum-order polynomial capable of satisfying these constraints. Since there will be four unknowns, a cubic polynomial provides exactly the required degrees of freedom for a unique solution.

In trajectory planning applications, cubic polynomials are widely used in robotics and control systems due to their ability to generate smooth and continuous paths between two points. They ensure continuity of position and velocity while maintaining computational simplicity. Furthermore, the present formulation considers motion up to acceleration, which corresponds to the second derivative of position. A cubic polynomial naturally provides continuous acceleration profiles, making it suitable for UAV trajectory generation without introducing discontinuities in motion. Higher-order polynomials, such as quintic or minimum-snap ormulations, allow additional constraints on higher-order derivatives such as acceleration and jerk. However, these approaches significantly increase computational memory usage due to the larger number of coefficients and the need for higher numerical precision. This becomes particularly relevant for embedded flight controllers with limited computational resources (e.g., 8-bit or 16-bit microcontrollers), where floating-point precision and memory constraints restrict the practical implementation of higher-order models. Therefore, a cubic polynomial is selected as a suitable trade-off between trajectory smoothness, computational efficiency, and real-time implementability.

The proposed model can also be applied to implement motion planning algorithms for planar MUAVs, provided their constraints are known to the planar MUAVs in advance -apriori. This section focuses on estimating feasible trajectories for a MUAV given a set of waypoints. For this case, we consider the six waypoints outlined in the mission profile, and the corresponding trajectory for all four motors/vertices needs to be calculated, ensuring each motor has a unique solution. The objective is to avoid redundancy and the need for additional mapping functions to link the generated control signals to the actuators, thus enabling low-level control. The trajectory generation algorithm presented in this section is simulated, and the results are compared with those from a point

mass system and a topological manifold system in the simulation results and discussions section.

The spatial motion of the MUAV in the atmosphere can be represented as a parametric function of time and Cartesian coordinates alone. A cubic function may represent the MUAV's position at any point in the space-time domain. The representation of continuous motion as a function of time yields a time-continuous cubic polynomial trajectory parameterization [57]. It is also known as a third-order polynomial. Although higher-order polynomial equations are employed for various robotic applications, the formulations opt for the third-order polynomial. Mathematically, a polynomial of order 'k' needs 'k+1' conditions to be satisfied. A third-order polynomial can have two initial conditions and two final conditions. The two constraints can be state variables representing spatial position and time. In other instances, it can be their time derivatives. Vector field representation of the space-time continuum will yield a vector position in X, Y, Z such that $x\hat{i} + y\hat{j} + z\hat{k}$ and time t is a scalar quantity. Although the number of variables here is 4 ($f\{x, y, z, t\}$), their identities are two, viz. position [x,y,z] and time[t]. The number of initial conditions must be equal to the number of state variables for a dynamic system [37] because initial conditions represent the state of the dynamic system and can aid in predicting the succeeding state of the system. At the same time, the two initial conditions for a single point would be a spatial position, a vector quantity, and time, a scalar quantity. As per a priori analytics, the MUAV's position is a function of Cartesian coordinates and time f(x,y,z,t). So, (16) shows that for a point along the X-axis, the third-order polynomial has the initial position c_0, velocity c_1, acceleration c_2, and jerk c_3. Thus, the third-order polynomial is suitable for interpreting physical parameters.

Representations of (16) show that every parameter of point mass representation shall have a motion defined in parametric equation form. Its time derivatives are the velocity as in (17), and the second derivatives yield acceleration (18). The cubic equation has four coefficients; the matrix of these coefficients is the coefficient matrix. The matrix with elements of 't' is the parametric matrix with time as an independent-constraint parameter. The coefficient matrix values depend on the system's dynamic characteristics. The value 't' is the input parameter for the MUAV's translational and rotational motion. The parametric equation is a function of the independent-constraint parameter (time) and their derivatives (16) in matrix form about CG for transnational (19) and rotational components(20). The parametric equation will have different coefficients for different planes and different segments of the mission profile.

$$
\begin{aligned}
x(t) &= c_{0x} + c_{1x}t + c_{2x}t^2 + c_{3x}t^3,\\
y(t) &= c_{0y} + c_{1y}t + c_{2y}t^2 + c_{3y}t^3,\\
z(t) &= c_{0z} + c_{1z}t + c_{2z}t^2 + c_{3z}t^3,\\
\theta(t) &= c_{0\theta} + c_{1\theta}t + c_{2\theta}t^2 + c_{3\theta}t^3,\\
\phi(t) &= c_{0\phi} + c_{1\phi}t + c_{2\phi}t^2 + c_{3\phi}t^3,\\
\psi(t) &= c_{0\psi} + c_{1\psi}t + c_{2\psi}t^2 + c_{3\psi}t^3.
\end{aligned} \tag{16}
$$

$$
\begin{aligned}
\dot{x}(t) &= c_{1x} + c_{2x}2t + c_{3x}3t^2,\\
\dot{y}(t) &= c_{1y} + c_{2y}2t + c_{3y}3t^2,\\
\dot{z}(t) &= c_{1z} + c_{2z}2t + c_{3z}3t^2,\\
\dot{\theta}(t) &= c_{1\theta} + c_{2\theta}2t + c_{3\theta}3t^2,\\
\dot{\phi}(t) &= c_{1\phi} + c_{2\phi}2t + c_{3\phi}3t^2,\\
\dot{\psi}(t) &= c_{1\psi} + c_{2\psi}2t + c_{3\psi}3t^2.
\end{aligned} \tag{17}
$$

Given each segment's position, velocity, and acceleration, the time taken to complete the motion is calculated between two points (x_0, y_0, z_0) and (x_1, y_1, z_1), which is a forward kinematics problem. The velocity, acceleration, and jerk are defined as minimum and maximum values for each segment depending on the inertial and motor characteristics of the MUAV. Moreover, such a notation needs 12 boundary conditions to solve the motion in all six axes, and eventually, it will require a $6 \times n$ coefficient matrix to represent a point mass in 3D space.

The inverse kinematics problem is challenging. The objective of mapping a point mass as a rigid planar polygon (rotor polygon) with 'n' vertices is to reduce a polar coordinate system of the local body frame into one Cartesian coordinate system. The matrix in (6) gives the Cartesian coordinates of the rigid planar polygon (rotor polygon). Its size is $4 \times n$, where the first three rows give the coordinates of X, Y & Z (20) becomes redundant. The parametric function of any rigid planar polygon (rotor polygon)'s vertex is given by (21). However, this will also increase the elements in each variable. In other words, the mapping function converts a single point mass as multiple bodies constrained by geometry, i.e., Euclidean isometries. Thus, a non-transcendental parametric equation is formulated for path planning.

$$
\begin{aligned}
\ddot{x}(t) &= c_{2x}2t + c_{3x}6t^2,\\
\ddot{y}(t) &= c_{2y}2t + c_{3y}6t^2,\\
\ddot{z}(t) &= c_{2z}2t + c_{3z}6t^2,\\
\ddot{\theta}(t) &= c_{2\theta}2t + c_{3\theta}6t^2,\\
\ddot{\phi}(t) &= c_{2\phi}2t + c_{3\phi}6t^2,\\
\ddot{\psi}(t) &= c_{2\psi}2t + c_{3\psi}6t^2.
\end{aligned} \tag{18}
$$

$$\begin{bmatrix} x_{cg} \\ y_{cg} \\ z_{cg} \end{bmatrix}^T = \begin{bmatrix} 1 \\ t \\ t^2 \\ t^3 \end{bmatrix}^T \times \begin{bmatrix} c_{0x_{cg}} & c_{0y_{cg}} & c_{0z_{cg}} \\ c_{ax_{cg}} & c_{ay_{cg}} & c_{az_{cg}} \\ c_{bx_{cg}} & c_{by_{cg}} & c_{bz_{cg}} \\ c_{1x_{cg}} & c_{1y_{cg}} & c_{1z_{cg}} \end{bmatrix} \tag{19}$$

$$\begin{bmatrix} \theta_{cg} \\ \phi_{cg} \\ \psi_{cg} \end{bmatrix}^T = \begin{bmatrix} 1 \\ t \\ t^2 \\ t^3 \end{bmatrix}^T \times \begin{bmatrix} c_{0\theta_{cg}} & c_{0\phi_{cg}} & c_{0\psi_{cg}} \\ c_{a\theta_{cg}} & c_{a\phi_{cg}} & c_{a\psi_{cg}} \\ c_{b\theta_{cg}} & c_{b\phi_{cg}} & c_{b\psi_{cg}} \\ c_{1\theta_{cg}} & c_{1\phi_{cg}} & c_{1\psi_{cg}} \end{bmatrix} \tag{20}$$

$$\begin{bmatrix} V_{ix} \\ V_{iy} \\ V_{iz} \end{bmatrix}^T = \begin{bmatrix} 1 \\ t \\ t^2 \\ t^3 \end{bmatrix}^T \times \begin{bmatrix} c_{0x_i} & c_{0y_i} & c_{0z_i} \\ c_{1x_i} & c_{1y_i} & c_{1z_i} \\ c_{2x_i} & c_{2y_i} & c_{2z_i} \\ c_{3x_i} & c_{3y_i} & c_{3z_i} \end{bmatrix} \tag{21}$$

$$\begin{bmatrix} \vec{p_0} \\ \vec{p_a} \\ \vec{p_b} \\ \vec{p_1} \end{bmatrix} = \begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 \\ 1 & t_a & t_a^2 & t_a^3 \\ 1 & t_b & t_b^2 & t_b^3 \\ 1 & t_1 & t_1^2 & t_1^3 \end{bmatrix} \times \vec{C_{4\times 3}} \tag{22}$$

$$\begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 \\ 1 & t_a & t_a^2 & t_a^3 \\ 1 & t_b & t_b^2 & t_b^3 \\ 1 & t_1 & t_1^2 & t_1^3 \end{bmatrix}^{-1} \times \begin{bmatrix} \vec{p_0} \\ \vec{p_a} \\ \vec{p_b} \\ \vec{p_1} \end{bmatrix} = \vec{C}_{4\times 3} \tag{23}$$

$$\begin{bmatrix} p_0\hat{i} + p_0\hat{j} + p_0\hat{k} \\ p_a\hat{i} + p_a\hat{j} + p_a\hat{k} \\ p_b\hat{i} + p_b\hat{j} + p_b\hat{k} \\ p_1\hat{i} + p_1\hat{j} + p_1\hat{k} \end{bmatrix} = \begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 \\ 1 & t_a & t_a^2 & t_a^3 \\ 1 & t_b & t_b^2 & t_b^3 \\ 1 & t_1 & t_1^2 & t_1^3 \end{bmatrix} \times \begin{bmatrix} \vec{C_{0p}} & \vec{C_{ap}} & \vec{C_{bp}} & \vec{C_{1p}} \\ \dot{\vec{C}}_{0p} & \dot{\vec{C}}_{ap} & \dot{\vec{C}}_{bp} & \dot{\vec{C}}_{1p} \\ \ddot{\vec{C}}_{0p} & \ddot{\vec{C}}_{ap} & \ddot{\vec{C}}_{bp} & \ddot{\vec{C}}_{1p} \\ \dddot{\vec{C}}_{0p} & \dddot{\vec{C}}_{ap} & \dddot{\vec{C}}_{bp} & \dddot{\vec{C}}_{1p} \end{bmatrix} \tag{24}$$

If the coefficient matrix of each vertex is known, then the forward kinematics problem can be solved for any vertex of rigid planar polygon (rotor polygon) between two spatial points p_0 and p_1. (21) computes the vector field's magnitude for any rigid planar polygon (rotor polygon) vertex. (22) and (23) can be expressed using vector notations for two positions. Let the final position be denoted as $\vec{p_i} = p_{ix}\hat{i} + p_{iy}\hat{j} + p_{iz}\hat{k}$. The corresponding time derivatives are: $\dot{\vec{C}}_{0p} = \dot{C}_{0px}\hat{i} + \dot{C}_{0py}\hat{j} + \dot{C}_{0pz}\hat{k}$ for velocity, $\ddot{\vec{C}}_{0p} = \ddot{C}_{0px}\hat{i} + \ddot{C}_{0py}\hat{j} + \ddot{C}_{0pz}\hat{k}$ for acceleration, and $\dddot{\vec{C}}_{0p} = C_{0px}\hat{i} + C_{0py}\hat{j} + C_{0pz}\hat{k}$ for jerk, where $\vec{C_{0p}}$ represents the initial position. When the vertex of a rigid planar polygon (rotor polygon)

moves from p_0 to p_1, the MUAV's state can be described by 'n' linear equations in vector form.

The parametric matrix for points $\vec{p_0}(x0, y0, z0)$ and $\vec{p_1}(x1, y1, z1)$ as shown in (22). In (22) and (23), the vectors are in Euclidean dimensions, and each component has $\hat{i}, \hat{j}, \hat{k}$. Thus, the velocities, acceleration, and positions of various vertices of the rigid planar polygon (rotor polygon) enable the formulation to compute coefficients without losing homomorphism. In (22), each row is a vector component of the same vertex at different time intervals. The path between two arbitrary points, p_0 and p_1, transverses between two intermittent points, a and b, as shown in (21). The parametric motion of each vertex on the rigid planar polygon (rotor polygon) yields vector components along all three axes of the $SE(3)$. For a matrix to have an inverse, it must be a non-singular matrix. Four coordinates must be defined to achieve matrix multiplication, and a non-determinant T matrix or a trajectory segment must be divided into four sections. When (22) pre-multiplied with T^{-1}, it yields the coefficient matrix for a given time to accomplish the motion. As the matrix is non-singular determinant is $\det(T) = \prod_{1 \leq i < j \leq n}(t_j - t_i) = (t_a - t_0)(t_b - t_0)(t_1 - t_0)(t_b - t_a)(t_1 - t_a)(t_1 - t_b)$ as 'T' is Vandermonde matrix. (23) gives the inverse kinematics equation of any vertex of the rigid planar polygon (rotor polygon). As $t_1 > t_b > t_a > t_0$, the $\det(T)$ is always positive, and T is a matrix with only positive real integers. Thus, the determinant of the is greater than zero (and non-zero) for all positive real integers of t values as long as the t-values are distinct [71].

The state-space representation in vector form is shown on (24). Therefore, this creates a topological manifold for each motor. The control of motors should be such that all vertices of the rigid planar polygon (rotor polygon) remain in their defined manifold at any given time. This gives 'n' control parameters and 'n' state variables $r(t)$. The $r(t)$ in (25) is a set of parametric functions for the rigid planar polygon (rotor polygon) in three-dimensional space, which is obtained from (23) and (24). The time derivatives of (25) give the velocity, acceleration, and jerk as given in(26),(27), and (28), respectively.

$$\mathbf{r}_i(t) = \begin{bmatrix} x_i(t) \\ y_i(t) \\ z_i(t) \end{bmatrix}, \qquad \forall\, i \in \{0, \ldots, n-1\} \tag{25}$$

$$\dot{\mathbf{r}}_i(t) = \frac{d\mathbf{r}_i(t)}{dt} = \begin{bmatrix} \dot{x}_i(t) \\ \dot{y}_i(t) \\ \dot{z}_i(t) \end{bmatrix}, \qquad \forall\, i \in \{0, \ldots, n-1\} \tag{26}$$

$$\ddot{\mathbf{r}}_i(t) = \frac{d^2\mathbf{r}_i(t)}{dt^2} = \begin{bmatrix} \ddot{x}_i(t) \\ \ddot{y}_i(t) \\ \ddot{z}_i(t) \end{bmatrix}, \qquad \forall\, i \in \{0, \ldots, n-1\} \tag{27}$$

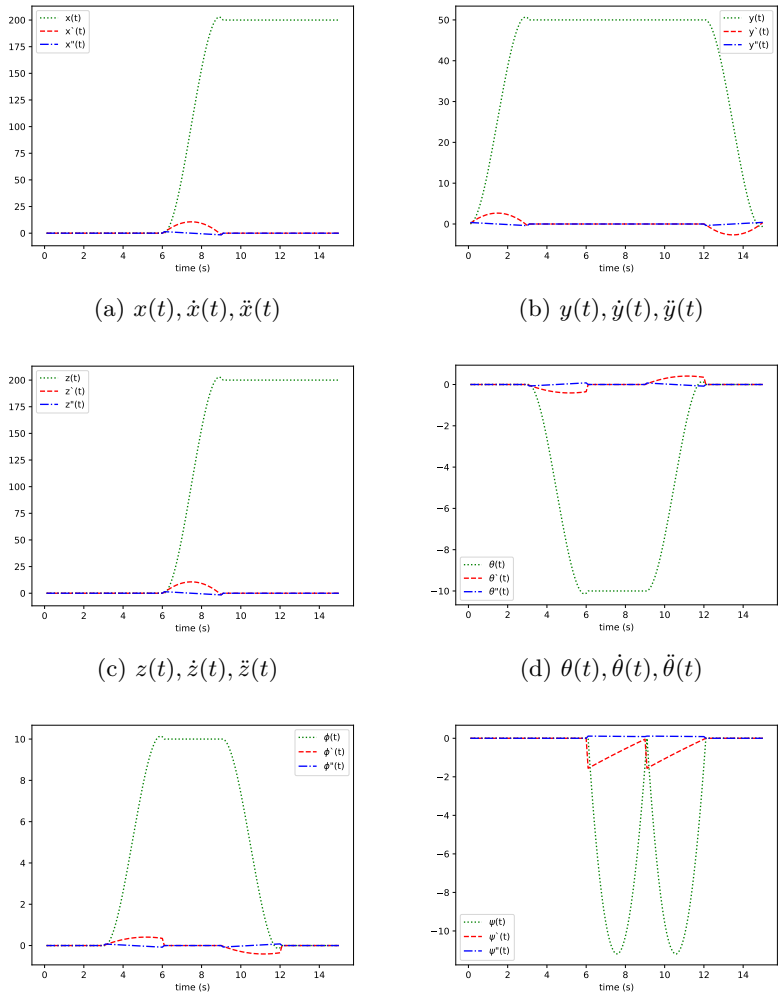
$$\dddot{\mathbf{r}}_i(t) = \frac{d^3\mathbf{r}_i(t)}{dt^3} = \begin{bmatrix} \dddot{x}_i(t) \\ \dddot{y}_i(t) \\ \dddot{z}_i(t) \end{bmatrix}, \qquad \forall\, i \in \{0, \ldots, n-1\} \tag{28}$$

The kinodynamic model can be deployed to various MUAVs and aerial robots, provided the system's dynamic characteristics are known for maximum and minimum velocities and accelerations. The time required to complete a trajectory varies for each MUAV based on its dynamic characteristics and response, and dynamic constraints are finally introduced into the path and motion planning of MUAV.

Algorithm 1 Algorithm.1 for Parametric Motion Planning

Require: $[\![x_{cg}]\!],\ [\![y_{cg}]\!],\ [\![z_{cg}]\!], [\![\theta_{cg}]\!],\ [\![\phi_{cg}]\!],\ [\![\psi_{cg}]\!], n, r, \gamma^\circ$
Ensure: $|x_{cg}| = |y_{cg}| = |z_{cg}| = |\theta_{cg}| = |\phi_{cg}| = |\psi_{cg}|$
Ensure: $f : (\mathbb{R}^2, SO(3)/\mathbb{R}^3) \rightarrow nSE(3)/\mathbb{R}^3$ ▷ (1),(4)
  for $i \leftarrow 0\ |X_{cg}|$ *by* 1 **do** ▷ $\forall V_i$
    $Trans_{Homo}[k] = f\{x_{cg}[k], y_{cg}[k], z_{cg}[k],$
  $\theta_{cg}[k], \phi_{cg}[k], \psi_{cg}[k]\}$ ▷ (2)
    $V_{plane}[k] = Trans_{Homo}[k] \times V_i$
  $\forall i \in [0 : n-1]$ ▷ (6)
Ensure: $r \simeq constant$ ▷ (7)
  Define $T : det(T) \neq 0$
  for *segment* $\leftarrow 0\ |x_{cg-1}|$ *by* 1 **do**
    Compute $Trans_{Homo}[segment]$,$V_{plane}[segment]$
    for $i \leftarrow 0\ n-1$ *by* 1 **do** ▷ $\forall V_i$
      Calculate $(x_a, y_a, z_a), (x_b, y_b, z_b)$ ▷ *Intermittent points length* $\propto$ *time (dynamic characteristics of system)*
      Compute T^{-1} and C ▷ (23)(24)
      Obtain $r_i(t), \dot{r_i}(t),\ \ddot{r_I}(t)$ and $\dddot{r_I}(t)$ ▷ (25),(26),(27)

The simulation results are used to compare CG-level planning with rotor point-set planning and to quantify configuration-dependent differences in actuator-level kinematic metrics. A quadcopter with an arbitrary controller has identified waypoints to reach from Position 0 to Position 3, illustrated in Figure 2, with a radius of $r = 0.1$. It is equipped with four BLDC motors, one for each vertex. For the given mission profile, the Algorithm.1 was used to generate the velocity profile of topological manifolds for each vertex. For a point mass representation, the given mission profile has five segments with 6 points in the coordinate system that define the entire mission profile about the CG of MUAVs. Here, two segments are rotational, and three are translational. Each segment has two via points, each with six coordinates about CG, i.e., x_{CG}, y_{CG}, z_{CG}, θ_{CG}, ϕ_{CG}, ψ_{CG}. Using (23) & (24) T^{-1} and C are computed. The time 't' depends on the motor characteristics. Here, it is assumed that $t_0 = 0$, $t_a = 9.81$, $t_b = 9.81^2$, & $t_1 = 9.81^3$.



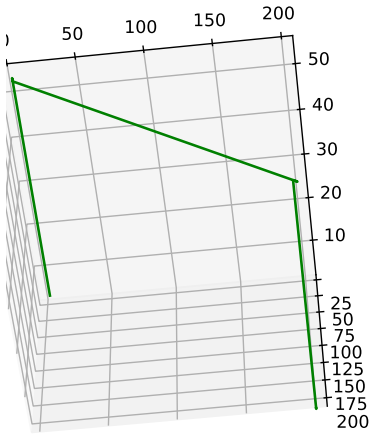
(a) $x(t), \dot{x}(t), \ddot{x}(t)$ (b) $y(t), \dot{y}(t), \ddot{y}(t)$

(c) $z(t), \dot{z}(t), \ddot{z}(t)$ (d) $\theta(t), \dot{\theta}(t), \ddot{\theta}(t)$

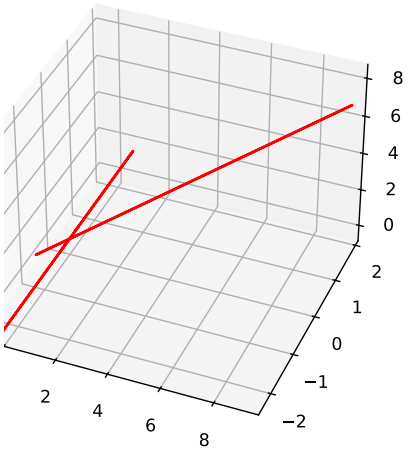
(e) $\phi(t), \dot{\phi}(t), \ddot{\phi}(t)$ (f) $\psi(t), \dot{\psi}(t), \ddot{\psi}(t)$

Fig. 4: Parametric motion of 6 parameters

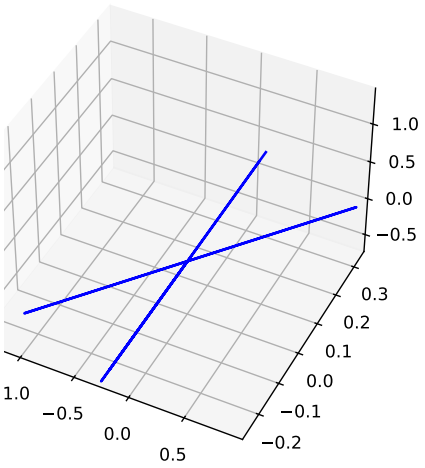
Figure.4 illustrates the distance, velocity, and acceleration of the MUAV's CG for the mission profile. Figure.4a, Figure.4b, and Figure.4c show the position, velocity, and acceleration of CG about X, Y, and Z axes, respectively. The angular displacement, angular velocity, and angular acceleration of the MUAV for θ, ϕ, and ψ are illustrated in the figure.4d, Figure.4e, and Figure.4f respectively. Figure.4 illustrates that the parametric representation of the path as a function of time yields a continuous time function of the path and results in optimal planning of a smooth trajectory. For point mass representation, the via



(a) $\mathbf{r(t)}$=$\mathbf{f\{}x(t), y(t), z(t)\mathbf{\}}$

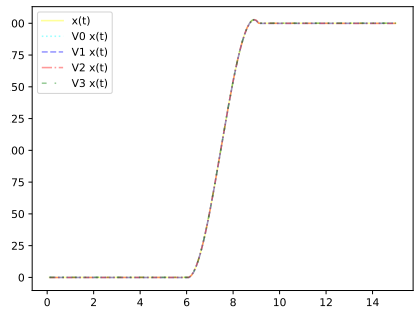


(b) $\dot{r}(t)$=$\mathbf{f\{}\dot{x}(t), \dot{y}(t), \dot{z}(t)\mathbf{\}}$

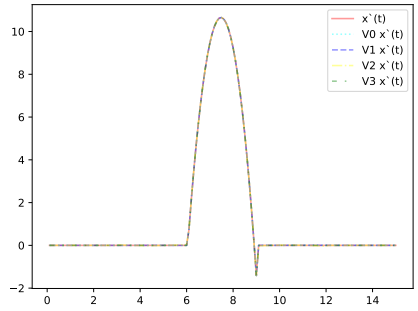


(c) $\ddot{r}\mathbf{(t)}$=$\mathbf{f\{}\ddot{x}(t), \ddot{y}(t), \ddot{z}(t)\mathbf{\}}$

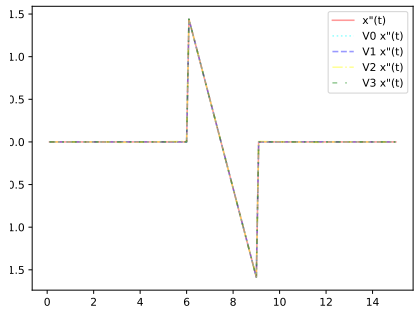
Fig. 5: Mission profile as a function of time



(a) $\mathbf{f\{}x(t), V0(t), V1(t), V2(t), V3(t)\}$



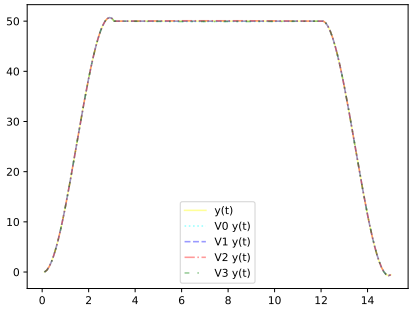
(b) $\mathbf{f\{}\dot{x}(t), V\dot{0}x(t), V\dot{1}x(t), V\dot{2}x(t), V\dot{3}x(t)\}$



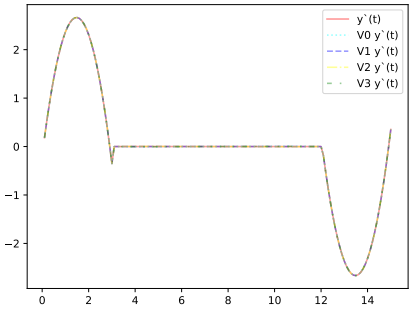
(c) $\mathbf{f\{}\ddot{x}(t), V\ddot{0}x(t), V\ddot{1}x(t), V\ddot{2}x(t), V\ddot{3}x(t)\}$

Fig. 6: Geometric manifolds in X-axis

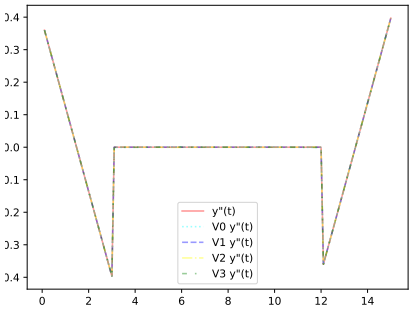
points are the control points of a planned motion profile. Via points are specific checkpoints positioned between two waypoints that allow for control and monitoring of progress. The system shall exhibit controllability at each segment's four points, $p_0, p_a, p_b \& p_1$, for which the number of state variables should be equal to the number of control variables. Figure.4 has six parameters representing the state, but the number of control variables is four. With this, the planned motion for the point mass of the MUAV control is not realisable without additional mapping functions. This is observable in Figure.5, which shows the displacement, velocity, and acceleration as a function of time in the Euclidean configuration



(a) $\mathbf{f\{}y(t), V0y(t), V1y(t), V2y(t), V3y(t)\}$



(b) $\mathbf{f\{}\dot{y}(t), V\dot{0}y(t), V\dot{1}y(t), V\dot{2}y(t), V\dot{3}y(t)\}$

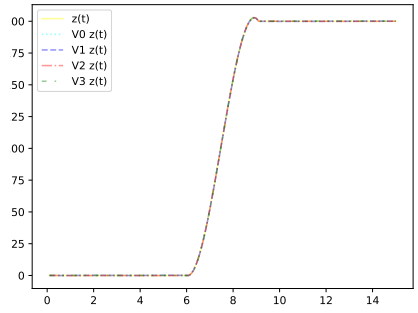


(c) $\mathbf{f\{}\ddot{y}(t), V\ddot{0}y(t), V\ddot{1}y(t), V\ddot{2}y(t), V\ddot{3}y(t)\}$

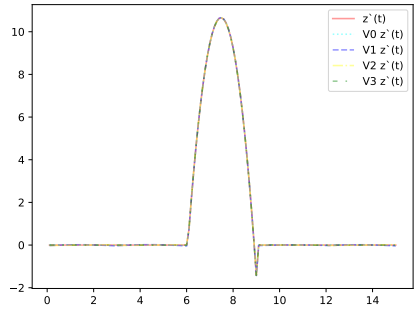
Fig. 7: Geometric manifolds in Y-axis

space of the MUAV when represented as CG or point mass, which has only one trajectory for all four actuators.

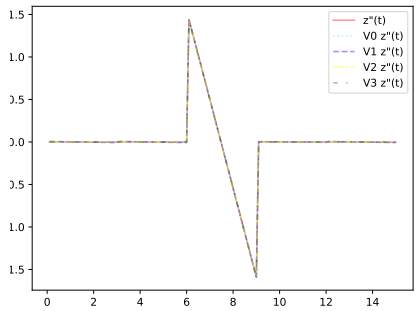
In Figures 6, 7, and 8, the horizontal axis represents time in seconds, while the vertical axis varies depending on the subplot: it shows position in centimeters for subplot (a), velocity in centimeters per second for subplot (b), and acceleration in centimeters per second squared for subplot (c). Figure 6 shows the position, velocity, and acceleration of the X-axis for CG, along with the vertices of the rigid planar polygon (rotor polygon). Similarly, Figure.7 and Figure.8 show the position and time derivatives of position about the Y-axis and Z-axis,



(a) $\mathbf{f\{}z(t), V0z(t), V1z(t), V2z(t), V3z(t)\}$



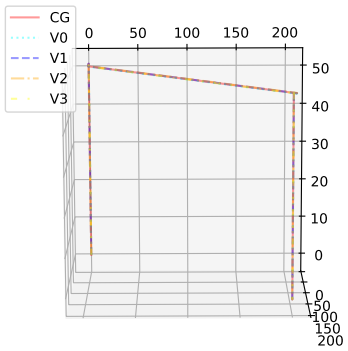
(b) $\mathbf{f\{}\dot{z}(t), V\dot{0}z(t), V\dot{1}z(t), V\dot{2}z(t), V\dot{3}z(t)\}$



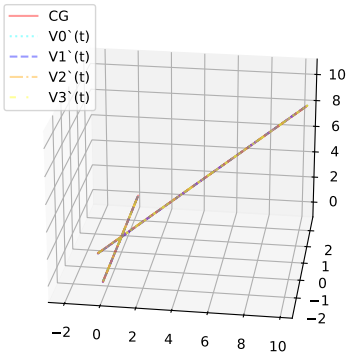
(c) $\mathbf{f\{}\ddot{z}(t), V\ddot{0}z(t), V\ddot{1}z(t), V\ddot{2}z(t), V\ddot{3}z(t)\}$

Fig. 8: Geometric manifolds in Z-axis

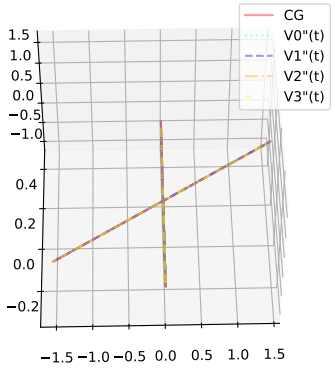
respectively. As is evident from the figure.4, Figure.6, Figure.7, and Figure.8, the velocity profiles of all vertices on a defined rigid planar polygon (rotor polygon) as a function of time. Figure.9 shows the spatial and kinematic manifolds for all the vertices and the CG of the quadcopter. It is evident that Figure.6, Figure.7, and Figure.8 are similar to those shown in Figure.4. The velocities, acceleration, and jerk of the vertices are parallel to the profiles of the CG because the MUAV is a rigid body. The tangent to the path of any vertex is always perpendicular to the rigid planar polygon (rotor polygon), as the tangent is the velocity vector of the path. All 'n+1' vertices (4 vertices for a quadcopter and CG) form



(a) $\mathbf{f\{}r(t), V0(t), V1(t), V2(t), V3(t)\}$



(b) $\mathbf{f\{}\dot{r}(t), \dot{V0}(t), \dot{V1}(t), \dot{V2}(t), \dot{V3}(t)\}$



(c) $\mathbf{f\{}\ddot{r}(t), \ddot{V0}(t), \ddot{V1}(t), \ddot{V2}(t), \ddot{V3}(t)\}$

Fig. 9: Geometric manifolds as Parametric functions

a polygon with CG as the centre. This polygon is convex and has a geometric manifold through which it displaces as a function of time, as shown in the figure.9a. Figures 9b and 9c illustrate the velocity and acceleration of the rigid planar polygon (rotor polygon)'s vertices as a time function. Thus, four unique control profiles are generated as a function of time for four BLDC motors of the quadcopter.

The plots clearly show that the planned trajectory, as well as the velocity and acceleration profiles of the point-mass model, are identical to those of the proposed manifold model's CG. However, the manifold model offers several advantages. It accommodates 'n' trajectories for 'n' BLDC motors and provides reliable planning and decision-making capabilities.

This section demonstrates the feasibility of generating parametric trajectories for each vertex, corresponding to individual actuators, given target position and orientation specifications of the CG. Accordingly, the inverse kinematics problem for the MUAV is solved within the proposed framework, enabling actuator-level trajectory generation directly from CG pose commands. The formulation operates entirely in Cartesian coordinates, allowing trajectory planning to be performed without the explicit use of Euler angles. Although orientation parameters are employed during state estimation in earlier sections, they are implicitly captured within the Cartesian representation of the vertex trajectories during planning. As a result, the inverse kinematic solution avoids singularities commonly associated with angle-based parameterisations in both control and kinematic computations. Each vertex trajectory is subject to holonomic constraints defined in Cartesian space, ensuring kinematic feasibility while implicitly enforcing the overall orientation of the MUAV structure. This unified treatment enables smooth, continuous, and singularity-free motion generation, establishing a direct and consistent mapping from CG pose targets to actuator configurations.

3.4 Theoretical Validation and Simulation Proof

Consider an example of two MUAVs, MUAV A and MUAV B, designed with identical kinematic parameters. MUAV A and MUAV B are scaled versions of each other, meaning that while their dimensions differ slightly, their kinematic behaviours—such as velocity, acceleration, and angular motion—are the same. Both MUAVs are required to navigate through a cylindrical pipe. The path planning and trajectory generation for these MUAVs are performed using a point mass model, where only the CG kinematics are considered, ignoring the physical dimensions of the MUAVs. In the point mass system, the admissible collision-free path planned for MUAV A is identical to the path for MUAV B since both have the same kinematic parameters. This implies that, theoretically, both MUAVs should be able to follow the same trajectory through the pipe. However, MUAV B has slightly larger dimensions than MUAV A. Due to this size difference, MUAV B cannot pass through the pipe, even though it has the same kinematics as MUAV A. The reason is that the point mass model does not account for the physical dimensions or holonomic constraints. That is the MUAV's size relative

to the pipe's diameter. Thus, while the point mass system predicts the same admissible path for both MUAVs, in reality, MUAV B cannot follow the path due to its larger size, which causes it to collide with the pipe's walls. This example highlights the importance of considering holonomic constraints, such as physical dimensions, when planning paths for real-world applications where the environment imposes spatial limitations. This is the paradox of point mass kinodynamic planning. In contrast, when topological manifolds are generated, they inherently account for these physical dimensions, ensuring the planned paths are kinematically feasible and physically realisable. Theorem.1 enhances the decision problem towards completeness of kinodynamic path planning by explicitly addressing the relationship between physical dimensions and path feasibility.

Finally, as per Laumond, *"If there exists a collision-free and admissible path between two configurations A and B in symmetric small-time controllable systems, then there also exists a collision-free path between these two configurations* [42, 73]."

Theorem 1. *Let there be two rigid planar polygons (rotor polygons) A' and B' with different radii, i.e.,r values, r_A and r_B. Then the admissible collision-free path of rigid planar polygon (rotor polygon) A' is the admissible collision-free path for rigid planar polygon (rotor polygon) B' if and only if the r value of rigid planar polygon (rotor polygon) B', i.e., r_B, is less than or equal to the r value of rigid planar polygon (rotor polygon) A', i.e., r_A. Mathematically, this can be expressed as:*

$$Path(A') = Path(B') \iff r_B \leq r_A$$

The topological arguments combined with analytical geometry resulted in a nonholonomic motion planning for MUAV [42]. The presented model has holonomic constraints as co-planar constraints of topological arguments and has the following traits:

1. **Symmetric convex polygon:** MUAV is represented as a symmetric convex polygon in the Euclidean space $\mathbb{R}^3$, i.e., SE(3) group. The rigid planar polygon (rotor polygon) is subject to holonomic geometric constraints that preserve fixed inter-point distances between vertices. The instantaneous velocity of a vertex arises from the superposition of translational motion of the centre of gravity and rotational motion of the rigid body and is therefore not, in general, constrained to be perpendicular to the polygonal plane. While the motors generate thrust perpendicular to their base planes, the kinematic velocity of each motor is governed by rigid-body motion and depends on both the translation of the centre of gravity and the angular rotation of the UAV. As such, motor velocity vectors are not generally perpendicular to the UAV polygonal plane.
2. **Euclidean isometries:** The distance between the motor's shaft and CG serves as a holonomic co-planar constraint and preserves the homomorphism

of the topology. This enables the rigid body constraint along with the geometric constraint for MUAVs.

3. **Finite-Time Local Reachability:** Due to the imposed bounds on velocity, acceleration, and jerk, the system admits a strictly positive minimum transition time between distinct states. Consequently, the proposed framework guarantees finite-time local reachability of neighbouring states via time-parameterised kinematic trajectories, but does not guarantee small-time controllability, which requires reachability in arbitrarily small time using admissible control inputs. This notion of reachability is purely kinematic and should not be interpreted as local or small-time controllability in the control-theoretic sense [42].
4. **Topological manifolds:** A MUAV with 'n' control variables will have 'n+1' topological manifolds computed. However, the path or trajectory of CG is a reference.

Simulation Validation of Theorem 1

Proof (Simulation Proof).

To validate Theorem 1, we simulate a fixed center-of-gravity (CG) trajectory and attitude profile and evaluate how actuator-level (vertex) kinematic metrics vary with rotor arm length. The CG position and orientation are prescribed as:

$$\left.\begin{aligned} x(t) &= 2\sin(0.4t), \\ y(t) &= 2\cos(0.4t), \\ z(t) &= 0.4t + 1, \\ \psi(t) &= 0.4\sin(0.25t), \\ \theta(t) &= 0.15\sin(0.3t), \\ \phi(t) &= 0.15\cos(0.3t), \end{aligned}\right\} \quad t \in [0, 12] \text{ s.} \tag{29}$$

For each multirotor configuration, the i-th rotor (vertex) position is obtained from the standard rigid-body mapping

$$V_i(t) = p_{cg}(t) + R(t)\, r_i, \tag{30}$$

where $p_{cg}(t) = [x(t), y(t), z(t)]^T$, $R(t) = R_{ZYX}(\psi(t), \theta(t), \phi(t))$, and r_i denotes the fixed body-frame rotor location.

We consider the following configurations and rotor arm lengths: tricopter ($r = 0.30$ m), pentacopter ($r = 0.55$ m), hexacopter ($r = 0.66$ m), and octacopter ($r = 0.88$ m). For each case, we compute path metrics for both the CG and rotor trajectories, including path length, curvature, and jerk. Curvature is computed as

$$\kappa(t) = \frac{\|\dot{p}(t) \times \ddot{p}(t)\|}{\|\dot{p}(t)\|^3}, \tag{31}$$

and jerk magnitude as $j(t) = \| \dddot{p}(t)\|$, with reported mean and maximum values evaluated over $t \in [0, 12]$ s.

Table 2: Trajectory Metrics for Point-Mass CG and 2-Manifold Multirotor Configurations

Configuration	CG Path Length	CG Mean Curve	CG Max Curve	CG Mean Jerk	Vertex Mean Curve	Vertex Max Curve	Vertex Mean Jerk	Vertex Max Jerk
Point-Mass (CG)	10.733	0.49688	0.50000	0.12799	–	–	–	–
Tricopter	10.733	0.49688	0.50000	0.12799	0.49712	0.53651	0.12800	0.12875
Pentacopter	10.733	0.49688	0.50000	0.12799	0.49771	0.57814	0.12802	0.13006
Hexacopter	10.733	0.49688	0.50000	0.12799	0.49808	0.59836	0.12804	0.13087
Octacopter	10.733	0.49688	0.50000	0.12799	0.49902	0.63956	0.12807	0.13184

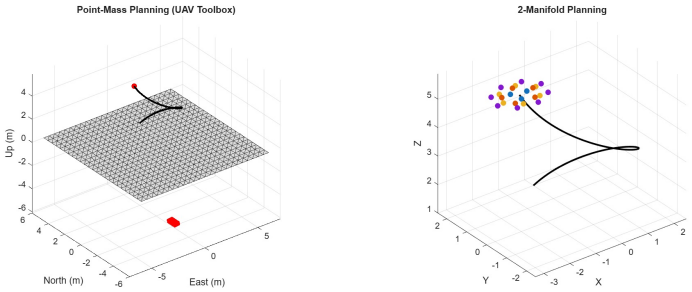


Fig. 10: Comparison of planned trajectories using Point-Mass CG planning (left) and rotor point-set (2-manifold) planning (right). Identical CG motion yields configuration-dependent rotor trajectories.

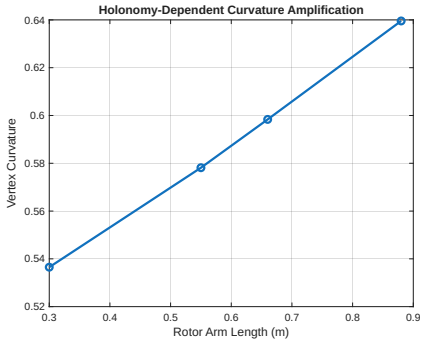


Fig. 11: Maximum vertex curvature amplification with increase in arm length

Figure 10 compares the CG trajectory (left) with the corresponding rotor trajectories (right) for all configurations. While the CG trajectory remains identical by construction, rotor trajectories differ with configuration due to the geometry-induced offset from the CG motion.

Table 2 and Figure 11 show that CG curvature remains nearly invariant (mean ≈ 0.49688, max $= 0.50$), whereas rotor-level curvature increases monotonically with rotor arm length. Specifically, maximum vertex curvature increases from 0.53651 (tricopter, $r = 0.30$ m) to 0.63956 (octacopter, $r = 0.88$ m), and mean vertex curvature increases from 0.49712 to 0.49902. This trend indicates curvature amplification of offset (rotor) trajectories while the CG path curvature remains unchanged.

These results provide numerical evidence consistent with Theorem 1: increasing vehicle geometric extent (larger r) tightens actuator-level feasibility margins for a fixed CG path. Consequently, a path that is admissible for a smaller rotor polygon may not remain admissible for a larger rotor polygon in geometrically constrained environments, even when CG-level planning indicates feasibility.

When represented as a point mass, a MUAV's non-holonomic motion planning problem does not incorporate the geometric constraints of quadcopters. A free-flying rigid body has six degrees of freedom; point-mass planning models capture CG translation but do not represent geometric extent or rotor locations. An MUAV can have an 'n' number of motors, which are control parameters causing an over-actuation or under-actuation problem. In UAVs, the geometric constraints are vital in determining the control, actuation, motion, and representation of the position and orientation of the MUAV. The formulation presented in this article transforms a point mass into a topological manifold. This transformation represents the motion of the MUAV as a differential manifold of a rigid planar polygon (rotor polygon) in an Euclidean space. This formulation incorporates the geometric constraints of non-holonomic motion planning, making this a kinodynamic motion planning problem. The manifolds are control domains. The mapping follows the formulation of the geometric manifold's motion as a third-order polynomial function of time for both forward kinematics and inverse kinematics problems of the MUAV. This model demonstrates computational feasibility as it has a real and distinct solution in forward and inverse kinematics problems.

The analysis indicates that the representation of the MUAV as a manifold with a parametric function enables it to be more sophisticated than point mass motion with differential constraints. The rigid planar polygon (rotor polygon) is homomorphic and satisfies Euclidean isometrics. However, the MUAV's geometric constraints are the motor's position and velocities for a given motion segment. The admissible collision-free path condition of two configurations does not hold for manifolds because they incorporate geometric constraints in planning. At the same time, the proposed formulation exposes configuration-dependent feasibility effects by evaluating actuator-level curvature, acceleration, and jerk profiles that are not visible in CG-only planning. It also generates a manifold for the control manifold.

Nevertheless, the model can reduce the decision-making problem and collision avoidance like a point-mass model because the rigid planar polygon (rotor polygon) is attached to the point mass. The paradox of point mass kinodynamic planning reveals the critical limitations of oversimplifying MUAV path planning by neglecting physical dimensions and holonomic constraints. Incorporating topological manifolds in path planning serves to counter this paradox by ensuring that both kinematic feasibility and physical realizability are considered.

Furthermore, as demonstrated by Theorem 1, the relationship between the physical dimensions of rigid planar polygons (rotor polygons) and their admissible paths highlights the necessity of integrating these factors into the decision-making process for kinodynamic path planning. By addressing the conditional relationship between physical dimensions and path feasibility, the theorem enhances the completeness of kinodynamic path planning, providing a more robust framework for real-world applications where spatial constraints cannot be ignored. Therefore, the point mass motion model's characteristics remain in the kinodynamic modelling. Further work is to identify the optimal control strategy for the kinodynamic model of the MUAVs and deploy these models for autonomous motion planning and control for experimental validations.

3.5 Engineering Implications for UAV Design

The integration of the proposed framework establishes a unified design paradigm for MUAVs. The proposed representation eliminates control singularities arising from actuator redundancy and configuration mismatch by embedding the system dynamics within a geometric mapping that directly relates Cartesian trajectories to feasible actuator commands. This mapping ensures that planned trajectories remain physically realisable at the low-level control layer, thereby maintaining consistency between planning and execution. Unlike conventional approaches that decouple trajectory planning and control design, the present framework emphasises the necessity of a configuration-aware methodology that explicitly accounts for actuator-level realizability during high-level planning. Furthermore, the formulation generalises to multirotor systems beyond conventional quadrotor architectures, enabling applicability to a broader class of UAV configurations. By transforming trajectory planning into a Cartesian-coordinate-based formulation, the framework eliminates the reliance on polar or Euler-angle representations for orientation, which are commonly associated with singularities and ambiguity. In addition to planning and control, the same formulation serves as a principled state-estimation model, where the system state can be updated through observations of the CG position and rigid-body orientation. The manifold-based formulation provides a systematic mechanism for handling actuator redundancy, while the associated geometric mapping extends to arbitrary rigid-body configurations. This capability enables the direct application of the framework to next-generation UAV architectures with non-standard geometries or reconfigurable layouts. From an implementation perspective, the framework can be integrated into existing UAV systems as a modular supervisory layer.

Such integration enables real-time constraint checking, adaptive control allocation, and fault detection without requiring fundamental modifications to existing low-level controllers. Simulation studies and experimental validation of the proposed two-manifold formulation are presented in [27, 28], demonstrating its practical feasibility and performance benefits.

4 Conclusions

This work presented a generalised mathematical framework for MUAV trajectory planning and analysis in which the vehicle is represented as a rigid planar point-set (rotor polygon) embedded in $\mathbb{R}^3$. The formulation explicitly enforces fixed inter-point distance constraints and provides a direct kinematic mapping from CG motion to actuator-level (rotor/vertex) trajectories with solutions in both forward and inverse kinematics. The framework yields state estimation of actuators in forward kinematics and time-dependent trajectory in inverse kinematics at the actuator level in Cartesian coordinates. This mapping enables the evaluation of curvature, acceleration, and jerk at both the CG and rotor points, thereby exposing configuration-dependent feasibility effects that are not visible in conventional point-mass planning abstractions.

Simulation studies demonstrated that multirotor platforms sharing identical CG trajectories can exhibit substantially different actuator-level kinematic profiles as vehicle geometry changes. These results highlight that feasibility and admissibility assessments based solely on CG motion may be insufficient in geometrically constrained environments, and emphasise the advantage of rotor-level metrics during trajectory evaluation and planning. Dynamic characteristics such as actuator limits and vehicle-specific performance envelopes can be incorporated through boundary conditions and feasibility constraints, allowing the same kinematic formulation to be applied across different multirotor configurations.

Further scope of work involves benchmarking studies, and applications will also be conducted to quantify computational cost as a function of vehicle configuration and trajectory complexity. Future work will also involve integration of the proposed framework with ROS/ROS2 and Gazebo-based simulation platforms to incorporate realistic UAV dynamics, control architectures, and environmental interactions.

Declarations

Acknowledgements

The authors acknowledge Dr Abhishek Sarkar of BITS Pilani for his role as coguide and for his constructive criticism. His support and encouragement were so much very instrumentally important in accomplishing this work.

Data and Code Availability

The datasets and custom scripts (Python and MATLAB) developed in support of the findings of this work are openly available in the Figshare repository at `https://doi.org/10.6084/m9.figshare.32120170`.

Clinical Trial Registration

Not applicable.

Consent to Publish

Not applicable.

Ethics Approval and Consent to Participate

Not applicable.

Funding

This research received no external funding.

Competing Interests

The authors declare that they have no competing interests.

References

1. Shubhani Aggarwal and Neeraj Kumar. Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges. *Computer Communications*, 149:270–299, 2020.
2. A. Alaimo, V. Artale, C. Milazzo, A. Ricciardello, and L. Trefiletti. Mathematical modeling and control of a hexacopter. In *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1043–1050, 2013.
3. Andrea Alaimo, Valeria Artale, Cristina Lucia Rosa Milazzo, and Angela Ricciardello. Pid controller applied to hexacopter flight. *Journal of Intelligent & Robotic Systems*, 73(1):261–270, 2014.
4. Khattab M Ali Alheeti, Muzhir Shaban Al-Ani, Abdulkareem Kareem Najem Al-Aloosy, Abdulkareem Alzahrani, and Duaa Abdul Sattar Rukan. Intelligent mobile detection of cracks in concrete utilising an unmanned aerial vehicle. *Bulletin of Electrical Engineering and Informatics*, 11(1):176–184, 2022.
5. Reg Austin. *Unmanned aircraft systems: UAVS design, development and deployment*. John Wiley & Sons, 2011.
6. Luitpold Babel. Three-dimensional route planning for unmanned aerial vehicles in a risk environment. *Journal of Intelligent & Robotic Systems*, 71(2):255–269, 2013.

7. Luitpold Babel. Coordinated target assignment and uav path planning with timing constraints. *Journal of Intelligent & Robotic Systems*, 94(3):857–869, 2019.
8. Zoran Benić, Petar Piljek, and Denis Kotarski. Mathematical modelling of unmanned aerial vehicles with four rotors. *Interdisciplinary Description of Complex Systems - scientific journal*, 14(1):88–100, 1 2016.
9. Hady Benyamen. Stability and control derivatives identification for an unmanned aerial vehicle with low cost sensors using an extended kalman filter algorithm. Master's thesis, University of Kansas, 2019.
10. Yasmina Bestaoui Sebbane. Mission planning. *Intelligent Systems, Control and Automation: Science and Engineering*, 58:45 – 97, 2012.
11. Yasmina Bestaoui Sebbane. *Motion Planning*, pages 59–170. Springer International Publishing, Cham, 2014.
12. Yasmina Bestaoui Sebbane. *Planning and Decision Making for Aerial Robots*, chapter Introduction, pages 1–58. Springer International Publishing, Cham, 2014.
13. Yasser BOUZID, Houria SIGUERDIDJANE, and Yasmina BESTAOUI. Generic dynamic modeling for multirotor vtol uavs and robust sliding mode based model-free control for 3d navigation. In *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 970–979, 2018.
14. Alexandre Filipe Zambujo de Brito. Localization and trajectory control algorithms applied on drones. Master's thesis, Universidade NOVA de Lisboa, 2018.
15. Pin-Chun Cho and Rajan Batta. Uav search path optimization for recording emerging targets. *Military Operations Research*, 26(3):27–48, 2021.
16. Anežka Chovancová, Tomáš Fico, Ľuboš Chovanec, and Peter Hubinsk. Mathematical modelling and parameter identification of quadrotor (a survey). *Procedia Engineering*, 96:172–181, 2014.
17. Reinis Cimurs and Il Hong Suh. Time-optimized 3d path smoothing with kinematic constraints. *International Journal of Control, Automation and Systems*, 18(5):1277–1287, 2020.
18. Civil UAV Assessment Team. Earth observations and the role of uavs: A capabilities assessment. Technical report, NASA, 2006.
19. Redouane Dargham, Adil Sayouti, and Hicham Medromi. Article: Euler and quaternion parameterization in vtol uav dynamics with test model efficiency. *International Journal of Applied Information Systems*, 9(8):25–28, October 2015.
20. Yulong Ding, Bin Xin, and Jie Chen. Precedence-constrained path planning of messenger uav for air-ground coordination. *Control Theory and Technology*, 17(1):13–23, 2019.
21. Egidio D'Amato, Massimiliano Mattei, and Immacolata Notaro. Bi-level flight path planning of uav formations with collision avoidance. *Journal of Intelligent & Robotic Systems*, 93(1):193–211, 2019.
22. Manal S. Esmail, Mohamed H. Merzban, Ashraf A. M. Khalaf, Hesham F. A. Hamed, and Aziza I. Hussein. Attitude and altitude tracking controller for quadcopter dynamical systems. *IEEE Access*, 10:53344–53358, 2022.
23. Marco Esposito. Ai-guided optimal deployments of drone-intercepting systems in large critical areas. In *OVERLAY 2021: 3rd Workshop on Artificial Intelligence and Formal Verification, Logic, Automata, and Synthesis*, page paper 17. CEUR Workshop Proceedings, 2021.
24. Juan Manuel Fortuna-Cervantes, Marco Tulio Ramírez-Torres, José Martínez-Carranza, JS Murguía-Ibarra, and Marcela Mejía-Carlos. Object detection in aerial navigation using wavelet transform and convolutional neural networks: A first approach. *Programming and Computer Software*, 46(8):536–547, 2020.

25. Bernard Friedland and Stephen W. Director. *Control Systems Design: An Introduction to State-Space Methods*. McGraw-Hill Higher Education, 1985.
26. Pablo Garcia-Aunon, Matilde Santos Peñas, and Jesus Manuel de la Cruz García. Parameter selection based on fuzzy logic to improve uav path-following algorithms. *Journal of Applied Logic*, 24:62–75, 2017.
27. Yuvaraj George, Yendluri Venkata Daseswara Rao, and Abhishek Sarkar. A 2-manifold model multi-rotor uav control system. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 47(1), 2025.
28. Yuvaraj George, Yendluri Venkata Daseswara Rao, and Abhishek Sarkar. A 2-manifold topological approach to uav localization and control: Enhancing accuracy and robustness with actuator state estimation. *Engineering reports*, 47(1), 2026.
29. Chad Goerzen, Zhaodan Kong, and Bernard Mettler. A survey of motion planning algorithms from the perspective of autonomous uav guidance. *Journal of Intelligent and Robotic Systems*, 57(1):65–100, 2010.
30. Hamidreza Heidari and Martin Saska. Collision-free trajectory planning of multi-rotor uavs in a wind condition based on modified potential field. *Mechanism and Machine Theory*, 156:104140, 2021.
31. Matthew J. Henchey, Rajan Batta, Mark Karwan, and Agamemnon Crassidis. A flight time approximation model for unmanned aerial vehicles: Estimating the effects of path variations and wind. *Military Operations Research*, 19(1):51–68, 2014.
32. Florian Holzapfel and Stephan Theil. *Advances in aerospace guidance, navigation and control*. Springer, 2011.
33. Yujiao Hu, Yuan Yao, Qian Ren, and Xingshe Zhou. 3d multi-uav cooperative velocity-aware motion planning. *Future Generation Computer Systems*, 102:762–774, 2020.
34. Qikai Huang et al. Mathematical modeling of quadcopter dynamics. *Rose-Hulman Undergraduate Research Publications*, 2016.
35. Zipeng Huang, Robert Bauer, and Ya-Jun Pan. Closed-loop identification and real-time control of a micro quadcopter. *IEEE Transactions on Industrial Electronics*, 69(3):2855–2863, 2022.
36. Moad Idrissi, Mohammad Salami, and Fawaz Annaz. A review of quadrotor unmanned aerial vehicles: Applications, architectural design and control algorithms. *Journal of Intelligent & Robotic Systems*, 104(2):1–33, 2022.
37. Reza N. Jazar. *Path Planning*, pages 569–606. Springer US, Boston, MA, 2007.
38. Oguz Kose and Tugrul Oktay. Dynamic modeling and simulation of quadrotor for different flight conditions. *European Journal of Science and Technology*, 15:132–142, 2019.
39. Rumit Kumar, Mahathi Bhargavapuri, Aditya M. Deshpande, Siddharth Sridhar, Kelly Cohen, and Manish Kumar. Quaternion feedback based autonomous control of a quadcopter uav with thrust vectoring rotors. In *2020 American Control Conference (ACC)*, pages 3828–3833, 2020.
40. Sevkuthan Kurak and Migdat Hodzic. Control and estimation of a quadcopter dynamical model. *Periodicals of Engineering and Natural Sciences*, 6(1):63–75, 2018.
41. Emmanouil Lakiotakis, Pavlos Sermpezis, and Xenofontas Dimitropoulos. Joint optimization of uav placement and caching under battery constraints in uav-aided small-cell networks. In *Proceedings of the ACM SIGCOMM 2019 Workshop on Mobile AirGround Edge Computing, Systems, Networks, and Applications*, MAGESys'19, page 8–14, New York, NY, USA, 2019. Association for Computing Machinery.

42. Jean-Paul Laumond, Sepanta Sekhavat, and Florent Lamiraux. Guidelines in nonholonomic motion planning for mobile robots. *Robot motion planning and control*, pages 1–53, 1998.
43. Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.
44. Jameson Y. S. Lee, Kam K. Leang, and Woosoon Yim. Design and control of a fully-actuated hexrotor for aerial manipulation applications. *Journal of Mechanisms and Robotics*, 10(4):041007, 04 2018.
45. Jeroen A.J. Ligthart, Pakorn Poksawat, Liuping Wang, and Henk Nijmeijer. Experimentally validated model predictive controller for a hexacopter**the authors gratefully acknowledge the partial sponsorship by dstg, australia. *IFAC-PapersOnLine*, 50(1):4076–4081, 2017. 20th IFAC World Congress.
46. Teppo Luukkonen. Modelling and control of quadcopter. *Independent research project in applied mathematics, Espoo*, 22:22, 2011.
47. Blanca López, Javier Muñoz, Fernando Quevedo, Concepción A. Monje, Santiago Garrido, and Luis E. Moreno. Path planning and collision risk management strategy for multi-uav systems in 3d environments. *Sensors*, 21(13), 2021.
48. B. Ma and J. Yan. A theoretical path computation model for rectangular rigid-bodies entering an entrance. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 10(4/W3-2022):159 – 164, 2022.
49. Richard E Maine and Kenneth W Iliff. Application of parameter estimation to aircraft stability and control: The output-error approach. Technical Report NASA Ref. 1168, NASA, 1986.
50. Vandana Mansur, Srinivasulu Reddy, Sujatha R, and R. Sujatha. Deploying complementary filter to avert gimbal lock in drones using quaternion angles. In *2020 IEEE International Conference on Computing, Power and Communication Technologies (GUCON)*, pages 751–756, 2020.
51. R. Ryan McCune and Gregory R. Madey. Swarm control of uavs for cooperative hunting with dddas. *Procedia Computer Science*, 18:2537–2544, 2013. 2013 International Conference on Computational Science.
52. Daniel Mellinger. *Trajectory generation and control for quadrotors*. PhD thesis, University of Pennsylvania, 2012.
53. R.V. Meshcheryakov, P.M. Trefilov, A.V. Chekhov, S.A.K Diane, K.D. Rusakov, E.A. Lesiv, M.A. Kolodochka, K.O. Shchukin, A.K. Novoselskiy, and E. Goncharova. An application of swarm of quadcopters for searching operations. *IFAC-PapersOnLine*, 52(25):14–18, 2019.
54. Alaeddin Milhim, Youmin Zhang, and Camille-Alain Rabbath. Gain scheduling based pid controller for fault tolerant control of quad-rotor uav. In *AIAA infotech@ aerospace 2010*, page 3530. AIAA inc.,, 2010.
55. Andreas Mueller, Jozsef Kovecses, Charles Kim, Chandramouli Padmanabhan, and Gabor Orosz. Joint special issue: Design and control of responsive robots. *Journal of Mechanisms and Robotics*, 15(3):030301, 05 2023.
56. Minh T. Nguyen, Linh H. Truong, and Trang T.H. Le. Video surveillance processing algorithms utilizing artificial intelligent (ai) for unmanned autonomous vehicles (uavs). *MethodsX*, 8:101472, 2021.
57. Hannes Ovrén and Per-Erik Forssén. Trajectory representation and landmark projection for continuous-time structure from motion. *The International Journal of Robotics Research*, 38(6):686–701, 2019.
58. Abraham J Pachikara, Joseph J Kehoe, and Rick Lind. Path-parameterization approach using trajectory primitives for three-dimensional motion planning. *Journal of Aerospace Engineering*, 26(3):571–585, 2013.

59. M. Pachter and C.H. Houpis. *Electropneumatic and electrohydraulic instruments: Modeling*. CRC Press, 2017.
60. Christoph Peukert, Patrick Pöhlmann, Marcel Merx, Jens Müller, and Steffen Ihlenfeldt. Investigation of local and modal based active vibration control strategies on the example of an elastic system. *Journal of Machine Engineering*, 19(2):32 – 45, 2019.
61. Ramiro Ibarra Pérez, Gerardo Romero Galvan, Aldo Jonathan Muñoz Vázquez, Silvia Florida Melo, and David Lara Alabazares. Attitude control of a quadcopter using adaptive control technique. In Le Anh Tuan, editor, *Adaptive Robust Control Systems*, chapter 6. IntechOpen, Rijeka, 2017.
62. Bartomeu Rubí, Ramón Pérez, and Bernardo Morcego. A survey of path following control strategies for uavs focused on quadrotors. *Journal of Intelligent & Robotic Systems*, 98:241–265, 2020.
63. Iman Saboori, M.B. Menhaj, Tehran Iran, and Bahram Karimi. Optimal robot path planning based on fuzzy model of obstacles. In *IECON Proceedings (Industrial Electronics Conference)*, page 383 – 387, 2006.
64. Bhaskar Saha, Cuong C. Quach, and Kai Goebel. Optimizing battery life for electric uavs using a bayesian framework. In *2012 IEEE Aerospace Conference*, pages 1–7, 2012.
65. Anand Sanchez-Orta, Vicente Parra-Vega, Carlos Izaguirre-Espinosa, and Octavio Garcia. Position–yaw tracking of quadrotors. *Journal of Dynamic Systems, Measurement, and Control*, 137(6):061011, 06 2015.
66. Jerzy Z Sasiadek and Ignacy Dulęba. 3d local trajectory planner for uav. *Journal of Intelligent and Robotic Systems*, 29(2):191–210, 2000.
67. Ivana Yoshie Sumida, Francisco Haroldo, Campos Velho, Eduardo Fávero Pacheco da Luz, Ronaldo Vieira Cruz, Luiz Amorim Carlos, and Sandoval Góes. Flight dynamics parameter estimation of a rotary wing aircraft using theoutput error minimization with natural and meta heuristic methods. In *Proccedings of 1st Pan-American Congress on Computational Mechanics – PANACM 2015*, 2015.
68. Marco Tognon and Antonio Franchi. *Theory and Applications for Control of Aerial Robots in Physical Interaction Through Tethers*, volume 140. Springer Nature, 2020.
69. Mohamed Tolba and Bijan Shirinzadeh. Generic modeling and control of unbalanced multirotor uavs. *Aerospace Science and Technology*, 121:107394, 2022.
70. Huu Khoa Tran, Juing-Shian Chiou, and Viet-Hung Dang. New fusion algorithm-reinforced pilot control for an agricultural tricopter uav. *Mathematics*, 8(9):1499, 2020.
71. Luis Verde-Star. Inverses of generalized vandermonde matrices. *Journal of mathematical analysis and applications*, 131(2):341–353, 1988.
72. Xueyuan Wang and M. Cenk Gursoy. Learning-based uav trajectory optimization with collision avoidance and connectivity constraints. *IEEE Transactions on Wireless Communications*, 21(6):4350–4363, 2022.
73. Yebin Wang, Devesh K. Jha, and Yukiyasu Akemi. A two-stage rrt path planner for automated parking. In *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*, pages 496–502, 2017.
74. Christian Zammit and Erik-Jan Van Kampen. *3D real-time path planning of UAVs in dynamic environments in the presence of uncertainty*, chapter Session: Motion Planning, Sensing, and Operations for Aerospace Robotic Systems I. AIAA Inc.,, 2021.

75. Jian Zhang, Jianbo Hu, Juyu Lian, Zongji Fan, Xuejun Ouyang, and Wanhui Ye. Seeing the forest from drones: Testing the potential of lightweight drones as a tool for long-term forest monitoring. *Biological Conservation*, 198:60–69, 2016.
76. Xiaodong Zhang, Xiaoli Li, Kang Wang, and Yanjun Lu. A survey of modelling and identification of quadrotor robot. *Abstract and Applied Analysis*, 2014:16, 2014.
77. Yongkun Zhou, Bin Rao, and Wei Wang. Uav swarm intelligence: Recent advances and future trends. *IEEE Access*, 8:183856–183878, 2020.