Ain Shams Engineering Journal 16 (2025) 103301

Contents lists available at ScienceDirect

Ain Shams Engineering Journal

journal homepage: https://www.sciencedirect.com

Full Length Article

Cognitive strategies for UAV trajectory optimization: Ensuring safety and energy efficiency in real-world scenarios

Mushtaq Muhammad Umer [a,*], Hein Venter [a,*], Owais Muhammad [b], Tamoor Shafique [c], Fuad A. Awwad [d], Emad A.A. Ismail [d]

[a] Department of Computer Science, University of Pretoria, Pretoria, South Africa
[b] School of Information Engineering, SWUST, Mianyang, China
[c] School of Digital Technology, Innovation and Business, University of Staffordshire, United Kingdom
[d] Department of Quantitative Analysis, College of Business Administration, King Saud University, P.O.Box 71115, Riyadh 11587, Saudi Arabia

ARTICLE INFO

Keywords:
Energy-efficient trajectory planning
Unmanned aircraft vehicles (UAVs)
Obstacle avoidance
Optimization
Real-time applications

ABSTRACT

Many sectors in aerial transportation use unmanned aircraft vehicles (UAVs) extensively. This becomes even more challenging in complex environments where not only it is required to avoid obstacles, but it also must be maintained for a prolonged period of time. This paper presents a novel approach to increase UAV autonomy through safe and efficient flight trajectory design. An optimization problem is formulated with external and internal safety constraints, and traversing collision free paths. The proposed work offers an energy efficient RRT algorithm, which is used to assess multiple trajectory alternatives. The simulation results confirm the achieved performance in finding the optimal energy path while obeying to the safety constraint. The data and performance metrics, show the system operated in a safe and energy efficient manner. This work provides a unified framework for UAV trajectory planning that guarantees a trade-off between safety and energy efficiency.

1. Introduction

Unmanned Aircraft Vehicles (UAVs) play a critical role in modern aerial transportation, offering versatile applications across industries such as logistics, surveillance, and disaster management. However, their operation in complex environments presents unique challenges, including the need to avoid obstacles and ensure prolonged, efficient operation. These challenges are particularly pronounced in scenarios requiring both safety and energy efficiency. This paper focuses on addressing these issues by proposing a novel approach to UAV trajectory design that optimizes flight paths to achieve a balance between safety and energy consumption. UAVs are also able to work where humans might be reluctant or even unable to go. Hence, they are breathing new life into many different forms of our everyday work. UAVs have been found particularly effective in facility inspection [1]; without UAVs, human inspectors may have to climb to great heights or use expensive equipment to assess the structural integrity of bridges, tall buildings, and other critical infrastructure. UAVs, being multi-faceted sensor platforms [2] fitted with thermal imaging cameras and GPS (Global positioning system) enable rapid searches for hunting persons in remote or disaster-hit areas. These features significantly reduce response time and increase the likelihood of a successful rescue. Also UAVs that are used in aerial transportation are mainly highlighted with respect to its application as a means of transportation of medical devices, food supply and other necessary resources to remote locations [3]. The fact that these systems are designed and operated with a wide range of characteristics has also contributed to their versatility. Because they are able to take off and land vertically, hover in place and fly through narrow rooms [4]; which means, their low cost and ability to fit in diverse payloads makes them scalable and amenable to mass adoption and since then they are used in several application segments, right from agriculture to defense & logistics to mining & monitoring the environment. They do however pose many technical challenges when trying to navigate UAVs through complex environments with a high level of clutter. The advanced navigation capability is one of the fundamental UAV requirements in such environment [5]. Obstacle detection and avoidance is critical for safe navigation in autonomous agents, particularly in dynamic and unpredictable environments where the position and behavior of obstacles may change in the way that the agent could not have accounted for in advance. Autonomous vehicles use a combination of different classes of advanced

* Corresponding authors.
E-mail address: mu.mushtaq@up.ac.za (M.M. Umer).

https://doi.org/10.1016/j.asej.2025.103301
Received 21 December 2024; Received in revised form 15 January 2025; Accepted 31 January 2025

Available online 6 February 2025

sensors such as LiDAR (Light Detection and Ranging) [6], ultrasonic sensors [7], and cameras [8], along with algorithms that help in obstacle detection and avoidance. Apart from planning the collision-free movement, UAVs need to go through the process of generating a feasible path and trajectory based on task performance requirements. Example of a use case could be structural inspections, where the UAV may need to hover as still as possible at certain angles and distances from the target to log detailed data [9]. Modern UAVs are characterized by advanced controls systems due to their challenge of complexity and uncertainty and adaptive systems being designed [10], which allow them to operate successfully in dynamic environments. Throttle settings, yaw, roll and pitch are three axes for which more advanced stabilization algorithms can respond to quick changes in inertia due to forces like turbulence, wind shear, etc. Drones that have a weather insulation casing and are coated with water-repellant offer relatively good operation in wet conditions as well. This adds a degree of redundancy to the motors as in multi-rotor drone, if one motor fails due to the mechanical or operating condition, the UAV is still fit to fly and can complete the mission. From real-time weather sensors to predictive AI models [11], emerging technologies are increasingly giving UAVs advanced environmental adaptability and allowing them to operate more safely and reliably in complex scenarios.

Aerial energy efficiency is definitely essential in the way of the UAV flight duration distance range and task adding success [12]. Most UAVs are battery-driven, and while such technology is already highly developed, it can indeed pose a problem being constrained by the energy density achievable by commercial battery series. These limitations are problematic for long-duration flights or missions in computed cluttered environments where speed, altitude and trajectory have to be updated frequently to avoid hazards. These maneuvers result in a higher energy demand and a worse flight endurance and mission effectiveness of the UAV. In pursuit of this goal, researchers and engineers are developing energy-efficient navigation patterns oriented on the particularities of UAV needs [13][14]. New Improved algorithms, encoded to discern the least-cost path, considering everything from wind conditions to hindrances and elevations. UAVs are also designed for hovering maneuvers to enable them to perform stationary operations like inspections or surveillance with the least amount of energy consumption [1]. The potential of energy harvesting techniques as an innovative way to increase UAV flight durations is mentioned in [15]. UAV frames can be embedded with solar panels which can be used to harvest solar energy when UAV needs to operate in day. For example, UAV energy harvesting methods involve the usage of piezoelectric materials embedded in UAV structures [16] to scavenge energy from mechanical vibrations, regenerative braking, which transfers to the UAV during braking or descending, and others in the similar domains. Such improvements provide additional power supply and reduce reliance on batteries, thus extending mission duration where energy is a constraint. Battery technology improvements are an essential element of the solutions for UAV energy efficiency issues [17]. Modern lithium-ion and lithium-polymer batteries have higher energy density, since this means longer flight time without backpacking weight. Next-generation battery chemistries (like solid-state batteries or lithium-sulfur batteries) are also studied in [18]. Moreover; hybrid power systems that integrate batteries with fuel cells or small internal combustion engines are also being examined to provide increased endurance over 5 continuum for certain applications [19]. The use of lightweight materials has also helped in energy efficiency by lowering the weight of UAVs [20]. The composite materials being used extensively in UAV frame structures of many UAVs, Carbon fiber, advanced composite polymers and similar materials which are beneficial in the sense that it allows to carry a heavier payload or energy store without a relatively high amount of energy consumption. These materials also help UAV's aerodynamic design as they reduce the drag and thus increase energy efficiency further.

The development of autonomous capabilities has grown to be a major focus of UAV research, not least for the reasons that complex operations necessitate higher efficiency, reliability, and versatility [21]. These are UAVs that have software with AI (Artificial Intelligence) and ML (Machine Learning) algorithms such as Support Vector Machines (SVM), Convolutional Neural Networks (CNN) and reinforcement learning to improve decision-making in dynamic environments [22][23][24][25]. Similarly, Swarm Intelligence Optimization methods, including Genetic Algorithms (GA) [26] and Squirrel Flying Optimization (SFO) [27] have demonstrated effectiveness in solving complex optimization problems. While these algorithms demonstrate remarkable performance across various domains, this work specifically focuses on an energy-efficient variant of the Rapidly-exploring Random Tree (RRT) algorithm, meticulously designed for UAV trajectory optimization. Unlike traditional ML techniques that often rely on extensive training datasets or swarm-based methods that can be computationally intensive, the proposed RRT algorithm operates with high computational efficiency and adaptability. It excels in dynamically generating collision-free paths in complex environments, integrating both internal (energy consumption) and external (obstacle avoidance) constraints in real time. These capabilities make the energy-efficient RRT algorithm particularly advantageous for UAV applications where rapid decision-making, constrained resources, and real-time adaptability are critical.

Various path planning algorithms have been investigated in the broader literature so that a UAV can avoid a crash in dynamic and cluttered environment quickly and safely. Aerospace mobility aids such algorithms for UAVs, as it allows to navigate in an environmental area that contains natural and artificial objects, complying with the limitations of their physical and operational capabilities. One of the most prominent among these techniques are grid-based techniques which divide the environment into discrete cells which only recalculate potential paths that would be feasible for the UAV between its initial and target points [28]. Grid-based approaches are effective for structured environments, but can become computationally intensive in large-scale or extremely cluttered environments [29]. Sampling-based methods (e.g., Rapidly-exploring Random Trees (RRT) [30][31], Probabilistic Roadmaps (PRM)) [32] are particularly advantageous, given their ability to manage high dimensionality. These techniques essentially find paths by sampling random points in the environment, which results in a network of possible pathways that do not fatally collide. One example is RRT which rapidly explores the search space and works well for real-time cases whilst PRM focuses on covering the feasible paths which can be reused over several queries. A second type of advanced methodologies for UAV trajectory planning is optimization-based algorithms [33], which use algorithms from Mixed-Integer Linear Programming (MILP). A mixture of features of these algorithms is often adopted to strike a balance between computational efficiency and robustness. You could use contracting methods to generate a solution by way of sampling-based methods and subsequent optimization, better accuracy. In addition, because of AI and machine learning-based technologies, UAV could not only realize environment elements and changes of the environment but also propose a new method of path planning with the help of each previous path plan [34].

In dynamic environment avoiding obstacles, traditional reactive navigation strategies like Artificial Potential Field (APF) and Dynamic Window Approach (DWA) have been used [35]. These techniques make it possible to modify flight paths in real time while still ensuring safe operations. While these methods have been effective, corollary problems such as local minima and computation overhead are still problematic [36], potentially necessitating the use of more sophisticated heuristics and machine learning. Furthermore, sensor fusion strategies to enhance situational understanding also form a cornerstone of UAV investigation [37], rendering them adept and functional within arduous environments. Considering the strengths and weaknesses of various sensors, the use of a combination of LiDAR, cameras, ultrasonic sensors, infrared sensors, and IMU can enable UAVs to achieve a comprehensive and accurate perception of the environment. For example, LiDAR enables efficient three-dimensional mapping at high resolution based on distance measurement by laser pulse that equips UAVs to detect obstacles and create environment models in higher detail. An ultrasonic sensor can also be an alternative within the program which will be used to measure the distance accurately in nearby as well as in smoky/rainy conditions in which cameras can lose here visual input needed for object and terrain detection. Combining this data from various sensors allows to offset their individual strengths and weaknesses, ensuring that the environment is mapped simultaneously as a single environment model. LiDAR can enable precise obstacle detection in low light conditions where cameras might struggle [38]. On the other hand, when reflective surfaces cause problems for LiDAR, cameras and ultrasonic sensors can offer additional information. By combining all this heterogeneous sensor data, the UAVs can undergo precise path planning [39], dynamic obstacle avoidance [40], and real-time decision-making [41], even in dense or unpredictable environments. While GPS is generally sufficient for navigation in open areas, the deployment of SLAM (Simultaneous Localization and Mapping) algorithms [42][43] has greatly expanded the operations of UAVs in GPS-denied environments, like indoors, underground, or in dense forests. Because, SLAM algorithms can simultaneously enable UAVs to build a map of an unknown environment while estimating their locations [44], making it possible to navigate autonomously in environments without or with unreliable GPS signals. Visual SLAM and LiDAR-based SLAM have proven effective, and the first uses only camera data to recognize landmarks, while the latter provides accurate 3D maps. However, sensor fusion also enhances SLAM by integrating it with sensors to enable UAVs to be applied in multiple applications [45]. UAVs outfitted with these technologies can inspect machinery, check inventory, and fly in tight spaces in indoor industrial environments. In disaster response situations like collapsed buildings, they can generate detailed maps to direct rescue teams. Furthermore, developments in edge computing and AI have improved SLAM's ability to process information in real-time, enabling UAVs to adapt to evolving environments and maintain situational awareness amid high-velocity flights.

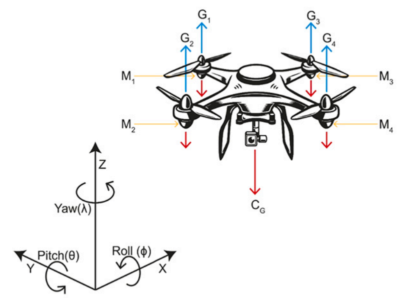
While balanced representations are more amenable to energy conservation, they still suffer from the limitations of conventional hybrid quantum-classical methods. Future lines of research are linked to the use of artificial intelligence for autonomous decision-making, and/or lightweight hardware to implement complex algorithms without overloading the UAV.

1.1. Contributions

The following is a summary of this paper's substantial achievements:

- Proposed an energy-efficient trajectory planning framework that significantly enhances UAV flight endurance by optimizing energy consumption during operation.
- Developed an optimization model incorporating internal and external safety constraints, ensuring collision-free paths while maintaining actuator performance limits.
- Introduced an improved RRT algorithm for effective obstacle avoidance in complex environments, validated through simulations across diverse scenarios.

The rest of the paper is as follows. The section 2 provides a detailed introduction and description of the proposed materials and methods. The problem formulation and presentation of different trajectory planning methodologies is discussed and elaborated in section 3. Simulation and analysis is explained in the section 4 while the conclusions can be accessed in the section 5.



$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} \frac{1}{m} & \frac{1}{m} & \frac{1}{m} & \frac{1}{m} \\ \frac{-d}{y_1} & \frac{-d}{y_1} & \frac{d}{y_1} & \frac{d}{y_1} \\ \frac{-d}{y_2} & \frac{d}{y_2} & \frac{d}{y_2} & \frac{-d}{y_2} \\ \frac{T}{y_3} & \frac{-T}{y_3} & \frac{T}{y_3} & \frac{-T}{y_3} \end{bmatrix} \begin{bmatrix} G_1 \\ G_2 \\ G_3 \\ G_4 \end{bmatrix}$$

$$\begin{cases} v_{1max} = \frac{4}{m} G_{max}; \\ v_{2max} = \frac{2d}{y_1} G_{max}; \\ v_{3max} = \frac{2d}{y_2} G_{max}; \\ v_{3max} = \frac{2T}{y_3} G_{max}; \end{cases}$$

Fig. 1. Quadrotor UAV system model.

2. Materials and methods

In this study, a widely recognized quadrotor UAV model is utilized and then characterized by four motors positioned at the extremities of a cross-shaped frame. The proposed system model is demonstrated in Fig. 1. The analysis begins with a simplified deterministic model that serves as a foundation for understanding the more intricate and uncertain dynamics of real-world systems:

$$\begin{aligned} \ddot{a} &= v_{1_a} - \frac{x_1}{m}\dot{a}; \;\; \ddot{\theta} = v_2 - \frac{d x_4}{y_1}\dot{\theta} \\ \ddot{b} &= v_{1_b} - \frac{x_2}{m}\dot{b}; \;\; \ddot{\phi} = v_3 - \frac{d x_5}{y_2}\dot{\phi} \\ \ddot{c} &= -g + v_1(\cos\phi\cos\theta) - \frac{x_3}{m}\dot{c} \\ \ddot{\lambda} &= v_4 - \frac{d k_6}{y_3}\dot{\lambda} \end{aligned} \tag{1}$$

Where:
$v_{1a} = v_1(\cos(\phi)\sin(\theta)\cos(\lambda) + \sin(\phi)\sin(\lambda))$;
$v_{1b} = v_1(\cos(\phi)\sin(\theta)\cos(\lambda) - \sin(\phi)\cos(\lambda))$;

a, b and c represent the coordinates of the quadrotor centre of gravity in the earth frame. The pitch, roll and yaw angles are represented by, θ, ϕ and λ respectively. Here m is the mass, and y_i are the moments of inertia, where $(i = 1, 2, 3)$ along the directions a, b and c respectively. The parameters with drag... $x_i(1....6)$, where... d describes the distance between the centre of gravity and each of the rotor. v_1 is the applied linear acceleration to quadrotor that is applied in z-direction of frame of the body. v_2, v_3 and v_4 are, the angular accelerations induced in the θ, ϕ and λ directions respectively. The variable g represents the acceleration due to gravity.

Although this representation of the UAV model is indeed a simplified version, it remains sufficiently complex to pose significant challenges for practical implementation and analysis. Therefore, various simplification techniques are explored that will streamline our approach without compromising the robustness and effectiveness of the controller that is intended to implement. By assuming small angles i.e. θ and ϕ and a constant yaw angle $\lambda = 0$, the equation (1) can be rewritten as:

$$\begin{aligned} \ddot{a} &= v_1\theta - \frac{x_1}{m}\dot{a}; \;\; \ddot{\theta} = v_2 - \frac{d x_4}{y_1}\dot{\theta} \\ \ddot{b} &= v_1\phi - \frac{x_2}{m}\dot{b}; \;\; \ddot{\phi} = v_3 - \frac{d x_5}{y_2}\dot{\phi} \\ \ddot{c} &= -g + v_1 - \frac{x_3}{m}\dot{c} \\ \ddot{\lambda} &= v_4 - \frac{d k_6}{y_3}\dot{\lambda} \end{aligned} \tag{2}$$

If negligible drag coefficients at low speeds are considered:

$$\begin{aligned} \ddot{a} &= v_1\theta;\ \ddot{\theta} = v_2 \\ \ddot{b} &= v_1\phi;\ \ddot{\phi} = v_3 \\ \ddot{c} &= -g + v_1 \\ \ddot{\lambda} &= v_4 \end{aligned} \tag{3}$$

According to the above equations, it can be assumed that in the hovering condition, $v_1 \approx g$, such that:

$$\begin{aligned} \ddot{a} &= g\theta;\ \ddot{\theta} = v_2 \\ \ddot{b} &= -g\phi;\ \ddot{\phi} = v_3 \\ \ddot{c} &= -g + v_1 \\ \ddot{\lambda} &= v_4 \end{aligned} \tag{4}$$

So the flatness-based trajectory planning problem is basically amount of thrust that need to be applied to first order trajectory. This characteristic is important for giving the quadrotor accurate speed on the intended trajectory. But the basic model makes it very difficult to apply this model for these purposes, because the computations it requires are quite heavy, and can be time-consuming making real-time applications hard. In order to ease these difficulties, a number of simplifications have been introduced that have resulted in a model that is significantly less complicated to work within that is both computationally efficient and capable of achieving the control objectives.

Before continuing with the detailed exposition of the control strategy that is to be applied, it is important to first examine the relationship between the quadrotor accelerations and the thrusts produced by its rotors. Grasping this relationship is crucial for creating a stout control mechanism that can adapt to changing flight conditions and trajectory needs. Therefore, the following relation is obtained:

$$\begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{pmatrix} = \begin{pmatrix} \frac{1}{m} & \frac{1}{m} & \frac{1}{m} & \frac{1}{m} \\ \frac{-d}{y_1} & \frac{-d}{y_1} & \frac{d}{y_1} & \frac{d}{y_1} \\ \frac{-d}{y_2} & \frac{d}{y_2} & \frac{d}{y_2} & \frac{-d}{y_2} \\ \frac{T}{y_3} & \frac{-T}{y_3} & \frac{T}{y_3} & \frac{-T}{y_3} \end{pmatrix} \begin{pmatrix} G_1 \\ G_2 \\ G_3 \\ G_4 \end{pmatrix} \tag{5}$$

Where $G_i (i = 1, \ldots 4)$ is the thrust generated by i-th rotor and T is the thrust-to-moment scaling factor. To enhance the clarity and simplicity of the problem, it is essential to normalize the vectors within this matrix. This process involves dividing each component of the vectors by their respective maximum values, resulting in dimensionless numbers. By doing so, each component is transformed to a scale between 0 and 1, thereby facilitating a more straightforward analysis and solution of the problem. Let's assume that $G_{1\max} = G_{2\max} = G_{3\max} = G_{4\max} = G_{\max}$, then the maximum value of the input v_i is mentioned in the following equation:

$$\begin{cases} v_{1\max} = \dfrac{4}{m} G_{\max}; \\ v_{2\max} = \dfrac{2d}{y_1} G_{\max}; \\ v_{3\max} = \dfrac{2d}{y_2} G_{\max}; \\ v_{4\max} = \dfrac{2T}{y_3} G_{\max}; \end{cases} \tag{6}$$

Now the normalized relation is:

$$\begin{pmatrix} \frac{v_1}{v_{1\max}} \\ \frac{v_2}{v_{2\max}} \\ \frac{v_3}{v_{3\max}} \\ \frac{v_4}{v_{4\max}} \end{pmatrix} = \begin{pmatrix} +\frac{1}{4} & +\frac{1}{4} & +\frac{1}{4} & +\frac{1}{4} \\ -\frac{1}{2} & -\frac{1}{2} & +\frac{1}{2} & +\frac{1}{2} \\ -\frac{1}{2} & +\frac{1}{2} & +\frac{1}{2} & -\frac{1}{2} \\ +\frac{1}{2} & -\frac{1}{2} & +\frac{1}{2} & -\frac{1}{2} \end{pmatrix} \begin{pmatrix} G_1 \\ G_2 \\ G_3 \\ G_4 \end{pmatrix} \tag{7}$$

This normalization process provided the big advantage of removing all physical parameters from the expression, embedding a clear separation in the analysis and simplifying the mathematical structure. This allowed to look at the system itself without getting distracted by continuously dealing with units. This contributes to evolution of equations and facilitates compared and interpretation of results in various cases and conditions, thus arriving to a universal interpretation of the system behavior.

3. Problem formulation

3.1. Flatness-based control

But there are going to have many different flavors of the core idea (control inputs as functions of the trajectory we want our system to track) for flatness based trajectory planning. What can be reconstructed, that is sufficiently unique of the outputs, within the scope of the control problem that can be engineered, and how it can be treated as the flat outputs of the system? The simplest definition of the flatness property means that all the system's states and inputs can be expressed as a dynamic output plus its derivatives for a specific dynamical system. Therefore, the trajectory planning and control design is greatly alleviated because the complicated dynamics is transformed into simple one.

$$\dot{a} = f(a, v);\ \ b = h(a); \tag{8}$$

Where $a \in R^n$ and $v \in R^m$, is flat if and only if there exists variables $F \in R$, which are called the flat outputs. This control approach is fundamental for the system's practical implementation: in fact, the parametrization of the control inputs v in terms of the flat outputs F has a pivotal role in the trajectory planning problem, where the nominal control inputs that will be applied during the mission can be described as a function of the desired trajectories. In our case, the flat outputs that we have selected are:

$F_1 = c; F_2 = a; F_3 = b; F_4 = \lambda (= 0$ as assumed in model simplification);

From these assumptions, following equation shows the parametrization of the control inputs and θ and ϕ in functions of the flat outputs:

$$\begin{aligned} \theta &= \frac{\cos F_4 \left(\ddot{F}_2 + \frac{x_1}{m} \dot{F}_2 \right) + \sin F_4 \left(\ddot{F}_3 + \frac{x_2}{m} \dot{F}_3 \right)}{\ddot{F}_1 + \frac{x_3}{m} \dot{F}_1 + g}; \\ \phi &= \frac{\sin F_4 \left(\ddot{F}_2 + \frac{x_1}{m} \dot{F}_2 \right) - \cos F_4 \left(\ddot{F}_3 + \frac{x_2}{m} \dot{F}_3 \right)}{\ddot{F}_1 + \frac{x_3}{m} \dot{F}_1 + g}; \end{aligned} \tag{9}$$

To finalize the flatness-based control algorithm, it only remains to select the reference trajectories F_i^*. To get smooth control inputs, the degree 5 Bezier polynomial function is used for F_1, and F_4, and degree 9 for F_2 and F_3, so that the reference trajectories are:

$$\begin{aligned} F_i^* &= f_5(t) = a_5^i t^5 + a_4^i t^4 + a_3^i t^3 + a_2^i t^2 + a_1^i t + a_0 (i = 1, 4); \\ F_i^* &= f_4(t) = a_9^i t^9 + a_8^i t^8 + a_7^i t^7 + a_6^i t^6 + a_5^i t^5 + \\ &\quad a_4^i t^4 + a_3^i t^3 + a_2^i t^2 + a_1^i t + a_0 (i = 2, 3); \end{aligned} \tag{10}$$

where, t is the time and a_i's are the constant coefficients. This type of control approach has been chosen to allow for the possibility to reverse design the parameters that determine the desired trajectory so that actuator constraints are not violated. The reason Bezier Polynomials are selected, since they provide a simple way of computing the control inputs to be applied along the trajectory. Indeed, these mentioned trajectory planning features in this paragraph will become clearer in the next sections, specially when there is a need to deal with the implementation. This phenomenon is referred as "internal safety".

3.2. Minimal energy trajectory planning (METP)

The three main constraints of this trajectory planning problem, covered in previous sections here, are: internal safety, external safety and minimum-energy. In the next paragraphs, it can also be seen that the realization of the third element is actually inherent in the overall control methodology. Indeed, as the implementation begins with the definition of a system capable of performing a trajectory with minimal energetic expenditure, it is imperative to formalize the problem in mathematical terms, essentially as a cost function.

$$
\begin{aligned}
T_1^* &= \frac{v_1^*}{v_{1\max}} - \frac{1}{2}\frac{v_2^*}{v_{2\max}} - \frac{1}{2}\frac{v_3^*}{v_{3\max}} + \frac{1}{2}\frac{v_4^*}{v_{4\max}};\\
T_2^* &= \frac{v_1^*}{v_{1\max}} - \frac{1}{2}\frac{v_2^*}{v_{2\max}} + \frac{1}{2}\frac{v_3^*}{v_{3\max}} - \frac{1}{2}\frac{v_4^*}{v_{4\max}};\\
T_3^* &= \frac{v_1^*}{v_{1\max}} + \frac{1}{2}\frac{v_2^*}{v_{2\max}} + \frac{1}{2}\frac{v_3^*}{v_{3\max}} + \frac{1}{2}\frac{v_4^*}{v_{4\max}};\\
T_4^* &= \frac{v_1^*}{v_{1\max}} + \frac{1}{2}\frac{v_2^*}{v_{2\max}} - \frac{1}{2}\frac{v_3^*}{v_{3\max}} - \frac{1}{2}\frac{v_4^*}{v_{4\max}};
\end{aligned}
\tag{11}
$$

Where, $u_{i_{\max}}$ are constants and u_i^* are the nominal control inputs that are expressed as:

$$v_1^* = \ddot{F}_1^* + g;\ v_2^* = \tfrac{1}{g}F_2^{4*};\ v_3^* = -\tfrac{1}{g}F_3{}^{4*};\ v_4^* = \ddot{F}_4^*$$

3.2.1. METP-internal safety

The optimization problem is still not defined completely: in fact, the F_i^* profile can be adjusted to guide the system from an initial to a final position, without violating the actuator constraints. That is basically what is referred to as "Internal Safety". In fact, the actuator constraints allow to reformulate the optimization problem as follows:

$$
\begin{cases}
\text{Minimize} & E(t_f) - E(t_0)\\
\text{Subject to} & 0 \le G_i^* \le 1 \quad (i = 1, \dots, 4)
\end{cases}
\tag{12}
$$

The nominal thrusts are in function of time t, thus solving the problem requires using calculus of variations which may impose heavy calculations. This problem can be overcome by considering only the extrema of the nominal thrusts, such that the control problem becomes:

$$
\begin{cases}
\text{Minimize} & E(t_f) - E(t_0)\\
\text{Subject to} & 0 \le G_{i_{\text{Ex}}}^* \le 1 \quad (i = 1, \dots, 4)\\
 & 0 \le G_i^*(t_0) \le 1\\
 & 0 \le G_i^*(t_f) \le 1
\end{cases}
\tag{13}
$$

where it is evident that the last two rows of the constraints indicate the initial and final conditions to be applied to the thrusts values, whereas, $G_{i_{\text{Ex}}}^*$ indicates the extrema of $G_i{}^*$. The extrema are in function of initial and final time (the only unknown is the final time). They can be written as $G_{i_{\text{Ex}}}^* = G_i{}^*(c_i{}^*)$, where $c_i{}^*$ are the critical points. These latter are the solutions of the derivative of $G_i{}^*$ with respect to time equalized to zero. From this approach, $G_{i_{\min}}^*$ can be defined and $G_{i_{\max}}^*$ as the global minima and global maxima such that:

$$G_{i_{\min}}^* \le G_i^* \le G_{i_{\max}}^* \quad \forall t \in [t_0, t_f] \tag{14}$$

It demonstrates that this approach leads to a solution of the control problem as in the first formulation but providing much less computations. It is necessary to highlight that the derivative of $G_i{}^*$ with respect to time equal to zero has a degree not less than five. This way, $G_i{}^*$ is a polynomial equation of degree five, and then its derivative has degree four and it can be used to solve the optimization problem. It is important to highlight the fact that this solution allows to reduce the weight of calculations such that the planning can be also realized on real time on board implementation. In fact, it is sufficient to plug the initial and final conditions to get the solution t_f.

3.3. METP-external safety

The optimization control problem as described so far can perform quite a lot, but just if the quadrotor moves through free space. As a matter of fact, this solution does not even provide any algorithms for the obstacle avoidance if the configuration space is also occupied by obstacles. In this work, the implementation of algorithm EA-RRT (Energy-Aware Rapidly-exploring Random Tree) is proposed, which enables to avoid misses and collision during the mission. The classical enhanced form of the RRT is suggested and there is no need to consider just a single last path that it supplies. Indeed, since minimum energy consumption is important in this task so different paths are explored to decide what is the minimum energy path. None of this test can be made prior to the execution of the RRT algorithm, since this algorithm will return the 3D points the are used to generate the final trajectory, hence it is clear that it is possible to calculate the energy spent on the mission only after we know the points to form the path. Algorithm 1 explains the proposed EA-RRT algorithm.

It is important to note that the robot avoidance algorithm that is used most on the move is RRT. This is a probabilistic method that examines this configuration space in the event the robot must move, and then gives the best final path from the give start to the final position. The algorithm does not construct a roadmap of the entire free configuration space, as in PRM (Probabilistic Roadmap Method), but only a partial representation of the relevant free configuration space that can contribute to solving a given problem. With this method it is possible to generate several potential trajectories from the position of the UAV to the destination point designated for the mission. In addition, a spline method was used and added into the algorithm to produce one more smoother route. It was implemented to provide two distinct results to the trajectory planning problem, such that the shortest and most smooth one can be compared to other approach. So to make it possible to compare the energy that would be consumed travelling along both the paths and then to select the best one, as described above.

Algorithm 1 Energy-Aware RRT (EA-RRT).

Input: Start position $start$, Goal position $goal$, Step size δ
Output: Path with minimum energy consumption
Initialize:
  $T \leftarrow \{start\}$ ⊳ Initialize tree with the starting point
  $paths \leftarrow \emptyset$ ⊳ List to store all possible paths
while Not ReachedGoal($T, goal$) **do**
  $q_{\text{rand}} \leftarrow$ RandomSample(ConfigurationSpace()) ⊳ Sample a random point
  $q_{\text{near}} \leftarrow$ NearestNode(T, q_{rand}) ⊳ Find the nearest node in T
  $q_{\text{new}} \leftarrow$ NewConfiguration($q_{\text{near}}, q_{\text{rand}}, \delta$) ⊳ Generate a new point
  if CollisionFree($q_{\text{near}}, q_{\text{new}}$) **then**
    $T \leftarrow T \cup \{q_{\text{new}}\}$ ⊳ Expand the tree
    if PathToGoal($q_{\text{new}}, goal$) **then**
      $paths \leftarrow paths \cup$ GetPath($T, q_{\text{new}}, goal$) ⊳ Store the path
    end if
  end if
end while
$energyCost \leftarrow$ Map($ComputeEnergy, paths$) ⊳ Compute energy consumption for each path
$minEnergyPath \leftarrow paths$[argmin($energyCost$)] ⊳ Select the path with minimum energy
Return: $minEnergyPath$

4. Simulation and analysis

In the implementation setup, it is the detailed demonstration of the proposed work is described and how the core concept of the suggested model is developed. To recap, generating a lifecycle path for the quadrotor is aimed to traverse from a start position to a target position, avoiding physical obstacles, whilst minimizing energy consumption, as per our operational definitions in previous sections of the paper. This main goal can, as suggested before splitting into three sub-goals and each of this must have a specific implementation.

- **EA-RRT Implementation:** Firstly, the proposed EA-RRT algorithm allows obtaining a rough safe path among the obstacles, simply consisting of points connected by segments. We use the connecting points between two segments as waypoints for our problem. In fact, this phase is only required to generate a collision-free path, but it does not provide any trajectory.
- **Energy Optimization:** After generating the path, for each subsequent pair of points, the optimization problem proposed in section 2 is adopted in order to compute the best time that minimizes energy consumption during the mission.
- **Building Final Trajectory:** Finally, all the previous solutions are combined by using the generated array of execution times given for all pairs of points to compute the final minimal energy trajectory.

4.1. No obstacle scenario

The first proposed scenario is the base scenario that is available to the quadrotor. In fact, here, a free environment is considered, since the quadrotor is always beginning from some position and just need to reach an assigned final position. This is a special case, as in this special case a portion of the algorithm can be completely omitted: If a free-collision path is covered, using the RRT algorithm as an external safety portion of the overall solution does not need to be covered, as there would be no collision possible. This is removed from the part of algorithm which takes away computational weight and it can be interpreted in terms of execution time.

In Fig. 2, the optimization execution spikes are relatively smaller, which indicates that computation does not change with complexity of trajectories. In acceleration graph, both trajectory time and trajectory energy are quite aligned to each other which brings a strong relationship of acceleration and energy consumption. For the velocity, the computational overhead remains relatively constant, indicating the stability of the optimized performance regardless of the flight patterns of UAVs. Fig. 2 provides some further evidence that the optimization algorithm prioritizes energy consumption with high importance and indicates this is corroborated with trajectories in their synchronized energy and trajectory profiles.

Fig. 3 depicts the normalized thrust control by the no-obstacle design problem as a function of time, demonstrating the reduced thrust as a result of the optimization and the trade-off between the trajectory performance and energy efficiency. Thrust oscillation is normalized in the range of [0.2, 0.8] along the leg, revealing system adaptability to environmental and resource limitations, and the energy and trajectory scoring factors remain highly correlated with execution times revealing effective computation. The optimization execution time is consistent throughout, demonstrating the algorithm's ability to limit execution time even under differing thrust demands. The results indicate the effectiveness of trajectory planning for simultaneously ensuring safe and energy-efficient UAV performance.

In addition, it can be observed that, given a trajectory, the trajectory execution time and total energy used are consistent, which demonstrates that the algorithm could minimize energy usage while maintaining the required flight time and safety. Fig. 3 shows the optimal path found

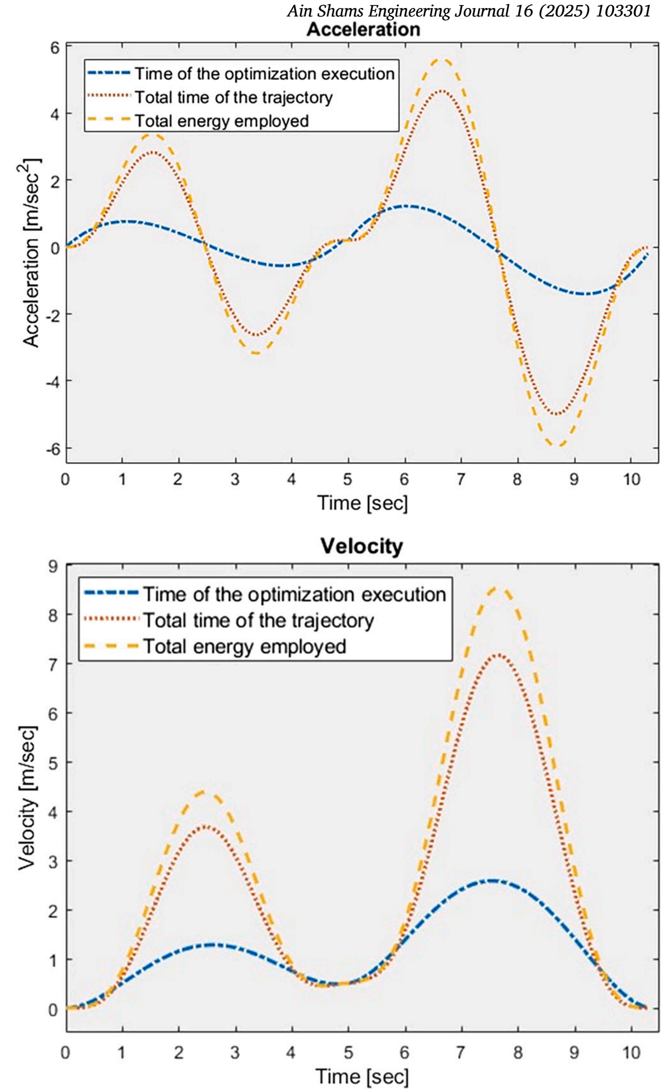


Fig. 2. No-obstacles scenario - acceleration and velocity results.

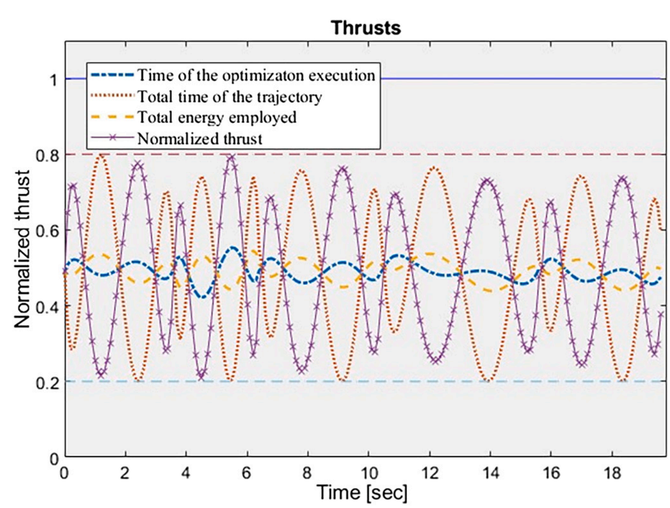


Fig. 3. No-obstacles scenario - normalized thrusts.

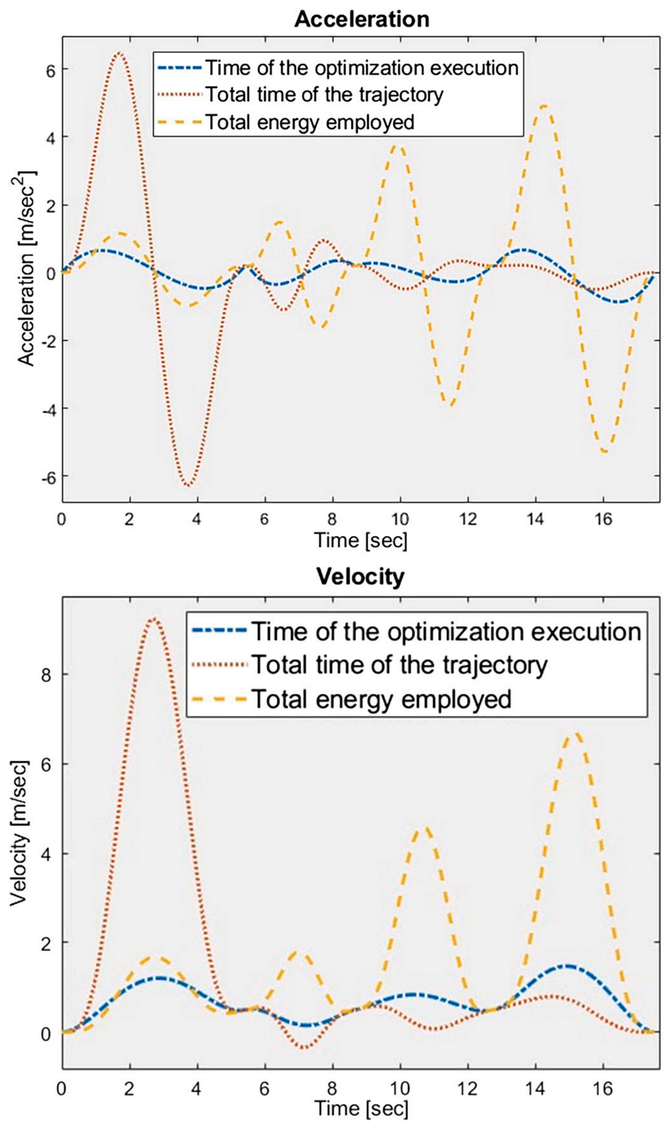


Fig. 4. One-obstacle scenario - acceleration and velocity results.

by the optimization algorithm to find a balance between the multiple objectives incorporated within our model combining energy efficiency, real-time computations feasibility, and operational safety, therefore, it is a robust solution for UAV trajectory planning on cloud environments.

4.2. One-obstacle scenario

In this section, the measurements from the second scenario, characterized by the presence of one obstacle, are reviewed based on simulation results. As there was only one trajectory planning in the no-obstacle situation, so in that empty environment it was very obvious that the trajectory is both minimum time and minimum energy trajectory. In one-obstacle case, however, one can have the RRT algorithm that will generate a smooth trajectory.

In Fig. 4, the acceleration profile shows clear variations compared to the no-obstacle case. These deviation imply the UAV in actively rolling to avoid the obstacle, means that the control forces required are large. With regard to those parameters, the time of optimization execution keeps a more stable profile and is smoother than the more fluctuating profiles of total time of the trajectory and total energy performed. These consistency highlight the optimization algorithm's attempt to maintain computational stability amidst the environmental perturbations introduced by the obstacle. The insight of total energy employed curve indicates that much more energy is expended in maneuvering modes, particularly at the avoidance stages [10-12] seconds. This means higher thrust requirements to perform rapid acceleration or turnaround. However, unlike the position plot, the velocity profile indicates significant oscillation away from steady flight, with peaks reaching up to almost 9 m/s at [2-3] seconds and again at [10-11] seconds. In this case again, we can see the execution time, which, for this example, is once again a smoother curve of execution time, showing that the optimization algorithm is doing robust trajectory planning. Real trajectory and energy profiles diverge, particularly when high velocities are reached, revealing the extra energy and control changes needed.

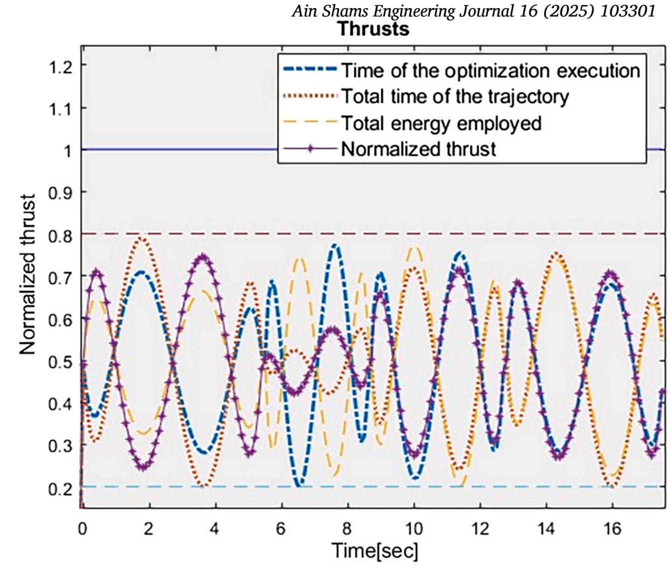


Fig. 5. One-obstacles scenario - normalized thrusts.

Fig. 5 shows the normalized thrust graph for the continuous modulation for one-obstacle case. This graph shows the tradeoff between thrust modulation, energy consumption, and trajectory time. UAV accomplishes its mission goals without compromising on these conflicting parameters. The optimization algorithm put forward in this paper is executed in discrete time, hence maintaining a trade-off between computation load and real-time responsive. Somewhat sinusoidal, "detection and avoidance" behavior implies that there are adaptive control systems (if they exist), adapted to external stimuli, to this one obstacle.

4.3. Three-obstacle scenario

Finally, the three-obstacles scenario is discussed in this section. This scenario serves as the primary test bench for demonstrating the effectiveness of the proposed approach. Fig. 6 shows the acceleration and velocity results in the three-obstacle case. The acceleration and velocity profiles show the UAV dynamics. Each metric shows oscillatory behavior as the UAV must constantly adjust its motion to avoid obstacles. Acceleration peaks occur with fast changes of velocity, especially near critical places within the trajectory they hit an obstacle. The optimization curve emerges more frequently in acceleration activity with high activity levels, emphasizing the role of optimization in long-lasting critical maneuvers. There are spikes in velocity at 5 seconds and 12 seconds. These could be when the UAV accelerates to avoid an obstacle or recovers from deceleration caused by previous maneuvers. Although these maneuvers and corrections are important for safe navigation, they demand more energy consumption, as understood from the dependency of energy usage on both velocity and acceleration magnitudes. Optimization execution time refers to the instances in which the control

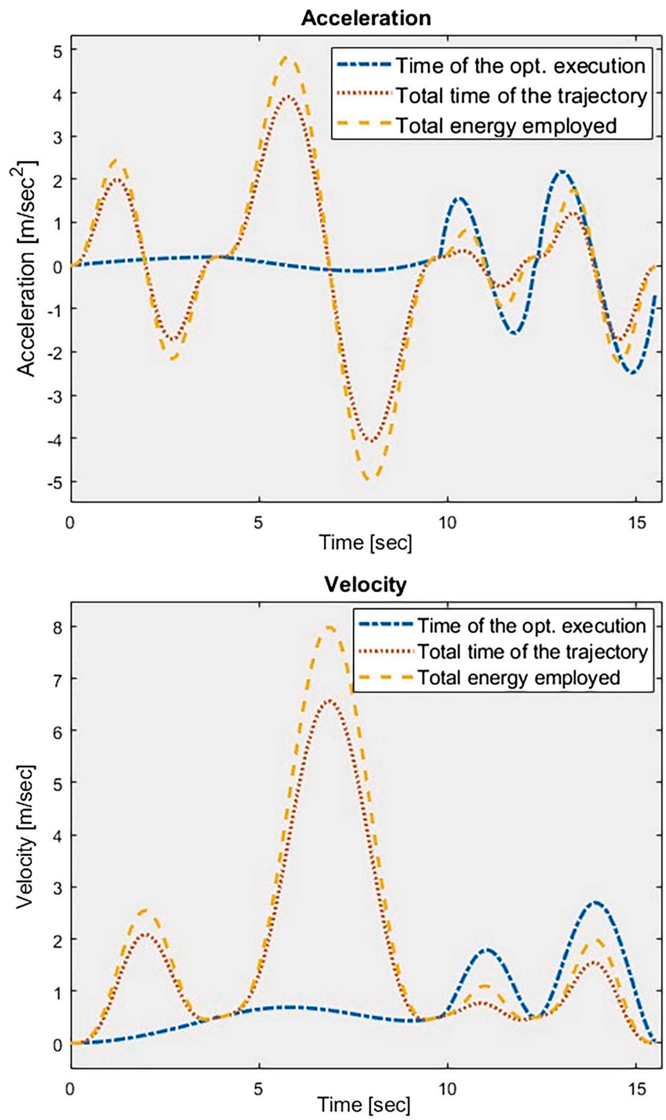


Fig. 6. Three-obstacle scenario - acceleration and velocity results.

algorithm is engaged in a recalculate operation for UAV trajectory. The phases where the motion is greatly adjusted match the specific time adjustability of the optimization strategy used. This shows a good trade-off between trajectory constraints, energy consumption and collision avoidance.

Fig. 7 shows the normalized thrust profile of a UAV flying through a 3-obstacle scenario. It details how the thrust of the UAV is dynamically modulated, achieving success in avoiding the obstacle while respecting the constraints of the mission. The thrust varies along the trajectory, reflecting the UAV's need to adapt constantly to changing environmental and obstacle conditions. The UAV demonstrates periodic thrust modulation, exhibiting alternating phases of acceleration (increased thrust) followed by deceleration (decreased thrust) as the UAV moves through and around obstacles. Numerous spikes and dips in thrust can be seen at various time intervals [5, (9-10)] seconds and near 15 seconds, indicating periods of high activity (obstacle avoidance, trajectory correction, etc.). This is particularly significant as we see optimization intervals quite frequently throughout the orbit, particularly during intervals which experience increased thrust oscillations, proving that real time capability is an essential task to reliably promote safe navigation. Additionally, the execution of the optimization, indicated by the blue dotted line, correlates with specific times of thrust activity, demonstrating the real-time adaptive nature of the UAV's control system. The interplay

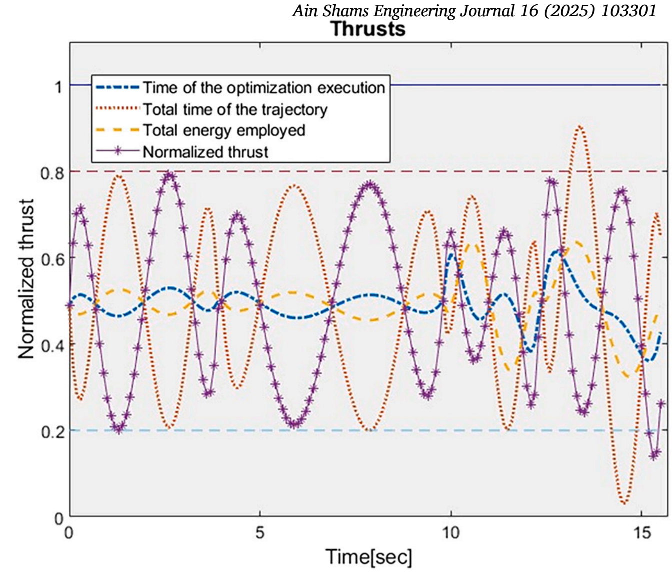


Fig. 7. Three-obstacles scenario - normalized thrusts.

between thrust and energy consumption reveals the energy-intensive nature of quick thrust changes, which calls for careful control methods.

5. Conclusion

Energy consumption rates directly influence the effective duration of UAVs' battery life and thus limit UAVs' operational range. Short UAV Flight endurance, which can be as high as 15-30 minutes, is among its most significant drawbacks. The literature offers a number of approaches to tackle this issue, with each approach prioritizing the optimization of differing parameters with respect to UAV performance. In this paper, the problem is explained the problem through another algorithmic way of motion planning taking into priority the energy use of the UAVs to elongate their life. Here, a safe and energy-efficient trajectory planning method is developed, which generates collision-free plans that move from a specified initial state toward a desired terminal state in an energy-conserving way during the hop. Central to this strategy was the formulation of an optimization problem seeking to minimize the energy expenditure over the trajectory. Herein, the internal and external safety is detailed which, once implemented was in the control system enabling to prevent actuator saturation and hence an additional safety measure to reach through constrained environment. It used an improved RRT algorithm (Rapidly-exploring Random Tree) because it checked many possible paths before it started moving straight on a path. This allowed to determine whether the minimum energy pathway coincided with that of the shortest path. To create such a solution, the theoretical foundation of the proposed work is defined as well as its most salient characteristics. Following the conceptual underpinnings, extensive simulations were performed under diverse conditions to illustrate the effectiveness and robustness of our algorithm. These simulations were a "calibration test" where we varied combinations of parameters in search of top configurations that yield the best results. In the second step, to reduce energy, fmincon function in MATLAB was called to solve the optimization problem in trajectory refinement.

The simulation studies demonstrated with many obstacles that the algorithm will learn a minimum energy path from one target location to another. In addition, UAV performance metrics were quantified and explored such as thrust, speed and acceleration. These results confirmed that control system could plan energy-efficient paths, while ensuring safety in presence of adversarial scenarios. An even more complex test case presented three varying barriers to avoid, this resulted in several path-finding costly actions to be done, but ultimately was successful in the fact that the platform was able to determine a position that would not lead to a collision for all of its mission criteria while ensure the path taken would stay within the actuator energy draw parameters. While the results were promising and demonstrated the effectiveness of the approach, the areas for future work were also identified. Apart from the mentioned fact, it was discovered that the computation time of the system is actually not that small, whereby the algorithm may not be run in real time. This could help to enhance the speed in calculations, for example, for iterations by considering to either try other algorithms instead of fmincon function or tune the parameters of fmincon function itself to make it work faster. This optimization would suit online applications very well and where performance needs to be derived from real-time. In this work, it is aimed to create a control system that can be used to solve a multitude of real world problems related to UAV trajectory planning and we will test our control system under multiple conditions. As a first step towards this goal, by analyzing trajectory optimization tools in a safe and energy-efficient manner, stronger foundations for future works considering faster but more complex algorithms are now at hand. Subsequent iterations of this work will seek to optimize the algorithm in a manner that accounts for real-time processing constraints with significance placed on safety, wherein high energy conservation is imperative, to allow for offline scenarios and real-time UAV incorporation.

CRediT authorship contribution statement

Mushtaq Muhammad Umer: Writing – original draft, Methodology, Investigation, Formal analysis, Conceptualization. **Hein Venter:** Supervision, Formal analysis. **Owais Muhammad:** Software, Methodology. **Tamoor Shafique:** Writing – review & editing. **Fuad A. Awwad:** Investigation, Funding acquisition. **Emad A.A. Ismail:** Validation, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known financial or personal conflicts of interest that could have influenced the work reported in this manuscript.

Acknowledgements

Researchers Supporting Project number (RSPD2025R1060), King Saud University, Riyadh, Saudi Arabia.

References

[1] Jordan S, Moore J, Hovet S, Box J, Perry J, Kirsche K, et al. State-of-the-art technologies for uav inspections. IET Radar Sonar Navig 2018;12(2):151–64.
[2] Hussain A, Li S, Hussain T, Lin X, Ali F, AlZubi AA. Computing challenges of uav networks: a comprehensive survey. Comput Mater Continua 2024;81(2).
[3] Gupta A, Afrin T, Scully E, Yodo N. Advances of uavs toward future transportation: the state-of-the-art, challenges, and opportunities. Future Transp 2021;1(2):326–50.
[4] Santoso F, Garratt MA, Anavatti SG. State-of-the-art intelligent flight control systems in unmanned aerial vehicles. IEEE Trans Autom Sci Eng 2017;15(2):613–27.
[5] Bijjahalli S, Sabatini R, Gardi A. Advances in intelligent and autonomous navigation systems for small uas. Prog Aerosp Sci 2020;115:100617.
[6] Yeong DJ, Velasco-Hernandez G, Barry J, Walsh J. Sensor and sensor fusion technology in autonomous vehicles: a review. Sensors 2021;21(6):2140.
[7] Xu W, Yan C, Jia W, Ji X, Liu J. Analyzing and enhancing the security of ultrasonic sensors for autonomous vehicles. IEEE Internet Things J 2018;5(6):5015–29.
[8] Parekh D, Poddar N, Rajpurkar A, Chahal M, Kumar N, Joshi GP, et al. A review on autonomous vehicles: progress, methods and challenges. Electronics 2022;11(14):2162.
[9] Elmeseiry N, Alshaer N, Ismail T. A detailed survey and future directions of unmanned aerial vehicles (uavs) with potential applications. Aerospace 2021;8(12):363.
[10] Zuo Z, Liu C, Han Q-L, Song J. Unmanned aerial vehicles: control methods and future challenges. IEEE/CAA J Autom Sin 2022;9(4):601–14.
[11] Bayomi N, Fernandez JE. Eyes in the sky: drones applications in the built environment under climate change challenges. Drones 2023;7(10):637.
[12] Jin H, Jin X, Zhou Y, Guo P, Ren J, Yao J, et al. A survey of energy efficient methods for uav communication. Veh Commun 2023;41:100594.
[13] Suanpang P, Jamjuntr P. Optimizing autonomous uav navigation with d* algorithm for sustainable development. Sustainability 2024;16(17):7867.
[14] Arafat MY, Alam MM, Moh S. Vision-based navigation techniques for unmanned aerial vehicles: review and challenges. Drones 2023;7(2):89.
[15] Fang Z, Savkin AV. Strategies for optimized uav surveillance in various tasks and scenarios: a review. Drones 2024;8(5):193.
[16] Aabid A, Parveez B, Raheman MA, Ibrahim YE, Anjum A, Hrairi M, et al. A review of piezoelectric material-based structural control and health monitoring techniques for engineering structures: challenges and opportunities. In: Actuators, vol. 10. MDPI; 2021. p. 101.
[17] Boukoberine MN, Zhou Z, Benbouzid M. A critical review on unmanned aerial vehicles power supply and energy management: solutions, strategies, and prospects. Appl Energy 2019;255:113823.
[18] Dörfler S, Walus S, Locke J, Fotouhi A, Auger DJ, Shateri N, et al. Recent progress and emerging application areas for lithium–sulfur battery technology. Energy Technol 2021;9(1):2000694.
[19] Li J-C, Xu H, Zhou K, Li J-Q. A review on the research progress and application of compressed hydrogen in the marine hydrogen fuel cell power system. Heliyon 2024.
[20] Mohsan SAH, Khan MA, Noor F, Ullah I, Alsharif MH. Towards the unmanned aerial vehicles (uavs): a comprehensive review. Drones 2022;6(6):147.
[21] Zhang Q, Jiang H, Guo Q, Tian J, Shi F, Umer MM, et al. Uav-assisted wireless communication network capacity analysis and deployment decision. In: International conference in communications, signal processing, and systems. Springer; 2020. p. 1185–95.
[22] Huang M, Wan N, Zhu H. Reconstruction of structural acceleration response based on cnn-bigru with squeeze-and-excitation under environmental temperature effects. J Civ Struct Health Monit 2024:1–19.
[23] Huang M, Zhang J, Li J, Deng Z, Luo J. Damage identification of steel bridge based on data augmentation and adaptive optimization neural network. Struct Health Monit 2024:14759217241255042.
[24] Zhang J, Huang M, Wan N, Deng Z, He Z, Luo J. Missing measurement data recovery methods in structural health monitoring: the state, challenges and case study. Measurement 2024:114528.
[25] Huang M, Zhang J, Hu J, Ye Z, Deng Z, Wan N. Nonlinear modeling of temperature-induced bearing displacement of long-span single-pier rigid frame bridge based on dcnn-lstm. Case Stud Therm Eng 2024;53:103897.
[26] Huang M-S, Gül M, Zhu H-P. Vibration-based structural damage identification under varying temperature effects. J Aerosp Eng 2018;31(3):04018014.
[27] Huang M, Ling Z, Sun C, Lei Y, Xiang C, Wan Z, et al. Two-stage damage identification for bridge bearings based on sailfish optimization and element relative modal strain energy. Struct Eng Mech 2023;86(6):715–30.
[28] Apostolidis SD, Vougiatzis G, Kapoutsis AC, Chatzichristofis SA, Kosmatopoulos EB. Systematically improving the efficiency of grid-based coverage path planning methodologies in real-world uavsí operations. Drones 2023;7(6):399.
[29] Sun D, Sun Z, Shao P. Accelerated path planning for large-scale grid maps. IEEE Access 2024.
[30] Véras LGD, Medeiros FL, Guimaráes LN. Systematic literature review of sampling process in rapidly-exploring random trees. IEEE Access 2019;7:50933–53.
[31] Xu T. Recent advances in rapidly-exploring random tree: a review. Heliyon 2024.
[32] Xu Z, Deng D, Shimada K. Autonomous uav exploration of dynamic environments via incremental sampling and probabilistic roadmap. IEEE Robot Autom Lett 2021;6(2):2729–36.
[33] Ait Saadi A, Soukane A, Meraihi Y, Benmessaoud Gabis A, Mirjalili S, Ramdane-Cherif A. Uav path planning using optimization approaches: a survey. Arch Comput Methods Eng 2022;29(6):4233–84.
[34] Puente-Castro A, Rivero D, Pazos A, Fernandez-Blanco E. A review of artificial intelligence applied to path planning in uav swarms. Neural Comput Appl 2022;34(1):153–70.
[35] Wu T, Zhang Z, Jing F, Gao M. A dynamic path planning method for uavs based on improved informed-rrt* fused dynamic windows. Drones 2024;8(10):539.
[36] Diaz-Vilor C, Lozano A, Jafarkhani H. Cell-free uav networks: asymptotic analysis and deployment optimization. IEEE Trans Wirel Commun 2022;22(5):3055–70.
[37] Chai R, Guo Y, Zuo Z, Chen K, Shin H-S, Tsourdos A. Cooperative motion planning and control for aerial-ground autonomous systems: methods and applications. Prog Aerosp Sci 2024;146:101005.
[38] Thakur A, Mishra SK. An in-depth evaluation of deep learning-enabled adaptive approaches for detecting obstacles using sensor-fused data in autonomous vehicles. Eng Appl Artif Intell 2024;133:108550.
[39] Maboudi M, Homaei M, Song S, Malihi S, Saadatseresht M, Gerke M. A review on viewpoints and path planning for uav-based 3-d reconstruction. IEEE J Sel Top Appl Earth Obs Remote Sens 2023;16:5026–48.

[40] Chen H, Lu P. Real-time identification and avoidance of simultaneous static and dynamic obstacles on point cloud for uavs navigation. Robot Auton Syst 2022;154:104124.

[41] Liu D, Zong Q, Zhang X, Zhang R, Dou L, Tian B. Game of drones: intelligent online decision making of multi-uav confrontation. IEEE Trans Emerg Top Comput Intell 2024.

[42] Chang Y, Cheng Y, Manzoor U, Murray J. A review of uav autonomous navigation in gps-denied environments. Robot Auton Syst 2023:104533.

[43] Gupta A, Fernando X. Simultaneous localization and mapping (slam) and data fusion in unmanned aerial vehicles: recent advances and challenges. Drones 2022;6(4):85.

[44] Yang T, Li P, Zhang H, Li J, Li Z. Monocular vision slam-based uav autonomous landing in emergencies and unknown environments. Electronics 2018;7(5):73.

[45] Xu X, Zhang L, Yang J, Cao C, Wang W, Ran Y, et al. A review of multi-sensor fusion slam systems based on 3d lidar. Remote Sens 2022;14(12):2835.

Dr. Mushtaq Muhammad Umer is a postdoctoral fellow at University of Pretoria, South Africa specializing in cognitive radio technologies, dynamic spectrum management, and optimization strategies for unmanned aerial vehicles (UAVs). Dr. Mushtaq holds a Bachelor's and Master's degree in Software Engineering form Pakistan and earned his Ph.D. in Control Science and Engineering from Southwest University of Science and Technology, China, in June 2024. His research focuses on developing innovative solutions for energy-efficient and safe UAV operations in real-world scenarios, integrating advanced algorithms and adaptive techniques. He has authored several scholarly articles and actively contributes to advancing knowledge in wireless communication, autonomous systems, and network optimization.