Scholars Journal of Engineering and Technology
OPEN ACCESS
Abbreviated Key Title: Sch J Eng Tech
ISSN 2347-9523 (Print) | ISSN 2321-435X (Online)
Journal homepage: https://saspublishers.com

Quadrotor Trajectories Optimisation using Metaheuristic Algorithm in Python

Oumar Abderamane Mahamat[1*], Dr Mahamat Issa Hassan[2]

[1]Doctoral School of Environmental and Technical Sciences, University of N'Djamena, N'Djamena Chad
[2]Associate Professor, Faculty of Applied and Pure Science, University of N'Djamena, N'Djamena Chad

DOI: https://doi.org/10.36347/sjet.2024.v12i11.002 | **Received:** 28.09.2024 | **Accepted:** 04.11.2024 | **Published:** 16.11.2024

***Corresponding author:** Oumar Abderamane Mahamat
Doctoral School of Environmental and Technical Sciences, University of N'Djamena, N'Djamena Chad

Abstract **Review Article**

Quadrotors, known for their agility and ability to perform intricate manoeuvres, have become increasingly popular in the last decade. Precisely tracking complex flight paths has been a major research focus. While trajectory optimization offers a continuous search space and allows for utilizing problem-specific knowledge, its reliance on local optimization methods necessitates a global planner for generating feasible trajectories from arbitrary starting points to goals. Designing such trajectories for quadrotors, with their five degrees of freedom and intricate dynamic constraints (e.g., limitations on state variables), presents a significant challenge. Existing global planners struggle to solve these trajectory generation problems for complex dynamical systems in a practical timeframe. This paper explores the use of metaheuristic optimization for planning quadrotor flight trajectories, specifically focusing on the FSA and FSASCA algorithms. These methods have demonstrated promising results in specific scenarios, but their performance depends heavily on the application and its constraints. The paper emphasizes a trade-off between solution quality and execution speed: seeking faster execution may require sacrificing solution quality, while prioritizing optimality may lead to longer execution times.

Keyword: Quadrotor, Trajectories, Metaheuristic, Optimization.

1. INTRODUCTION

Quadrotors, rapidly gaining traction as versatile aerial platforms and transportation vehicles, hold the promise of revolutionizing various fields, such as structural inspection, package delivery, and search and rescue. Their ability to transform cost-intensive and time-consuming tasks into dynamic, efficient, and reliable solutions is poised to surpass traditional methods. However, realizing this potential requires addressing the challenges associated with designing quadrotors capable of executing these tasks efficiently and concurrently (Natarajan *et al.,* 2021). One significant challenge lies in enabling quadrotors to dynamically plan and execute trajectories in complex environments while adhering to safety constraints and operating with limited environmental knowledge. The motion planning problem for quadrotors involves finding a feasible trajectory that respects both the quadrotor's physical limitations (movement and control constraints) and guarantees collision-free flight between defined starting and goal locations. Trajectory design for quadrotors demands a continuous evolution of relevant states over a finite time interval, a requirement imposed by control algorithms. While triangulated polygonal trajectories (consisting of straight line segments connected by "teleportation" points) offer a simple and efficient approach for multi-rotor vehicles in obstacle-free spaces, they are insufficient for quadrotors due to their dynamic and control constraints. Recent research has focused on utilizing a well-established mathematical framework to describe feasible quadrotor motion in a compact form as a reduced-order 3D polynomial trajectory formulation. This formulation, restricted by kinematic limits, provides a promising approach for solving the quadrotor motion planning problem (Foehn *et al.,* 2021).

2. BACKGROUND AND LITERATURE REVIEW

2.1 Fundamentals of Quadrotor Trajectories

Quadrotor systems, characterized by their use of four rotors to generate lift, have emerged as versatile flying devices capable of navigating, mapping, collecting data, and exploring environments on land, in the air, and underwater. Recent technological advancements have fuelled the development of autonomous quadrotor systems, paving the way for explorations of unknown environments represented by sparsely sampled continuous occupancy grids. Applications like search and rescue, geological sampling, and environmental monitoring are poised to benefit from these developments (Mokrane *et al.,* 2022). The miniaturization of sensing and processing technologies, such as IMUs, cameras, barometers, lidars, GPS, and others integrated into embedded chips, combined with advancements in flying vehicles (including airplanes, helicopters, and multirotor), has made the realization of autonomous flight increasingly feasible. However, bringing such automation to fruition in real-world settings remains challenging due to the inherent uncertainties and dynamic nature of environments. Quadrotors, with their distinctive upside-down flight capability and superior agility, offer unique advantages for various applications, including package delivery, pesticide spraying, and firework displays. However, their high manoeuvrability also introduces complexity to the planning process, necessitating sophisticated approaches to ensure safe and efficient operation (Natarajan *et al.,* 2021).

2.2 Quadrotor Dynamics and Trajectory Optimization

Quadrotors, a type of rotorcraft, utilize four rotors to generate both lift and propulsion. These rotors are symmetrically positioned at the four corners of the rigid body, providing vertical movement through thrust and controlling the vehicle's tilt using the generated torque (Hehn *et al.*, 2012). The increasing popularity of quadrotors as aerial vehicles has propelled trajectory optimization into a crucial area of research. Optimizing trajectories allows for enhanced operational capabilities, enabling quadrotors to execute complex manoeuvres with increased precision and efficiency.

2.3 Basic Dynamics of Quadrotors

A quadrotor is an aerial vehicle that is designed to lift and navigate along a three-dimensional space. Quadrotors have been emerged rapidly in autonomous aerial vehicle community. Together with their applications, they have been extensively studied recently. Trajectory optimization in quadrotors plays a crucial role in improving the performance and efficiency of these aerial vehicles. Quadrotors, or quadcopters, have gained immense popularity in various applications such as aerial photography, surveillance, and environmental monitoring due to their ability to hover, take off, and land vertically. However, controlling quadrotors can be challenging due to their inherent nonlinearity and strong coupling. Therefore, research efforts have focused on the design and testing of quadrotor control algorithms to achieve desired performance. One important aspect of control design is the trajectory that quadrotors follow to achieve the desired state in a defined time interval. Quadrotor control algorithms rely on mathematical models of the considered vehicle and apply either open-loop (feedforward) or closed-loop (feedback) control. The trajectory can be freely defined by the user, but it is essential to minimize performance criteria such as task completion time, fuel consumption, and energy expenditure (Hehn *et al.,* 2012).

2.4 Metaheuristic Optimization

An exciting and important area of research is motion planning for quadrotors that tracks and executes complex, often aggressive, manoeuvres using images from a single viewpoint. Quadrotors exhibit exceptional agility in performing aerial acrobatics such as flips, dives, and fast spirals, that create captivating views. Quadrotors' exposure to obstacles is frequently similar to the manoeuvres they perform. A prior understanding of the route may be expressed as waypoints and timing windows that actively relies on the model. Furthermore, environments with dense obstacles may create trajectories where it is challenging to remain within the obstacles-free corridor. Consider an initial trajectory that is not smooth. Using other methods, it might be smoothed, but such approaches generate more aggressive trajectories. A paradigm is presented that combines a graph search over a piecewise linear graph with trajectory optimization. The graph discretizes the search space and is naturally robust to a flow disturbance. The continuous dynamics is handled using a predefined control vector instead of precomputed control inputs. This is a powerful assumption since it converts a much more complex kinodynamic planning problem into simple geometric queries on the graph structure (Odili, 2018).

2.5 Common Metaheuristic Algorithms

A discussion of metaheuristic algorithms would be incomplete without an insight into some specific algorithms concerned with metaheuristic optimization tasks. The following algorithms are common in the discussion of metaheuristic optimization performance: The African Buffalo Optimization Algorithm, the Firefly Algorithm, and the Particle Swarm Optimization Algorithm. The African Buffalo Optimization Algorithm is one of the latest metaheuristic optimization algorithms. This algorithm is based on the collective movement of African buffalo (Nigeria and Ghana buffaloes). Foraging is a type of social activity performed by buffalo that helps to improve the overall foraging efficiency of the entire group (Odili, 2018).

The Firefly Algorithm, proposed by Yang in 2008; is a new style of metaheuristic optimization algorithm based on the behaviour of firefly attractions. The Firefly Algorithm is inspired and developed based on the nature of bioluminescence (natural glow) and mating behaviour among fireflies. The Particle Swarm Optimization Algorithm, developed by Kennedy and Eberhart in 1995, is a population-based optimization technique inspired by the social behaviour of bird flocking or fish schooling. This optimization algorithm has been used in various applications and is favoured as a metaheuristic optimization algorithm. The Particle Swarm Optimization Algorithm is composed of a set of individuals (particles), which are potential solutions that move in the solution space. Each particle adjusts its position based on its own experience and that of its neighbours. The adjustment is based on the velocity equation, which is a linear combination of three basic components: inertia, cognitive, and social (Natarajan *et al.*, 2021).

2.6 Integration of Metaheuristic Optimization in Quadrotor Trajectories

Over the past years, regarding to using of quadrotors in various applications, ranging from military applications to civilian use, the control and trajectory optimization for these flying robots have become a matter of considerable interest. The inability of quadrotors to exactly model their dynamics and uncertainties in environment make the control and path planning problems even harder. Researches on applications of different metaheuristic optimization techniques in trajectory optimization of quadrotors have yielded promising results. By metaheuristic optimization techniques, a cost function is minimized with respect to parameters of a polynomial trajectory resulting in a smooth reference trajectory for quadrotor, and therefore, the guidance, navigation and control of quadrotor become easier problems to resolve (Safaee & K. Kamaleddin Mousavi Mashhadi, 2017).

Cost function includes terms that imply trajectory adherence, smoothness, non-oscillation, maximum velocities and accelerations etc. Several metaheuristic optimization techniques including Genetic Algorithm, Particle Swarm Optimization have been applied for quadrotor trajectory optimization. Both of these techniques have been applied to quadrotor trajectory optimization based upon use of polynomials of 4th and 6th degree (Natarajan *et al.*, 2021).

3. The Proposed System

Future Search Algorithms are a family of nature-inspired optimizers derived from observed social behaviour of biological species in nature. The original FSA was proposed as a global optimizer, modelling individual social behaviour as a collaborative search activity. As an improvement, the concept of elite individuals was added, which can promote the collaborative search among the better individuals (Tang & Wang, 2023). The quadrotor’s state (position and speed) is modelled as a function of time, and the vehicle’s model is taken into consideration. The trajectory optimization scheme mainly consists of three steps: sampling, geometric planning, and trajectory smoothing. The SCA was developed in 2016 with three successful levels. the central goal of this algorithm is to develop an optimization framework based on the Sine-Cosine Algorithm (SCA) and Future Search Algorithm (FSA), targeting the efficient resolution of trajectory optimization problems for quadrotors specifically in path planning. This paper presents an approach that enables quadcopters to navigate in complex environments by taking into account dynamic constraints and obstacles present. The simulation plane was built on python to represent the crucial aspect of trajectory planning as well as to determine the constraints of the quadcopter.

These constraints are defined by:

- **Quadcopter Dynamics:** Its physical characteristics such as mass, inertia and propulsive power determine its flight capabilities and limit its ability to accelerate, decelerate and change direction.
- **Speed:** The quadcopter's maximum and minimum speed, as well as its maximum acceleration, are important constraints to ensure safe and realistic flight.
- **Number of Obstacles:** The environment is analysed to identify and quantify the number of obstacles. This information is crucial for trajectory planning, which aims to avoid them while optimizing the path taken.

Once the constraints are established, the trajectory planning is optimized using meta-heuristic algorithms. Trajectory optimization aims to maximize an objective function that incorporates key criteria such as the quadcopter must reach its destination in the shortest possible time. The implemented optimization approach is designed to provide an efficient objective function that allows to quantify the performance of the generated trajectory. Furthermore, the approach aims to ensure a high convergence rate, allowing the algorithms to quickly reach an optimal or close to optimal solution.

The following steps presents the overall proposed trajectory optimization process, which uses the metaheuristic algorithms.

1. **Definition of Constraints:** The dynamic constraints of the quadcopter, such as its maximum speed, acceleration and payload limits, are defined.
2. **Generation of Initial Trajectories:** A set of initial trajectories is generated from the single trajectory planning method.
3. **Trajectory Optimization:** The algorithms are used to optimize the initial trajectories according to the defined objective function.
4. **Convergence Evaluation:** Convergence analysis is performed to evaluate the effectiveness of the optimization and the ability of the algorithms to find an optimal solution.

The trajectory optimization approach presented in this paper is distinguished by its unique trajectory planning and in-depth convergence analysis, this method allows generating high-performance, efficient and convergent quadcopter trajectories, thus opening the way to new applications in various fields.

4. COMPARISON AND CONCLUSION

A comparison of the two algorithms, FSA and FSASCA, based on their execution times and convergence behaviours. the objective function values of both algorithms are plotted over the iterations. This graph shows how each algorithm progresses towards finding the optimal path. FSASCA requires more computation than FSA, as indicated by the longer execution time. This additional computation time can be attributed to the self-adaptive mechanisms of FSASCA which likely involve more complex computations. FSASCA converges faster to a better solution, as indicated by the faster and smaller objective function values. This suggests that it is more effective at finding optimal paths while avoiding obstacles, despite the longer execution time.

Scenario 1:

We set the obstacle number as 5 and speed to 0.2 while the obstacle radius to 1.0. for the scenario one the execution time 1.5461 seconds and 1.2278 seconds. The result of the objective function on iteration is shown in the figure below:

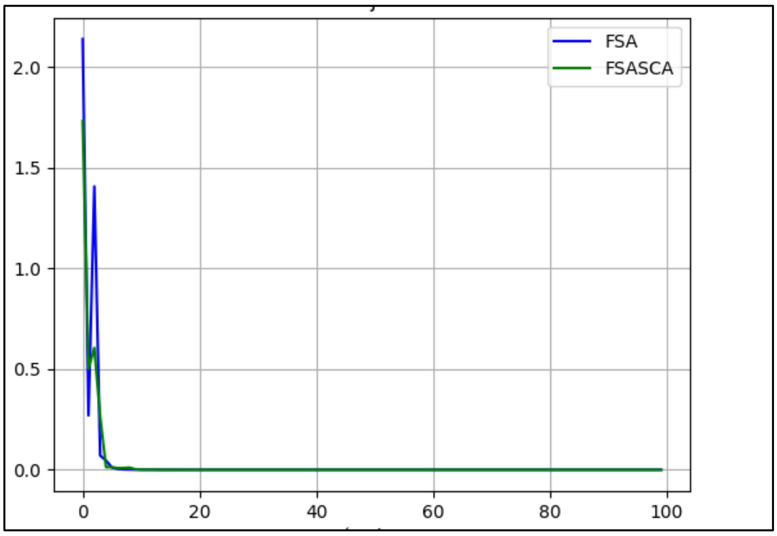

Figure 1: Simulation of scenario 1

Scenario 2:

For Scenario 2, with 50 obstacles, a speed of 0.2, and an obstacle radius of 1.0, the execution time for FSA was 9.4988 seconds, while FSASCA completed in 7.7264 seconds. The figure below illustrates the convergence of the objective function across iterations for both algorithms.

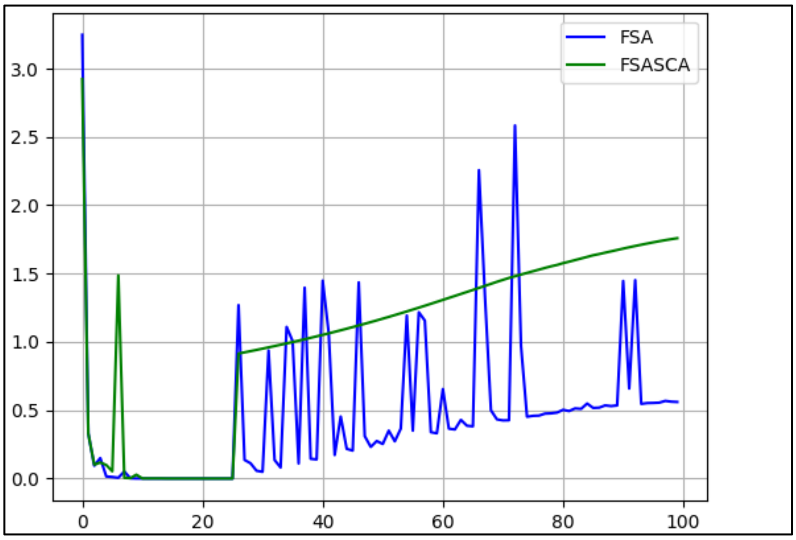

Figure 2: Simulation of scenario 2

While FSASCA may require more execution time than FSA, it consistently delivers superior results by converging more efficiently to the optimal solution. The selection between these two algorithms ultimately depends on the specific application's requirements. If rapid execution is paramount, FSA might be the preferred choice. However, if attaining the most optimal path is the primary objective, then FSASCA's greater accuracy might be a better fit, even with the extended processing time.

5. CONCLUSION AND PERSPECTIVES

The results of our investigation reveal a compelling trade-off between execution speed and solution quality. While FSA offers rapid execution, FSASCA, despite being slower, consistently demonstrates more efficient convergence toward the optimal solution. Ultimately, the choice between these algorithms hinges on the specific application and its constraints. If swift execution is paramount, some compromise in solution quality may be necessary. Conversely, prioritizing an optimal solution may necessitate a longer execution time.

Future research could focus on developing hybrid algorithms that leverage the strengths of both approaches. Such a hybrid algorithm could initiate with a quick run of FSA to generate an approximate solution, followed by refinement using FSASCA to achieve a more precise optimal solution. Additionally, optimizing both algorithms individually remains an avenue for further exploration.

REFERENCES

- Hehn, M., Ritz, R., & D'Andrea, R. (2012). Performance benchmarking of quadrotor systems using time-optimal control. *Autonomous Robots*, *33*. 10.1007/s10514-012-9282-3.
- Mokrane, A., Benallegue, A., Choukchou-Braham, A., El Hadri, A., & Cherki, B. (2022). Guidance, navigation and control for autonomous quadrotor flight in an agricultural field: The case of vineyards. *Sensors*, *22*, 22. 10.3390/s22228865.
- Natarajan, R., Choset, H., & Likhachev, M. (2021). Interleaving Graph Search and Trajectory Optimization for Aggressive Quadrotor Flight. *IEEE Robotics and Automation Letters*. PP. 1-1. 10.1109/LRA.2021.3067298.
- Odili, J. (2018). The Dawn of Metaheuristic Algorithms. *International Journal of Software Engineering and Computer Systems*, *4*, 49-61. 10.15282/ijsecs.4.2.2018.4.0048.
- Safaee, B., & Kamaleddin Mousavi Mashhadi, S. K. (2017). Optimization of fuzzy membership functions via PSO and GA with application to quad rotor. *Journal of AI and Data Mining*, *5*(1), 1-10.
- Tang, J., & Wang, L. (2023). Sine Cosine Algorithm for Elite Individual Collaborative Search and Its Application in Mechanical Optimization Designs. *Biomimetics*, *8*, 576. 10.3390/biomimetics8080576.
- Foehn, P., Romero, A., & Scaramuzza, D. (2021). Time-optimal planning for quadrotor waypoint flight. *Science robotics*, *6*. 10.1126/scirobotics.abh1221.