Journal of Intelligent & Robotic Systems (2025) 111:16
https://doi.org/10.1007/s10846-024-02214-z

REGULAR PAPER

Planning Aggressive Drone Manoeuvres: A Geometric Backwards Integration Approach

João Pinto[1] · Bruno J. Guerreiro[1,2] · Rita Cunha[1]

/ Accepted: 14 December 2024 / Published online: 13 January 2025

Abstract
This paper addresses the problem of performing aggressive manoeuvres by using multirotor vehicles that include passing through any specific point within the full state space of the vehicle. To this end, the design of optimal trajectories considers the dynamical model of the vehicles by numerically integrating it backwards in time, in the manifold where the dynamics evolve, and dividing the manoeuvres into three distinct phases to accommodate any combination of initial, desired, and final states. In the first phase, the vehicles fly from an initial to a launch configuration to achieve the necessary momenta to reach the desired one in the second phase. To ensure the feasibility of executing the second phase, the relation between snap and body torques is exploited by commanding the vehicles to track geodesic curves on SO(3) during the backwards integration. The vehicles are then driven to a final configuration in the third phase. Most existing solutions to execute aggressive and precise manoeuvres with these rotorcraft focus either on the attitude control problem, leaving the position in open-loop, or use different controllers for different sections of the manoeuvre. In this work, a single tracking controller is considered to validate the proposed trajectory planning strategy in a realistic simulation environment, which involves the PX4 firmware, and in a controlled experimental setup. The results demonstrate that accurate tracking of the designed trajectories enables the vehicles to perform 360-degree loops at great speed and manoeuvres that facilitate the exchange of a parcel between two multirotor vehicles during flight.

Keywords Trajectory planning · Trajectory tracking · Geometric numerical integration · Aggressive manoeuvres · Multirotors

1 Introduction

Autonomous flight of multirotors has been a subject of significant research – for the mechanical simplicity, compact size, high manoeuvrability, and the ability of these vehicles to carry a diverse set of sensors and a considerable payload. Given the features of these vehicles, relevant applications, namely disaster assessment and assistance, tracking of fast moving targets, and efficient package delivery, benefit from their use as long as agile motion and high-speed navigation are available. To address the scientific and technological challenges common to these applications, solid estimation, control, and planning strategies are required. The focus of this paper is on designing dynamically feasible and aggressive trajectories – which may involve fast changes in linear acceleration and its time derivatives – such that the vehicles reach a desired state (position, orientation, linear and angular velocity) in full state space, and devising controllers that enable the accurate tracking of those trajectories.

State-of-the-art methods for optimal trajectory planning leverage the differential flatness property [1] of the multirotor dynamics, which allows to formulate convex optimisation problems that include constraints on position, velocity, acceleration, and higher order time derivatives [2–6]. These methods have been successfully employed to generate polynomial trajectories that enable the vehicle to fly through narrow gaps and perch on inclined surfaces, under the assumption that an integrator model describes the multirotor dynamics. Even though the simplicity of these strategies is

---

✉ João Pinto
jpinto@isr.tecnico.ulisboa.pt

Bruno J. Guerreiro
bj.guerreiro@fct.unl.pt

Rita Cunha
rita@isr.tecnico.ulisboa.pt

[1] Institute for Systems and Robotics, LARSyS, Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal

[2] NOVA School of Science and Technology and CTS-Uninova, LASI, Universidade Nova de Lisboa, Caparica, Portugal

what makes them most attractive, to perform large excursions from the hover condition with high accuracy, a more expressive model of the vehicle dynamics should be considered. Approaches rooted in trajectory optimisation on Riemannian manifolds to plan feasible trajectories on both translation and rotation have been proposed [7], however, they require solving non-convex problems.

Recent years have also witnessed active research on planning time-optimal trajectories so that multirotor vehicles are able to fly seamlessly through multiple waypoints in minimum time. In [8], the authors depart from smooth, polynomial trajectories and present a solution to exploit the full actuator potential. Feasible, time-optimal trajectories are designed in [9] using a machine learning approach to refine the execution speed of pre-defined polynomial trajectories. These planning strategies, though, in addition to being computationally demanding, were not devised with the main goal of enabling multirotors to execute precise manoeuvres in full state space. To enable these vehicles to fly through narrow gaps under unknown disturbances, an efficient simultaneous planning and execution strategy has been proposed in [10]. Time-optimal piece-wise cubic functions serve as reference trajectories to be tracked, acceleration references are non-differentiable nonetheless. Discontinuities in the first time derivative of acceleration require the vehicle to be able to instantly change its angular velocity, which may not be suitable for applications in which high accuracy is necessary.

Other approaches rely on search-based algorithms to plan trajectories, often to circumvent obstacles in the environment by using motion primitives [11, 12]. These primitives are also used to search over the state space in [13, 14], however, search-based methods are usually computationally expensive and generally do not consider the rotational dynamics of the vehicles to perform precise attitude manoeuvres. Sampling-based kinodynamic methods [15] seek to identify feasible motions in the continuous solution space by drawing discrete samples and building connections between them, and they have proved to be effective at planning optimal and feasible paths for multirotors in 3D space [16, 17]. A kinodynamic sampling-based method that uses minimum-time motion primitives to guide the exploration of the solution space has been presented in [18] and it enables multirotor vehicles to fly at very high speeds. This class of methods, nevertheless, is not suitable for planning manoeuvres in full state space, as exploring the space of feasible solutions is impractical for dynamical systems such as a multirotor aerial vehicle. Model predictive control (MPC) schemes allow one to plan simultaneously state space and control trajectories for multirotor vehicles and enable them to perform aggressive manoeuvres [19, 20]. However, designing stabilising MPC controllers with a large basin of attraction is not straightforward [21, 22].

To perform autonomously manoeuvres in full state space with these flying platforms, various strategies propose different controllers for each section of the manoeuvre [23, 24], due to the cascaded dynamics and underactuated nature of the vehicles. Most often the rotational control loop stabilises the attitude dynamics, whereas position and linear velocity control are disregarded. Given the objective of driving multirotors to a well-defined state, both control loops should be considered throughout the manoeuvres.

In this paper, a strategy to plan demanding manoeuvres in full state space is presented, resorting to optimisation-based methods and exploiting the differential flatness of the multirotor dynamics. The manoeuvres are divided into three phases to enable vehicles to pass through initial, desired, and final configurations in state space with accuracy and precision. To plan the second phase, the vehicle dynamics are numerically integrated backwards in time, starting from the desired configuration, to find a *launch* configuration, which enables multirotors to achieve the momenta required to reach the desired one. By letting the attitude subsystem of the vehicle trace out geodesic curves on the 3D rotation matrices manifold $\mathrm{SO}(3)$ during the backwards integration, a *launch* configuration encoding the information necessary to plan minimum snap trajectories [2] for the second phase that require small body torques to be tracked by multirotor vehicles is determined. In comparison with other approaches that also propose designing trajectories by first integrating the vehicle dynamics backwards in time [23], the findings reported in this work show that taking into account the relation between snap (the fourth derivative of position) and body torques and the geometry of $\mathrm{SO}(3)$ leads to multirotor vehicles reaching desired configurations in full state space with higher accuracy.

To avoid the singularities associated with local coordinates of $\mathrm{SO}(3)$, such as Euler angles, and the ambiguities arising from the representation of the orientation of a rigid-body using unit quaternions [25], the orientation dynamics are expressed in $\mathrm{SO}(3)$. An approach to approximating the solution of differential equations on $\mathrm{SO}(3)$ is to embed the system into $\mathbb{R}^{3\times 3}$ and use numerical integration algorithms developed for Euclidean space, while considering the smallest time step possible so as to mitigate the inaccuracies that result from not constraining the trajectories of the system to evolve on $\mathrm{SO}(3)$. Projecting the approximated solution back to the manifold after each integration step is usually what follows – although this may introduce additional errors in accuracy. Another approach is to propagate the orientation dynamics using discrete geometric integration algorithms so that not only the structure of the underlying manifold is preserved, but also the computation of approximated solutions to arbitrary accuracy is possible [26, 27].

With the trajectory planning and control strategy developed, a multirotor is swung at great speed into a loop and manoeuvres that require the close interaction of two vehicles and a tight coordination of their position and attitude are performed. The motivation behind executing the latter manoeuvres arises from the preliminary version of this work [28]. In-flight parcel relay manoeuvres have been introduced therein to overcome the stringent limitations on the range and endurance of a single vehicle and, thus, enable an effective package delivery.

The main contributions of this paper are: i) a trajectory planning strategy that enables multirotors to perform precise manoeuvres, i.e. the vehicles are driven to a desired configuration in full state space, and that takes into account their dynamical model and exploits the relation between snap and body torques; and ii) its validation by means of realistic simulations that consider the PX4 autopilot software in-the-loop [29] and experimental trials. Relative to the preliminary version of this work, this paper considers the intrinsic geometry of the manifold where multirotor orientation dynamics evolve to numerically integrate them, significantly extends the simulation results previously obtained, and presents experimental results.

This paper is structured as follows. Section 2 first provides some preliminaries on the geometry of SO(3), then details how to numerically integrate differential equations on this manifold, using the Crouch-Grossmann method to ensure that the structure of the underlying manifold is preserved and the dynamics are accurately integrated; in Section 3, the model of the multirotor is presented; a planning strategy for aggressive, precise manoeuvres is proposed in Section 4; with the objective of validating the generated trajectories, Section 5 presents a tracking controller that takes into accounts first-order drag terms; Section 6, the analysis of simulation and experimental results of aggressive manoeuvres; and, lastly, some concluding remarks are offered in Section 7.

2 Preliminaries

2.1 Notation

Throughout this work, the set of real numbers is denoted by $\mathbb{R}$, the subset of positive real numbers by $\mathbb{R}^+$, and the set of $n \times m$ matrices with real entries by $\mathbb{R}^{n\times m}$. The symbol $\mathbb{R}^n$ represents the real coordinate space of dimension n, equipped with the Euclidean norm, $\|\boldsymbol{x}\| := \sqrt{\boldsymbol{x}^\top \boldsymbol{x}}$, for all $\boldsymbol{x} \in \mathbb{R}^n$, while $\mathbb{S}^n := \{\boldsymbol{x} \in \mathbb{R}^{n+1} \mid \|\boldsymbol{x}\| = 1\}$. The $n \times n$ identity matrix is denoted by $\mathbf{I}_n$, and an $n \times m$ matrix of zeros by $\mathbf{0}_{n\times m}$. The map $\operatorname{diag} : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ given a vector $\boldsymbol{u}$ provides a square diagonal matrix with the elements of $\boldsymbol{u}$ on the main diagonal. Finally, $\mathbf{S}(\cdot)$ is an isomorphism between $\mathbb{R}^3$ and the space of 3×3 skew-symmetric matrices, $\mathfrak{so}(3)$, defined by the condition $\mathbf{S}(\boldsymbol{a})\boldsymbol{b} = \boldsymbol{a} \times \boldsymbol{b}$ for all $\boldsymbol{a},\ \boldsymbol{b} \in \mathbb{R}^3$, and $\mathbf{S}^{-1} : \mathfrak{so}(3) \to \mathbb{R}^3$ denotes the inverse map.

2.2 Geometry and Numerical Integration in SO(3)

The special orthogonal group of degree 3, denoted SO(3), is an algebraic subgroup of the group of invertible 3×3 matrices $\mathrm{GL}(3, \mathbb{R})$ and is defined as

$$\mathrm{SO}(3) := \left\{\mathbf{R} \in \mathbb{R}^{3\times 3} \mid \mathbf{R}^\top\mathbf{R} = \mathbf{I},\ \det(\mathbf{R}) = 1\right\}. \tag{1}$$

This group is suited to representing rotations in $\mathbb{R}^3$ and, for that reason, SO(3) is often referred to as the *group of rotations*.

2.2.1 Geometry

Some preliminaries regarding the geometry of SO(3) are hereby presented. To formally discuss the properties of this group, key concepts related to Riemannian geometry are necessary. This section is self-contained, however, those interested in reading further on the topics of this theory mentioned in the following paragraphs are referred to [30] and [31].

The group of rotations, in addition to being algebraic, is a connected and compact Lie group. Hence, SO(3) is a Riemannian manifold. At a rotation matrix $\mathbf{R}$, the tangent space is denoted by $\mathrm{T}_{\mathbf{R}}\mathrm{SO}(3)$. The tangent space at the identity corresponds to the set of 3×3 skew-symmetric matrices,

$$\mathfrak{so}(3) \triangleq \mathrm{T}_{\mathbf{I}}\mathrm{SO}(3) = \left\{\boldsymbol{\Omega} \in \mathbb{R}^{3\times 3} \mid \boldsymbol{\Omega} = -\boldsymbol{\Omega}^\top\right\}, \tag{2}$$

and endowed with the *commutator bracket* $[\boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2] = \boldsymbol{\Omega}_1\boldsymbol{\Omega}_2 - \boldsymbol{\Omega}_2\boldsymbol{\Omega}_1$ becomes the *Lie algebra* of SO(3). The tangent space at a given rotation $\mathbf{R}$ is obtained from $\mathfrak{so}(3)$ as follows,

$$\widetilde{\mathfrak{so}}(3) \triangleq \mathrm{T}_{\mathbf{R}}\mathrm{SO}(3) = \mathbf{R}\,\mathfrak{so}(3) = \left\{\mathbf{R}\boldsymbol{\Omega} \in \mathbb{R}^{3\times 3} \mid \boldsymbol{\Omega} = -\boldsymbol{\Omega}^\top\right\}, \tag{3}$$

and it also becomes a Lie algebra under the commutator bracket. As the linear map $\mathbf{R} : \mathfrak{so}(3) \to \widetilde{\mathfrak{so}}(3)$ is invertible and preserves brackets, that is, $\mathbf{R}[\boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2] = [\mathbf{R}\boldsymbol{\Omega}_1, \mathbf{R}\boldsymbol{\Omega}_2]$, for all $\boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2 \in \mathfrak{so}(3)$, $\mathfrak{so}(3)$ and $\widetilde{\mathfrak{so}}(3)$ are *isomorphic* as Lie algebras [32].

At every point of SO(3), the tangent space $\mathrm{T}_{\mathbf{R}}\mathrm{SO}(3)$ is equipped with a positive-definite inner product, $\mathrm{g}_{\mathbf{R}} : \mathrm{T}_{\mathbf{R}}\mathrm{SO}(3) \times \mathrm{T}_{\mathbf{R}}\mathrm{SO}(3) \to \mathbb{R}$ that varies smoothly with $\mathbf{R}$. Given the definition of SO(3) in Eq. 1, it follows that the Frobenius inner product is the natural metric,

$$\mathrm{g}_{\mathbf{R}}\left(\mathbf{R}\boldsymbol{\Omega}_1, \mathbf{R}\boldsymbol{\Omega}_2\right) = \operatorname{tr}\left(\boldsymbol{\Omega}_1^\top\mathbf{R}^\top\mathbf{R}\boldsymbol{\Omega}_2\right) = \operatorname{tr}\left(\boldsymbol{\Omega}_1^\top\boldsymbol{\Omega}_2\right), \tag{4}$$

and, as a result, the norm on $\mathrm{T}_{\mathbf{R}}\mathrm{SO}(3)$ is given by the Frobenius norm,

$$\|\mathbf{R}\boldsymbol{\Omega}\|^2_{\mathbf{R}} = \mathrm{g}_{\mathbf{R}}\left(\mathbf{R}\boldsymbol{\Omega}, \mathbf{R}\boldsymbol{\Omega}\right) = \mathrm{tr}\left(\boldsymbol{\Omega}^\top\boldsymbol{\Omega}\right) = \|\boldsymbol{\Omega}\|^2_{\mathrm{F}}\,. \tag{5}$$

Considering the problem at hand and the solution proposed hereafter, one important concept of Riemannian geometry is *geodesic* curves. A *geodesic* is a parameterised curve on the manifold whose acceleration in the tangential component is zero. They correspond to paths that would be traced out by a being constrained to remain on the manifold moving continuously forward with constant velocity. Once the being is given an initial velocity, its motion is completely determined by the *intrinsic* geometry of the manifold. In fact, given any $\boldsymbol{v} \in \mathrm{T}_{\mathbf{R}}\mathrm{SO}(3)$, there exists a unique geodesic $\boldsymbol{\gamma}(t) : \mathrm{I} \to \mathrm{SO}(3)$ such that $\boldsymbol{\gamma}(0) = \mathbf{R}$ and $\boldsymbol{\gamma}'(0) = \boldsymbol{v}$. For SO(3) being a compact Riemannian manifold, by the Hopf-Rinow theorem, the domain I of $\boldsymbol{\gamma}$ is the entire real line, $\mathrm{I} \coloneqq \mathbb{R}$.

At each rotation $\mathbf{R}$, all geodesics starting at $\mathbf{R}$ can be collected into a single mapping called the *exponential map*. Let $\boldsymbol{v} \in \mathrm{T}_{\mathbf{R}}\mathrm{SO}(3)$ and $\boldsymbol{\gamma}_v : \mathbb{R} \to \mathrm{SO}(3)$ be the geodesic starting at $\mathbf{R}$ with velocity $\boldsymbol{v}$. The exponential map is a locally defined mapping that allows one to go from a tangent space to the manifold. It is defined as the function such that $\exp_{\mathbf{R}}(\boldsymbol{v}) = \boldsymbol{\gamma}_v(1)$, for all $\boldsymbol{v} \in \mathrm{T}_{\mathbf{R}}\mathrm{SO}(3)$. Provided this definition of exponential map, the geodesic curve $\boldsymbol{\gamma}(t)$ that passes through $\mathbf{R}$ with velocity $\mathbf{R}\boldsymbol{\Omega} \in \mathrm{T}_{\mathbf{R}}\mathrm{SO}(3)$ at $t = 0$ is expressed as $\boldsymbol{\gamma} : t \mapsto \exp_{\mathbf{R}}(t\mathbf{R}\boldsymbol{\Omega})$. The exponential map of SO(3) is formally defined as

$$\begin{aligned} \exp_{\mathbf{R}} &: \mathrm{T}_{\mathbf{R}}\mathrm{SO}(3) \to \mathrm{SO}(3), \\ &\mathbf{R}\boldsymbol{\Omega} \mapsto \mathbf{R}\exp(\boldsymbol{\Omega}), \end{aligned} \tag{6}$$

where $\exp : \mathfrak{so}(3) \to \mathrm{SO}(3)$ corresponds to the exponential map at the identity and is defined as the matrix exponential of $\boldsymbol{\Omega}$, i.e. $\exp(\boldsymbol{\Omega}) \triangleq \mathrm{expm}(\boldsymbol{\Omega}) = \sum_{k=0}^{\infty}(1/k!)\boldsymbol{\Omega}^k$. In the sequel, let $\boldsymbol{\omega} \in \mathbb{R}^3$ and $\exp\left(\boldsymbol{\omega}\right) \triangleq \exp\left(\mathbf{S}(\boldsymbol{\omega})\right) \in \mathrm{SO}(3)$. For $\boldsymbol{\omega} = \mathbf{0}$, $\exp\left(\boldsymbol{\omega}\right) = \mathbf{I}$; for nonzero $\boldsymbol{\omega}$, $\exp\left(\boldsymbol{\omega}\right)$ can be computed using a closed-form expression that is known as the Rodrigues' formula [33],

$$\exp\left(\boldsymbol{\omega}\right) = \mathbf{I} + \frac{\sin\left(\|\boldsymbol{\omega}\|\right)}{\|\boldsymbol{\omega}\|}\mathbf{S}(\boldsymbol{\omega}) + \frac{\sin^2\left(\frac{\|\boldsymbol{\omega}\|}{2}\right)}{\frac{\|\boldsymbol{\omega}\|^2}{2}}\mathbf{S}^2(\boldsymbol{\omega}). \tag{7}$$

The logarithmic map of SO(3) corresponds to the *inverse* of the exponential map and is defined as

$$\begin{aligned} \log_{\mathbf{R}} &: \mathrm{SO}(3) \to \mathrm{T}_{\mathbf{R}}\mathrm{SO}(3), \\ &\mathbf{Q} \mapsto \mathbf{R}\log\left(\mathbf{R}^\top\mathbf{Q}\right), \end{aligned} \tag{8}$$

where $\log : \mathrm{SO}(3) \to \mathfrak{so}(3)$ denotes the the logarithmic map at the identity. To define it, note that any rotation matrix can be expressed as $\mathbf{R} = \exp\left(\theta\hat{\mathbf{n}}\right)$, with $\hat{\mathbf{n}} \in \mathbb{S}^2$ and $\theta \in [-\pi, \pi]$. It follows from Eq. 7 that the magnitude of the rotation θ about the rotation axis $\hat{\mathbf{n}}$ can be obtained from the trace of $\mathbf{R}$ [33],

$$\cos\theta = \frac{\mathrm{tr}(\mathbf{R}) - 1}{2}, \tag{9}$$

Since a rotation $\mathbf{R}$ through angle π about $\hat{\mathbf{n}}$ can also be represented as a rotation through $-\pi$ about the oppositely-oriented axis $-\hat{\mathbf{n}}$, the mapping $\theta \mapsto \exp\left(\theta\hat{\mathbf{n}}\right)$ is not one-to-one. Consequently, $\log\mathbf{R} \triangleq \mathbf{S}(\theta\hat{\mathbf{n}})$ is not well-defined for $|\theta| = \pi$, as there exist two vectors of length π that represent the same rotation. Restricting θ to the interval $[0, \pi)$, the matrix logarithm is uniquely defined,

$$\log\mathbf{R} = \begin{cases} \mathbf{0} & \text{if } \theta = 0, \\ \frac{\theta}{2\sin\theta}\left(\mathbf{R} - \mathbf{R}^\top\right) & \text{if } \theta \in (0, \pi). \end{cases} \tag{10}$$

Having defined the exponential and logarithmic maps of SO(3), a geodesic curve $\boldsymbol{\gamma}(t)$ from $\mathbf{R}_1$ to $\mathbf{R}_2$ can be expressed as

$$\begin{aligned} &\boldsymbol{\gamma} : [0, 1] \to \mathrm{SO}(3), \\ &t \mapsto \mathbf{R}_1\exp\left(t\log\left(\mathbf{R}_1^\top\mathbf{R}_2\right)\right). \end{aligned} \tag{11}$$

It is plain to see that $\boldsymbol{\gamma}(0) = \mathbf{R}_1$ and $\boldsymbol{\gamma}(1) = \mathbf{R}_2$. Computing the derivative of Eq. 11, one can verify that $\boldsymbol{\gamma}'(0) = \boldsymbol{\gamma}(0)\log\left(\mathbf{R}_1^\top\mathbf{R}_2\right) = \mathbf{R}_1\log\left(\mathbf{R}_1^\top\mathbf{R}_2\right) \in \mathrm{T}_{\mathbf{R}_1}\mathrm{SO}(3)$ and $\exp_{\mathbf{R}_1}\left(\mathbf{R}_1\log\left(\mathbf{R}_1^\top\mathbf{R}_2\right)\right) = \mathbf{R}_2$. In addition, one can confirm that the length of the velocity vector remains constant along the curve as

$$\left\|\boldsymbol{\gamma}'(t)\right\|_{\boldsymbol{\gamma}(t)} = \left\|\log\left(\mathbf{R}_1^\top\mathbf{R}_2\right)\right\|_{\mathrm{F}}.$$

2.2.2 Numerical Integration

Differential equations on SO(3) can be written as [34],

$$\dot{\mathbf{R}} = \mathbf{R}\,\boldsymbol{\Lambda}(t, \mathbf{R}), \quad \mathbf{R}(0) = \mathbf{R}_0 \in \mathrm{SO}(3), \tag{12}$$

where $\boldsymbol{\Lambda}(t, \mathbf{R}) \in \mathfrak{so}(3)$ and $\mathbf{R} \in \mathrm{SO}(3)$, for all $t \geq 0$. Approximating, or numerically integrating, the solution of Eq. 12 using numerical algorithms which consider that $\mathbf{R}$ lives in Euclidean space do not ensure that the approximated solution $\mathbf{R}_{k+1}$ belongs to SO(3), even if $\mathbf{R}_k \in \mathrm{SO}(3)$. Numerical integration methods that preserve the underlying structure of the group of rotations should be considered [26].

It is known that the solution to the initial value problem Eq. 12 is formally given by [27, Lemma 3.1],

$$\mathbf{R}(t) = \mathbf{R}_0\exp(\boldsymbol{\Omega}(t)), \tag{13}$$

for $t \geq 0$ and $\boldsymbol{\Omega} : \mathbb{R} \to \mathfrak{so}(3)$.

In this work, the Crouch-Grossman (CG) method [35] is employed to respect the geometry of SO(3) and ensure the accurate integration of the dynamics. Given $\mathbf{R}_k$, which corresponds to the value of $\mathbf{R}$ at time $t = t_k$, the approximation of the solution of Eq. 12 at time $t_{k+1} = t_k + h$, with $h \in \mathbb{R} \setminus \{0\}$ is provided by

$$\mathbf{R}_{k+1} = \mathbf{R}_k \prod_{i=1}^{s} \exp\left(hb_i \boldsymbol{\Lambda}^i\right). \tag{14}$$

The intermediate updates for $i = 1, 2, \ldots, s$ are computed by the following expressions,

$$\boldsymbol{\Lambda}^i = \boldsymbol{\Lambda}\left(t_k + c_i h, \mathbf{R}^i\right), \tag{15}$$

$$\mathbf{R}^i = \mathbf{R}_k \prod_{j=1}^{i-1} \exp\left(ha_{ij} \boldsymbol{\Lambda}^j\right). \tag{16}$$

The coefficients $[a_{ij}]$ in Eq. 16; $[b_i]$ in Eq. 14; and $[c_i]$ in Eq. 15 can be found for the CG method of order three in [35], as for methods of higher order, the reader is referred to [36]. Note that the number of stages s does not necessarily correspond to the order of the CG method considered.

The CG method guarantees that the approximation $\mathbf{R}_{k+1}$ lies on SO(3), and using Eq. 7 to determine the matrix exponential is computationally inexpensive. Multiplying exponential matrices comes with a cost, though, for the additional computations required when compared to conventional Runge-Kutta methods developed for Euclidean space.

3 Quadrotor Model

Let $\mathcal{W}$ denote the world reference frame, which corresponds to the canonical basis of $\mathbb{R}^3$, $\mathcal{W} = \{\boldsymbol{e}_i\}_{i=1,2,3}$ and $\mathcal{B} = \{\boldsymbol{x}_B, \boldsymbol{y}_B, \boldsymbol{z}_B\}$, the body frame attached to the vehicle with origin at the centre of mass. The axes $\boldsymbol{x}_B$ and $\boldsymbol{y}_B$ are orthogonal and lie in plane defined by the centre of the multiple rotors; $\boldsymbol{z}_B$ is perpendicular to this plane and points upwards, in the direction of total thrust.

The configuration of the multirotor is defined by the location of the centre of mass, $\boldsymbol{p} = [x\ y\ z]^\top$ and its orientation with respect to $\mathcal{W}$, represented by a rotation matrix $\mathbf{R} \in \mathrm{SO}(3)$ from $\mathcal{B}$ to $\mathcal{W}$. Let the linear velocity of the centre of mass in $\mathcal{W}$ be denoted by $\boldsymbol{v} = [v_x\ v_y\ v_z]^\top$ and the angular velocity of $\mathcal{B}$ relative to $\mathcal{W}$, expressed in $\mathcal{B}$, by $\boldsymbol{\omega} = [p\ q\ r]^\top$.

As the multirotor moves through the air, it is subject to aerodynamic forces $\mathbf{F}_a$ and torques $\boldsymbol{\tau}_a$, which depend on airspeed, i.e. the velocity of the vehicle with respect to the surrounding air mass. Let $\boldsymbol{v}_a \coloneqq \boldsymbol{v} - \boldsymbol{v}_w$ denote this velocity vector; $\boldsymbol{v}_w$ corresponds to the wind velocity relative to $\mathcal{W}$. In this work, it is considered that $\boldsymbol{v}_w = \mathbf{0}$.

Finally, the kinematics and dynamics of the system are described by the following differential equations,

$$\dot{\boldsymbol{p}} = \boldsymbol{v}, \tag{17}$$

$$m\dot{\boldsymbol{v}} = -mg\boldsymbol{e}_3 + T\mathbf{R}\boldsymbol{e}_3 + \mathbf{R}\mathbf{F}_a, \tag{18}$$

$$\dot{\mathbf{R}} = \mathbf{R}\mathbf{S}(\boldsymbol{\omega}), \tag{19}$$

$$\mathbf{J}\dot{\boldsymbol{\omega}} + \mathbf{S}(\boldsymbol{\omega})\mathbf{J}\boldsymbol{\omega} = \boldsymbol{\tau} + \boldsymbol{\tau}_a, \tag{20}$$

where the control input to the system is comprised of the total thrust $T \in \mathbb{R}^+$, and the body torques $\boldsymbol{\tau} \in \mathbb{R}^3$, produced by the four rotors. The constants $m \in \mathbb{R}^+$ and $\mathbf{J} \in \mathbb{R}^{3\times 3}$ represent the mass and the inertia tensor (about the frame $\mathcal{B}$) of the multirotor, respectively; and $g = 9.8\ \mathrm{m\,s^{-2}}$ denotes the acceleration due to gravity. The term $\mathbf{S}(\boldsymbol{\omega})\mathbf{J}\boldsymbol{\omega}$ in Eq. 20 is referred to as the *Coriolis term* in this work.

3.1 Aerodynamic Forces and Torques

To derive expressions that capture the aerodynamic effects acting upon the multirotor when it passes through the surrounding air mass, the simplifying assumption that the vehicle is equipped with rigid propellers has been made.

The model proposed in [37] for the aerodynamic forces $\mathbf{F}_a$ and torques $\boldsymbol{\tau}_a$ to which a multirotor is subject is considered. In the xy-plane of $\mathcal{B}$, the vehicle experiences a drag force, referred to as *rotor drag* or H-force in the helicopter literature, that is a function of airspeed. For quasi-stationary flight, that is, for small $\boldsymbol{v}_a$, *rotor drag* can be expressed linearly in terms of airspeed,

$$\mathbf{F}_a = -\mathbf{D}\mathbf{R}^\top \boldsymbol{v}_a. \tag{21}$$

The matrix $\mathbf{D} = \mathrm{diag}(d_x, d_y, d_z)$, with d_x and d_y non-negative and $d_z = 0$, is constant and contains the rotor drag coefficients along its diagonal. These coefficients can also account for the drag due to the fuselage of the rotorcraft and other sources of drag. The aerodynamic torques that result from the torques created by the difference in lift between the advancing and retreating blades and the drag force $\mathbf{F}_a$ are expressed as

$$\boldsymbol{\tau}_a = -\mathbf{A}\mathbf{R}^\top \boldsymbol{v}_a - \mathbf{B}\boldsymbol{\omega}, \tag{22}$$

with $\mathbf{A}$ and $\mathbf{B}$ being constant matrices that depend on the geometry and profile of the considered rotor blades.

3.2 Numerical Integration

To numerically integrate the system of differential equations that govern the dynamics of a multirotor vehicle, let $\boldsymbol{f} : \mathbb{R}^6 \times$

$\mathrm{SO}(3) \times \mathbb{R}^+ \rightarrow \mathbb{R}^6$ and $\boldsymbol{g} : \mathbb{R}^6 \times \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}^3$. Defining the variable $\boldsymbol{x} = [\boldsymbol{p}^\top \ \boldsymbol{v}^\top]^\top \in \mathbb{R}^6$, the vehicle dynamics can be expressed as

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\left(\boldsymbol{x}(t), \mathbf{R}(t), T(t)\right), \tag{23}$$

$$\dot{\mathbf{R}}(t) = \mathbf{R}(t)\mathbf{S}(\boldsymbol{\omega}(t)), \tag{24}$$

$$\dot{\boldsymbol{\omega}}(t) = \boldsymbol{g}(\boldsymbol{x}(t), \boldsymbol{\omega}(t), \boldsymbol{\tau}(t)), \tag{25}$$

and, thus, the state of the system lives in $\mathcal{M} := \mathbb{R}^6 \times \mathrm{SO}(3) \times \mathbb{R}^3$. Since $\mathbb{R}^n$ constitutes a Lie group under addition, the direct product of SO(3) and $\mathbb{R}^n$ is also a Lie group [32]. Hence, one can implement a *Lie group integrator* (i.e. a numerical integration method that respects Lie-group structure) on the product group $\mathcal{M}$ to numerically integrate the solution of the system of differential equations above. Implementing a *Lie group integrator* on $\mathcal{M}$ is equivalent to employing a suitable numerical method on each group, SO(3) and $\mathbb{R}^n$, separately and coupling them together, as detailed next.

Given $(\boldsymbol{x}_k, \boldsymbol{\omega}_k)$, which respectively correspond to the value of state variables $\boldsymbol{x}$ and $\boldsymbol{\omega}$ at time $t = t_k$, the solution to Eqs. 23 and 25 at time $t_{k+1} = t_k + h$, with $h \in \mathbb{R} \setminus \{0\}$, can be approximated by $(\boldsymbol{x}_{k+1}, \boldsymbol{\omega}_{k+1})$, which is computed using an explicit Runge-Kutta method [38] on $\mathbb{R}^6 \times \mathbb{R}^3$. As for computing the approximation of the solution to Eq. 24 at time $t = t_{k+1}$, provided the value of $\mathbf{R}$ at time $t = t_k$, the CG method – presented in Section 2.2.2 – is applied.

4 Planning for Aggressive Manoeuvres

The choice of the position $\boldsymbol{p}$ and the yaw angle ψ of the vehicle as *flat outputs* allows the states and inputs of the system to be expressed algebraically in terms of these outputs and a finite number of their derivatives [2, 39]. This property simplifies significantly the generation of aggressive trajectories for multirotors since it is not necessary to plan them in full state space.

The motion planning problem can be tackled by formulating an infinite-dimensional optimisation problem over a time horizon. Pontryagin's maximum principle can be employed to optimise a suitable cost functional subject to the translational multirotor dynamics, and to efficiently obtain optimal trajectories in the space of flat outputs (see e.g. [11] and [12]). In the context of this work, however, restricting the attitude of the vehicles is necessary to perform precise manoeuvres. To achieve this goal, the manoeuvres are divided into three separate phases to accommodate any combination of initial, desired, and final states and the reference trajectories for each of them are generated by formulating a convex optimisation problem in the space of *flat outputs*, as detailed in this section.

So that the desired manoeuvres can be accurately tracked by multirotors, in Section 4.1, a planning strategy is presented to design aggressive manoeuvres that take into account the vehicle dynamics by numerically integrating them – in the manifold they evolve – backwards in time. The backwards integration allows to find a *launch* configuration through which the manoeuvre reference trajectory passes so that a multirotor builds up the necessary momenta to reach the desired state.

In this work, trajectories are parametrised by piecewise functions $f_k : \mathbb{R} \rightarrow \mathbb{R}$, with $k = 0, 1, \ldots, N$, over M time intervals. These serve as a basis for generating smooth trajectories $\boldsymbol{\sigma}(t) := \left[\boldsymbol{p}(t)^\top \ \psi(t)\right]^\top$, in the space of flat outputs as follows,

$$\boldsymbol{\sigma}(t) = \begin{cases} \sum_{k=0}^{N} \boldsymbol{\sigma}_{k1} f_k(t) & t_0 \leq t < t_1, \\ \sum_{k=0}^{N} \boldsymbol{\sigma}_{k2} f_k(t) & t_1 \leq t < t_2, \\ \quad \vdots & \\ \sum_{k=0}^{N} \boldsymbol{\sigma}_{kM} f_k(t) & t_{M-1} \leq t \leq t_M, \end{cases} \tag{26}$$

where the vector $\boldsymbol{\sigma}_{kn} = [x_{kn} \ y_{kn} \ z_{kn} \ \psi_{kn}]^\top \in \mathbb{R}^4$ weighs the contribution of the basis function $f_k(\cdot)$ to express the trajectory segment $\boldsymbol{\sigma}_n(t), \ n = 1, \ldots, M$, which can be written as

$$\boldsymbol{\sigma}_n(t) = \boldsymbol{\Sigma}_n \boldsymbol{\eta}(t), \ t_{n-1} \leq t \leq t_n, \tag{27}$$

with $\boldsymbol{\Sigma}_n = [\boldsymbol{\sigma}_{0n} \ \boldsymbol{\sigma}_{1n} \cdots \boldsymbol{\sigma}_{Nn}] \in \mathbb{R}^{4\times(N+1)}$ and $\boldsymbol{\eta}(t) = [f_0(t) \quad f_1(t) \ \cdots \ f_N(t)]^\top \in \mathbb{R}^N$.

Most often the basis functions are polynomial functions of degree N, for being C^N and being computationally inexpensive to evaluate them using digital computers. In this work, polynomials are expressed in a monomial basis, i.e. $f_k(t) \triangleq t^k$. Although for this polynomial basis numerical issues are prone to arise when attempting to generate long trajectories with high degree polynomials, short time spanning trajectories can be efficiently generated and stitched together, ensuring the continuity of the first ℓ derivatives of $f_k(\cdot)$. The choice of a value of ℓ – that is suitable in the context of the present paper – is discussed later in this section.

A finite-dimensional optimisation problem is formulated in order to determine the vector coefficients $\boldsymbol{\sigma}_{kn}$, which define the trajectory that passes through a set of waypoints $\boldsymbol{w}_n$ at times $t_n, \ n = 0, \ldots, M$. To coordinate the orientation of the vehicle, the desired orientation $\mathbf{R}_{des}$ is associated with each of the waypoints. As $\boldsymbol{z}_B$ points in the direction of thrust, the orientation of the vehicle, apart from the yaw angle, is determined by the direction of this axis. In this fashion, from $\mathbf{R}_{des}$ the desired $\boldsymbol{z}_B$ is obtained,

$$\boldsymbol{z}_{B,des}(t_n) = \mathbf{R}_{des}(t_n)\boldsymbol{e}_3. \tag{28}$$

Considering that $\boldsymbol{F}_a = \mathbf{0}$ and $\boldsymbol{\tau}_a = \mathbf{0}$, the linear velocity dynamics equation presented in Eq. 18 is used to formulate the following linear constraint on the linear acceleration of the vehicle. In this manner, a desired direction of thrust is imposed,

$$\left(\mathbf{I}_3 - \boldsymbol{z}_{B,des}\boldsymbol{z}_{B,des}^\top\right)\left(\dot{\boldsymbol{v}} + g\boldsymbol{e}_3\right) = \mathbf{0}. \tag{29}$$

To link both the direction and magnitude of thrust to $\boldsymbol{w}_n$,

$$\dot{\boldsymbol{v}} = \left(-g\mathbf{I}_3 + \frac{T_{des}}{m}\mathbf{R}_{des}\right)\boldsymbol{e}_3, \tag{30}$$

where T_{des} corresponds to the desired magnitude of thrust.

The cost function of the proposed optimisation problem is such that it minimises the integral of the squared norm of snap (the fourth derivative of $\boldsymbol{p}$) of the trajectory and the second derivative of ψ,

$$\begin{aligned} \text{minimise} \quad & \int_{t_0}^{t_M} \mu_p \left\| \frac{d^4\boldsymbol{p}}{dt^4} \right\|^2 + \mu_\psi \frac{d^2\psi}{dt^2}^2 \, dt \\ \text{s.t.} \quad & \boldsymbol{\sigma}(t_n) = \boldsymbol{w}_n, \ \text{for } n = 0, \ldots, M, \\ & \dot{\boldsymbol{\sigma}}(t_n) = \boldsymbol{v}_n, \ \text{for } n = 0, \ldots, M, \\ & \ddot{\boldsymbol{\sigma}}(t_n) = \boldsymbol{a}_n, \ \text{for } n = 0, \ldots, M, \end{aligned} \tag{31}$$

where $\mu_p, \mu_\psi \in \mathbb{R}$, and $\boldsymbol{w}_n, \boldsymbol{v}_n, \boldsymbol{a}_n \in \mathbb{R}^4$ correspond to the desired values for the flat outputs and their first two time derivatives at time $t = t_n$. The meaningfulness of the cost function selected lies in the fact that the input body torques $\boldsymbol{\tau}$ are algebraically related to the snap of the vehicle and the second derivative of the yaw angle [2]. This optimisation problem can be reformulated as a quadratic program (QP),

$$\begin{aligned} \underset{\boldsymbol{c}}{\text{minimise}} \quad & \boldsymbol{c}^\top \mathbf{H} \boldsymbol{c} + \boldsymbol{s}^\top \boldsymbol{c} \\ \text{s.t.} \quad & \mathbf{V}\boldsymbol{c} = \boldsymbol{d}, \end{aligned} \tag{32}$$

where vector $\boldsymbol{c}$ contains $4(N+1)M$ optimisation variables which correspond to the polynomial coefficients that parameterise the whole trajectory $\boldsymbol{\sigma}_{kn}$ – to readily find the solution to Eq. 31. The constants $\mathbf{H}$, $\boldsymbol{s}$, $\mathbf{V}$, $\boldsymbol{d}$ are obtained from Eq. 31, considering the trajectory parameterisation detailed in Eq. 26.

So as to guarantee the smoothness of the planned trajectories, constraints are included to ensure the continuity of the first four derivatives of $\boldsymbol{p}$ and the first two derivatives of ψ (therefore, it follows that $\ell = 4$). In particular, to achieve the desired orientation at each waypoint, it is necessary to impose the continuity of linear acceleration and jerk trajectories to ensure the continuity of orientation and roll and pitch rates trajectories. However, the continuity of these position derivatives alone is not sufficient to make sure that a quadrotor is able to follow accurately every single trajectory. Saturation and rate limit constraints may impede the execution of some manoeuvres.

With the objective of further ensuring the feasibility of trajectories, one may also enforce bound constraints on the position time derivatives to incorporate saturation and rate limit constraints into Eq. 31. Building upon Eq. 31 another optimisation problem can be formulated to plan trajectories in a centralised fashion for two multirotors while avoiding inter-vehicle collisions, as detailed in preliminary version of this work [28]. To plan trajectories for multiple vehicles simultaneously, the cost function of the optimisation problem is given by a sum of separate functions, each given by the integral of the norm of the snap and second time yaw derivatives of each vehicle.

4.1 Backwards Integration Strategy

In spite of the incorporation of Eqs. 29 or 30 into 31, the optimisation strategy does not necessarily imply that any controller tracking acceleration references can provide inputs to the vehicle such that it follows these references accurately. Since the multirotors performing aggressive, precise manoeuvres must achieve some concrete configuration in state space, a strategy is devised so that the vehicles are driven to a goal state G, consisting in commanding the attitude subsystem of the vehicles to describe geodesic curves and dividing the manoeuvres into three separate phases.

The motivation behind making multirotors trace out geodesics during the backwards integration stems from the relation between snap and body torques. Let $\boldsymbol{s}$ denote the snap of the vehicle. Taking $\boldsymbol{F}_a = \mathbf{0}$, the second time derivative of Eq. 18 is given by

$$m\boldsymbol{s} = \ddot{T}\boldsymbol{z}_B + 2\dot{T}\mathbf{S}(\mathbf{R}\boldsymbol{\omega})\boldsymbol{z}_B + T\left(\mathbf{S}\left(\mathbf{R}\dot{\boldsymbol{\omega}}\right)\boldsymbol{z}_B + \mathbf{S}^2(\mathbf{R}\boldsymbol{\omega})\boldsymbol{z}_B\right), \tag{33}$$

and projecting it onto the plane orthogonal to $\boldsymbol{z}_B$ yields

$$\begin{aligned} m\left(\boldsymbol{s} - \left(\boldsymbol{z}_B^\top\boldsymbol{s}\right)\boldsymbol{z}_B\right) &= T\mathbf{S}(\mathbf{R}\dot{\boldsymbol{\omega}})\boldsymbol{z}_B + 2\dot{T}\mathbf{S}(\mathbf{R}\boldsymbol{\omega})\boldsymbol{z}_B + \\ &\quad T\left(\mathbf{S}^2(\mathbf{R}\boldsymbol{\omega})\boldsymbol{z}_B - \left(\boldsymbol{z}_B^\top\mathbf{S}^2(\mathbf{R}\boldsymbol{\omega})\boldsymbol{z}_B\right)\boldsymbol{z}_B\right) \\ &= T\left((\dot{q} + pr)\boldsymbol{x}_B + (-\dot{p} + qr)\boldsymbol{y}_B\right) + 2\dot{T}(q\boldsymbol{x}_B - p\boldsymbol{y}_B). \end{aligned} \tag{34}$$

By providing zero body torques and constant thrust during the backwards integration, and starting with the third component of angular velocity $r = 0$, it follows from Eq. 34 that the translational subsystem of the vehicle describes a trajectory whose snap $\boldsymbol{s}$ in the x and y components of the body frame is zero. Furthermore, the angular velocity of the vehicle $\boldsymbol{\omega}$ remains constant and, therefore, the attitude subsystem of the vehicle describes a geodesic curve on SO(3). In this way, the *launch* configuration – from where the vehicle is thrust towards the desired state in full state space (*vide* Fig. 1) – encodes the information necessary to plan minimum snap trajectories to the desired state that demand small body torques to be tracked by multirotor vehicles.

It is established that the trajectory for each vehicle is composed of three segments, as illustrated in Fig. 1. In order to generate them, G is provided by specifying the desired position $\boldsymbol{p}_G$, velocity $\boldsymbol{v}_G$, orientation $\mathbf{R}_G$, and angular velocity $\boldsymbol{\omega}_G$. The three-phases manoeuvre design enables planning trajectories to G that pass through the launch configuration so that the momenta necessary to reach it is achieved, and to ensure the accurate tracking of reference trajectories. Moreover, planning feasible trajectories from G to a final configuration is facilitated since they start at a well-defined point in full state space, i.e. the goal configuration also includes – in addition to a desired position, linear velocity, and acceleration – a desired angular velocity. Each of the three segments is detailed as follows:

- The segment *Phase 1*, represented as a blue dashed line, corresponds to the trajectory from an initial position (denoted by a green cross in Fig. 1) to a launch position $\boldsymbol{p}_L$ from where the vehicle heads towards the goal position $\boldsymbol{p}_G$ with a launch velocity $\boldsymbol{v}_L$ and a suitable orientation $\mathbf{R}_L$, regarding the desired trajectory;
- For *Phase 2*, depicted as a continuous orange line, the trajectory that is generated for this segment takes into account that the quadrotor must reach the desired goal position $\boldsymbol{p}_G$, with a pre-defined velocity $\boldsymbol{v}_G$, orientation $\mathbf{R}_G$, and angular velocity $\boldsymbol{\omega}_G$;
- In the final segment, *Phase 3*, coloured in green, the vehicle is provided a trajectory from $\boldsymbol{p}_G$ until a final position (denoted by a red cross in Fig. 1). It may be imposed that the vehicle finishes at hover.

To obtain the launch point $\boldsymbol{p}_L$ and velocity $\boldsymbol{v}_L$ necessary for *Phase 2*, the multirotor dynamics are integrated backwards in time using the numerical integration algorithm detailed in Section 3.2 and starting from the desired goal state G, while considering that $\mathbf{F}_a = \mathbf{0}$ and $\boldsymbol{\tau}_a = \mathbf{0}$. The desired angular velocity at G is set to

$$\boldsymbol{\omega}_G = -\frac{1}{\mathcal{T}}\mathbf{S}^{-1}\left(\log\left(\mathbf{R}_G^\top \mathbf{R}_L\right)\right), \tag{35}$$

with $\mathcal{T}$ denoting the time duration of *Phase 2*. Providing zero body torques and constant thrust as inputs to the vehicle, the angular velocity remains constant during the backwards integration, $\boldsymbol{\omega}(t) = \boldsymbol{\omega}_G$, for all $t \in [0, \mathcal{T}]$, and the attitude subsystem of the multirotor describes a geodesic curve $\mathbf{R}_{des} : [0, \mathcal{T}] \to \mathrm{SO}(3)$ that passes through $\mathbf{R}_L$ at time $t = 0$ and $\mathbf{R}_G$ at time $t = \mathcal{T}$,

$$\mathbf{R}_{des}(t) = \mathbf{R}_G \exp\left(\frac{\mathcal{T} - t}{\mathcal{T}} \log\left(\mathbf{R}_G^\top \mathbf{R}_L\right)\right). \tag{36}$$

When $\mathbf{R}_L$ and $\mathbf{R}_G$ are apart by a rotation of π radians, i.e. $\mathbf{R}_G^\top \mathbf{R}_L = \exp\left(\pi \mathbf{S}(\hat{\mathbf{n}})\right)$, with $\hat{\mathbf{n}} \in \mathbb{S}^2$, the logarithm is not unique. To solve the ambiguity that arises in this case, the eigenvector associated with the eigenvalue equal to 1 of $\mathbf{R}_G^\top \mathbf{R}_L$ is scaled by π to compute the $\log(\cdot)$. Given the trajectory that one wishes the drone to describe, the thrust input $T_{des} : [0, \mathcal{T}] \to \mathbb{R}^+$ should be chosen accordingly. From Eq. 34, this input should be kept constant. Setting $T_{des}(t) = mg$, for all $t \in [0, \mathcal{T}]$, is suitable for most manoeuvres, such as the challenging ones presented in Section 6. Setting a higher value for $T_{des}(\cdot)$ enables multirotors to perform more aggressive manoeuvres at the expense of higher necessary momenta at the start of *Phase 2*, though.

It is possible to define a time-varying $T_{des}(\cdot)$ and obtain snap optimal trajectories that also require small torques to be tracked. When it comes to multirotor vehicles, however, the variation of the thrust input T over time may lead to the angular rates not being constant since

$$T = m\|\dot{\boldsymbol{v}} + g\boldsymbol{e}_3\|, \qquad p = -\frac{m}{T}\left(\boldsymbol{y}_B^\top \ddot{\boldsymbol{v}}\right), \qquad q = \frac{m}{T}\left(\boldsymbol{x}_B^\top \ddot{\boldsymbol{v}}\right).$$

The launch configuration that is obtained by numerically integrating the rotorcraft dynamics backwards in time might result in optimal trajectories to the goal configuration that demand considerable body torques, as the orientation of the vehicle does not trace out a geodesic curve on SO(3) during the backwards integration.

When considering time-varying thrust inputs, it is additionally important to have in mind that the thrust and torque inputs are intertwined, since both quantities depend on the rotational speed of the propellers of the multirotor. The trajectories segments for *Phase 2* that are planned considering an inadequate choice of $T_{des}(\cdot)$ during the backwards integration of the dynamics may require unfeasible thrust and

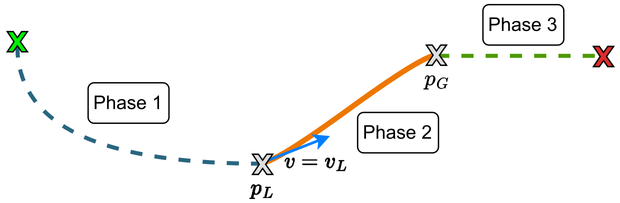


Fig. 1 Diagram illustrating the three phases of the proposed strategy to perform precise manoeuvres. In Phase 1, the vehicle moves from an initial position (green cross) to the launch position so that it builds up the momenta required for the next phase; in Phase 2, the vehicle aims to reach a goal position with a suitable launch velocity and orientation to achieve a desired configuration with precision; in Phase 3, the vehicle heads towards a final position (red cross)

torque inputs, in the sense that a set of rotational velocities for the rotors of the vehicle to produce the desired pair of inputs does not exist.

Each of the three trajectory segments is obtained by formulating and solving Eq. 31. Between the trajectory segments the continuity of the first four time derivatives of position is enforced to meet the smoothness requirement of the attitude trajectories. So that the vehicle hovers at the end of the manoeuvre, all position time derivatives are set to zero.

At the launch waypoint a linear acceleration constraint, in the form of Eq. 30, embeds the desired initial value of thrust for *Phase* 2, $T_{des}(0)$, and orientation $\mathbf{R}_L$ into the motion planning problem. As for the goal waypoint, a similar constraint is imposed for $T_{des}(\mathcal{T})$ and $\mathbf{R}_G$. Position and linear velocity constraints incorporate $\boldsymbol{p}_L$ and $\boldsymbol{p}_G$, and $\boldsymbol{v}_L$ and $\boldsymbol{v}_G$ into the optimisation problem, respectively. Despite imposing these constraints, the optimal acceleration trajectories obtained from solving Eq. 31 do not lead to geodesic curves between the rotations $\mathbf{R}_L$ and $\mathbf{R}_G$ because they are polynomial functions. In practice, though, imposing these constraints when generating minimum snap trajectories enables designing trajectory segments for *Phase 2* that require small body torques to be tracked, under the condition that the thrust input $T_{des}(\cdot)$ is kept constant (or slowly varying around a constant value) and zero body torques are provided as input to the vehicle during the backwards integration.

Having the knowledge that the multirotor dynamics are differentially flat (with *flat outputs* given by the position and the yaw angle), the rotational kinetic energy $\mathrm{KE}(t) = {}^1\!/_2\boldsymbol{\omega}^\top(t)\mathbf{J}\boldsymbol{\omega}(t)$ associated with reference trajectories can be computed. Using the *Phase 2* reference trajectories for the loop manoeuvre (Section 6.3) and the parcel relay manoeuvre (Section 6.4.1), and the geodesic curves that were considered to plan each of them, the corresponding KE trajectories are plotted in Fig. 2 to support the hypothesis regarding small torques being necessary to track the *Phase 2* reference trajectories. The plots demonstrate that the KE curves of the minimum snap trajectories gently undulate and only slightly deviate from the constant KE curves, which are associated with the geodesics.

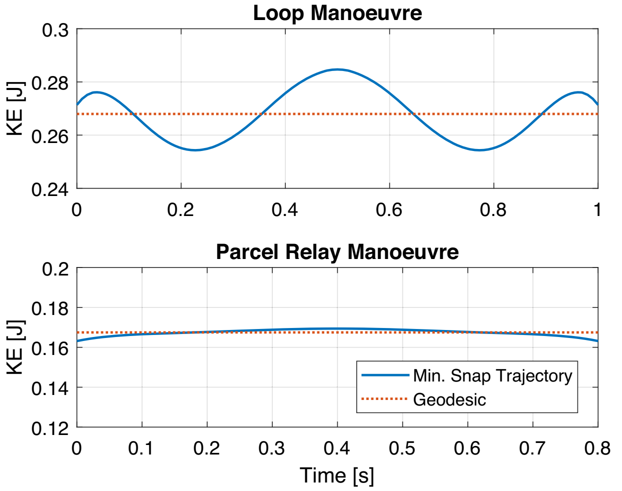


Fig. 2 Rotational kinetic energy associated with the Phase 2 trajectory segments of the loop manoeuvre (Section 6.3) and the parcel relay manoeuvre (Section 6.4.1), and of the geodesic curves that were considered to plan each of them

5 Tracking Controller

The translation and attitude of the multirotor are controlled using an inner-outer loop design. The theoretical findings detailed in [39] are here considered, which show that the multirotor dynamics expressed as in Section 3 are *differentially flat* – with flat outputs given by the position $\boldsymbol{p}$ and the yaw angle ψ.

The outer loop of the controller is responsible for tracking a trajectory defined by the reference position $\boldsymbol{p}_T$, linear velocity $\boldsymbol{v}_T$, linear acceleration $\boldsymbol{a}_T$, and linear jerk $\boldsymbol{j}_T$. This loop computes the desired thrust vector $\mathbf{F}_{des}$, which together with a yaw angle ψ_T, defines the desired body frame orientation $\mathbf{R}_{des}$. It also determines the desired angular rates $\boldsymbol{\omega}_{des}$ from the trajectory references mentioned. A low-level inner-loop system coupled to the vehicle that operates in hard real-time and benefits from running at high frequency is considered to provide a body torque input $\boldsymbol{\tau}$. Given the desired angular rates, the torque control law chosen is of the following form,

$$\boldsymbol{\tau} = -\mathbf{K}_\omega\left(\boldsymbol{\omega} - \boldsymbol{\omega}_{des}\right), \tag{37}$$

and it counteracts the aerodynamic torques and the Coriolis term by setting a sufficiently large gain $\mathbf{K}_\omega$. As both quantities are small in magnitude for virtually all flight regimes of small multirotor vehicles, a proportional gain is sufficient to provide stability against these disturbances.

The position and linear velocity errors are defined as

$$\boldsymbol{e}_p = \boldsymbol{p} - \boldsymbol{p}_T, \tag{38}$$

$$\boldsymbol{e}_v = \boldsymbol{v} - \boldsymbol{v}_T, \tag{39}$$

and are used to define a PD control law,

$$\mathbf{F}_{des} = -\mathbf{K}_p\boldsymbol{e}_p - \mathbf{K}_v\boldsymbol{e}_v + m\left(g\boldsymbol{e}_3 + \boldsymbol{a}_{ff}\right), \tag{40}$$

where the feedforward acceleration term $\boldsymbol{a}_{ff}$ is computed from the reference trajectory and includes a term to counteract the accelerations caused by drag,

$$\boldsymbol{a}_{ff} = \mathbf{R}_T\mathbf{D}\mathbf{R}_T^\top\boldsymbol{v}_T + \boldsymbol{a}_T. \tag{41}$$

How to determine $\mathbf{D}$ is detailed in Section 6.1; the rotation matrix $\mathbf{R}_T$ is computed from the reference trajectory and the drag coefficients [39]; the matrices $\mathbf{K}_p$ and $\mathbf{K}_v$ are diagonal, positive-definite and correspond to the position and velocity controller gains. The thrust input is obtained from the projection of $\mathbf{F}_{des}$ onto z_B,

$$T = \mathbf{F}_{des}^{\top} z_B. \tag{42}$$

To compute the desired orientation of the body frame $\mathbf{R}_{des}$, provided that z_B points in the direction of total thrust and $\|\mathbf{F}_{des}\| \neq 0$, the desired $z_{B,des}$ is

$$z_{B,des} = \frac{\mathbf{F}_{des}}{\|\mathbf{F}_{des}\|}. \tag{43}$$

Parameterising the vehicle orientation using Euler Angles, $\mathbf{R} = \mathbf{R}_3(\psi)\mathbf{R}_2(\theta)\mathbf{R}_1(\phi)$, where, for instance, $\mathbf{R}_3(\psi)$ corresponds to a rotation through the yaw angle ψ about $\boldsymbol{e}_3$, the desired orientation $\mathbf{R}_{des} = [\boldsymbol{x}_{B,des}\ \boldsymbol{y}_{B,des}\ z_{B,des}]$ can be readily computed. The x-axis of the body frame $\boldsymbol{x}_B = \mathbf{R}_3(\psi)\mathbf{R}_2(\theta)\mathbf{R}_1(\phi)\boldsymbol{e}_1$, and $\boldsymbol{e}_2^{\top}\mathbf{R}_3^{\top}(\psi)\boldsymbol{x}_B = \boldsymbol{e}_2^{\top}\mathbf{R}_2(\theta)\boldsymbol{e}_1 = 0$. This implies that $\boldsymbol{x}_B$ in addition to being orthogonal to z_B is also orthogonal to $\mathbf{R}_3(\psi)\boldsymbol{e}_2$. Given $z_{B,des}$ and a reference yaw angle ψ_T, the desired x and y axes of the body frame are determined as follows,

$$\boldsymbol{x}_{B,des} = \pm\frac{\boldsymbol{y}_C \times z_{B,des}}{\|\boldsymbol{y}_C \times z_{B,des}\|}, \quad \boldsymbol{y}_{B,des} = z_{B,des} \times \boldsymbol{x}_{B,des}, \tag{44}$$

where $\boldsymbol{y}_C = \mathbf{R}_3(\psi_T)\boldsymbol{e}_2$. To select the sign of $\boldsymbol{x}_{B,des}$, each of the two solutions is compared with the actual orientation of the multirotor to verify which one is closer.

Remark 1 (Uniqueness of the desired rotation matrix) The desired rotation matrix is not unique if $\boldsymbol{y}_C \times z_B = \mathbf{0}$, i.e. $\boldsymbol{y}_C$ is parallel to z_B, the vehicle rolls up to the point that the desired thrust vector lies in the xy plane of $\mathcal{W}$ and is aligned with $\boldsymbol{y}_C$. Moreover, at points near this singularity the rotation matrix is sensitive to large changes. To address such singularity, ψ_T has to be adjusted to ensure $\boldsymbol{y}_C$ is not aligned with z_B.

As for determining the desired angular rates, the following control law is considered,

$$\boldsymbol{\omega}_{des} = -\mathbf{K}_{\mathbf{R}}\boldsymbol{e}_{\mathbf{R}} + \boldsymbol{\omega}_{ff}, \tag{45}$$

where $\mathbf{K}_{\mathbf{R}}$ is a diagonal, positive-definite matrix and the feedforward term $\boldsymbol{\omega}_{ff}$ is determined from the reference trajectory and the drag coefficients [39]. Based on [40], the rotation error $\boldsymbol{e}_{\mathbf{R}}$ is defined as

$$\boldsymbol{e}_R = \frac{1}{2}\mathbf{S}^{-1}\left(\mathbf{R}_{des}^{\top}\mathbf{R} - \mathbf{R}^{\top}\mathbf{R}_{des}\right). \tag{46}$$

Proofs of stability and convergence are also presented for a similar controller in [40] but with: i) the inclusion of feedforward terms including the angular acceleration; ii) the addition of feedback terms cancelling the Coriolis term in Eq. 37; and iii) the assumption that all gain matrices are scalar multiples of the identity. The realisation of the controller presented in this work is different and does not quite satisfy all the assumptions listed above. However, as demonstrated in the results presented in Section 6, vehicles equipped with the controller proposed exhibit satisfactory tracking performance even when performing aggressive manoeuvres that require large excursions from the hover condition.

6 Experimental and Simulation Results

In order to validate the proposed approach for planning aggressive and precise manoeuvres, a performance comparison with existing work is conducted and next the following two scenarios are considered:

S1. flying a loop;
S2. performing a parcel relay manoeuvre.

The first scenario is addressed in a realistic simulation environment with the PX4 firmware in-the-loop to control computer-modelled multirotors carrying a high fidelity sensor suite on-board. At the heart of this environment is the Gazebo simulator [41]. The second scenario is addressed in the described simulation environment and in an indoor flying area featuring an OptiTrack motion capture system [42], comprised of eight cameras, which provides position and yaw angle measurements for the multirotors in the arena with millimetre and sub-degree accuracy at 60 Hz. To further demonstrate the suitability of the planning strategy proposed, the vehicles execute 1,000 trials in the simulation environment for each scenario. A video of the experiments performed is available at https://youtu.be/azqqJ64ueuo .

In the latter application scenario, it has been assumed that both drones have a gripping mechanism attached to their respective bottom. Initially, one of the vehicles is carrying a package attached to its bottom, while the other is not. Given a relay point, that is, a point in a global frame where the package is to be transferred, the two rotorcraft head towards that point while simultaneously adjusting their relative orientation in order to exchange a parcel between them. An illustration of a parcel relay manoeuvre is depicted in Fig. 3. While the problem of controlling a mechanism responsible for holding and transferring packages between vehicles is not addressed in this work, designing trajectories that facilitate performing these manoeuvres offers an interesting challenge as the vehicles must reach designated positions and orientations with high precision.

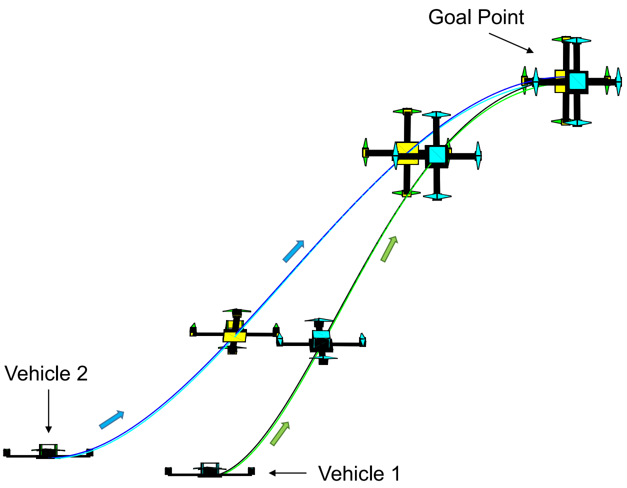


Fig. 3 Illustration of a parcel relay manoeuvre between two multirotors. It is considered that these are equipped with a gripping mechanism attached to their bottom. The two aerial vehicles vary their orientation and shorten the distance between them simultaneously until a goal configuration is achieved to exchange a parcel

The Intel® Aero Ready to Fly quadrotor, equipped with the PX4 autopilot has been employed to conduct experiments in the arena. The capabilities of the flight stack have been used for sensor fusion between the several on-board sensors and the motion capture system data, and for interaction with the aerial vehicles. Takeoff and landing manoeuvres have been conveniently handled to the autopilot by setting suitable flight modes.

A testing software environment has been developed in C++ to efficiently implement tracking controllers for quadrotors being controlled by a single off-board computer, where the Robot Operating System (ROS) is used to communicate with the autopilot on board the rotorcraft. The same software environment is also used to test controllers in the Gazebo simulator. The multirotor considered in these simulations is the 3DR® IRIS quadrotor. Additional software has also been designed to generate synthetic motion capture system data with the objective of creating a simulation environment in Gazebo that more closely resembles the considered indoor flying arena. To obtain pseudo external vision pose estimates, white Gaussian noise is added to the pose of the vehicle obtained from the Gazebo model states topic, and the resulting noisy pose of the vehicle is forwarded to the autopilot.

Simulations have also been carried out in MATLAB, considering the dynamical system detailed in Section 3, described by the physical parameters of the 3DR® IRIS quadrotor. The values of the physical parameters of this platform and of the Intel® Aero Ready to Fly quadrotor are in Table 1.

In what concerns the choice of the off-board controller gains, one should have in mind that the position tracking error converges to zero when there is no attitude tracking error [40] and, therefore, the attitude controller gains have to be carefully tuned. To ensure accurate tracking of both position and orientation references, the controller gains of the 3DR® IRIS and Intel® Aero Ready to Fly quadrotors have been set to $\mathbf{K}_p = 12.0\mathbf{I}_3$, $\mathbf{K}_v = 9.0\mathbf{I}_3$, and $\mathbf{K}_\mathbf{R} = 3.0\mathbf{I}_3$. As for determining the launch configuration of the vehicles, the quadrotor dynamics have been integrated backwards in time with the launch orientation $\mathbf{R}_L = \mathbf{I}_3$.

Table 1 Physical parameters of the multirotots used for experiments and simulations

		Intel Aero RTF	3DR IRIS
m	[kg]	1.35	1.52
J_{xx}	[kg m]	—	0.0295
J_{yy}	[kg m]	—	0.0295
J_{zz}	[kg m]	—	0.0596
μ	[kg m]	9.5×10^{-6}	$8.548\,58 \times 10^{-6}$
λ	[kg]	$3.064\,28 \times 10^{-4}$	$3.064\,28 \times 10^{-4}$

Throughout the plots presented in this section, the colour scheme adopted in Section 4.1 is considered: *Phase 1* is associated with the colour blue; *Phase 2*, with the colour orange; and *Phase 3*, with the colour green.

6.1 Identification

To control the vehicles in thrust, the desired throttle $T_{in} \in [0, 1]$ must be input to the autopilot and, therefore, functions that relate thrust T to throttle T_{in} are necessary. For the Intel® Aero Ready to Fly quadrotor, experimental points were acquired (T_{in}, T) by measuring the static thrust, i.e. thrust generated when the airspeed is zero by the four propellers in free air. To measure this quantity as accurately as possible, a sufficiently rigid, vertical 1-metre pole with discs at its top and bottom was fixed to a scale and the quadrotor was fastened to the top disc. The $\arctan(\cdot)$ function was selected to describe the relationship between the two variables,

$$T = a \arctan\left(bT_{in} + c\right) + d, \tag{47}$$

with $a = 8.84$, $b = 2.995$, $c = -1.478$, and $d = 10.37$.

As for the 3DR® IRIS quadrotor, to gather (T_{in}, T) data points, the mass of the simulated vehicle was varied linearly. For every value of mass, the drone was commanded to hover so as to obtain the corresponding value of throttle. The curve that best fitted the data points was a quadratic $T = \alpha T_{in}^2 + \beta T_{in} + \gamma$, with $\alpha = 34.068$, $\beta = 7.1202$, and $\gamma = -1 \times 10^{-2}$.

From blade element momentum (BEM) theory, it follows that the standard model for the thrust produced by a propeller is given by

$$T_p = \frac{\rho D^4}{4\pi^2}\omega_p^2 C_T, \tag{48}$$

where ρ is the density of the air; D, the diameter of the propeller; ω_p, its rotational speed in radians per second; and C_T, the thrust coefficient – which depends not only on the geometry and profile of the blades, but also on the geometry of the airflow [43]. More concretely, C_T can be expressed as a function of the *advance ratio* $J = 2\pi \left\|\boldsymbol{v}_a\right\| / (\omega_p D)$. For sufficiently large J, how large depends on the geometry of the propeller, C_T decreases roughly in linear fashion as J increases.

The motor-propeller model considered in the Gazebo simulator incorporates the airspeed of the vehicle into the standard model for the thrust produced by a hovering rotor in free air [44],

$$T_p = k_t(\boldsymbol{v}_a)\, \omega_p^2, \tag{49}$$

and the coefficient k_t is given by

$$k_t(\boldsymbol{v}_a) = \frac{\rho D^4}{4\pi^2} C_{T0} \max\left\{0,\ 1 - \left\|\boldsymbol{v}_a\right\| / 25\right\},$$

where the constant C_{T0} corresponds to the thrust coefficient at $J = 0$ and can be determined from static thrust tests.

To take into account the model implemented in Gazebo, the quadratic thrust curve determined for the 3DR® IRIS quadrotor was linearised around the hover condition, so that it is simple to invert, resulting in

$$\frac{T}{f(\boldsymbol{v}_a)} = a(T_{in} - b) + c, \quad \left\|\boldsymbol{v}_a\right\| < 25\,\mathrm{m\,s^{-1}}, \tag{50}$$

with constants $a = 45.344$, $b = 0.561$, and $c = mg$. The function $f : \mathbb{R}^3 \to \mathbb{R}$, $f(\boldsymbol{v}_a) = 1 - \left\|\boldsymbol{v}_a\right\| / 25$ so that the thrust-vs-throttle curve befits the Gazebo propeller model. In practice, though, as for small multirotors the rotational speed of the propellers, or, equivalently, the propeller tip speed, is significantly greater than airspeed, $J \approx 0$ for virtually all flight regimes of these vehicles. The inclusion of the function f into Eq. 50 also allows to capture how *rotor drag* imposes a tighter bound on the amount of thrust that can be generated by the spinning propellers.

Furthermore, in the Gazebo simulator the model for *rotor drag* is expressed as follows,

$$\mathbf{F}_D = -\omega_p \lambda \left(\mathbf{I}_3 - \boldsymbol{e}_3 \boldsymbol{e}_3^\top\right) \mathbf{R}^\top \boldsymbol{v}_a, \tag{51}$$

and it was drawn from [45]; λ is a positive constant. The value of this constant for the 3DR® IRIS quadrotor is known, however, it has been tweaked in order to achieve a more similar flight performance to the INTEL® Aero Ready to Fly quadrotor (*vide* Table 1). The model presented for the drag force in Section 3.1 does not depend on the rotational speed of the propellers, though, as measurements of these quantities are unavailable both on INTEL® Aero Ready to Fly and 3DR® IRIS quadrotors.

To determine matrix **D**, introduced in Eq. 21, it is assumed that the multirotor is hovering. In this flight condition, $\mu \sum_{p=1}^{4} \omega_p^2 = mg$, and $\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2 = \omega_1^2 - \omega_3^2 = \omega_2^2 - \omega_4^2 = 0$. Thus,

$$\omega_p = \widetilde{\omega} := \sqrt{\frac{mg}{4\mu}}, \quad p = 1,\ 2,\ 3,\ 4,$$

with $\mu = \frac{\rho D^4}{4\pi^2} C_{T0}$. Given $\widetilde{\omega}$,

$$\mathbf{D} = 4 k_D \widetilde{\omega} \lambda \left(\mathbf{I}_3 - \boldsymbol{e}_3 \boldsymbol{e}_3^\top\right), \tag{52}$$

with k_D being a tunable parameter. Although considering that $\omega_p = \widetilde{\omega}$ is valid in nominal flight conditions, as the sum of the rotor speeds is almost constant, i.e. $\sum_{p=1}^{4} \omega_p \approx 4\widetilde{\omega}$, in aggressive flight the same is not necessarily verified. The parameter k_D allows to adjust the linear drag model Eq. 21 and to leverage its simplicity to seamlessly design tracking controllers. Through experimentation in the Gazebo simulator, it has been found that considering the constant $k_D = 19/30$ for the loop manoeuvre, and $k_D = 4/19$ for the relay manoeuvre to obtain the estimate of **D** enables quadrotors to reach the desired goal configuration at the end of *Phase 2* of the manoeuvres with small tracking error.

6.2 Performance Comparison with the State-of-the-art

In this section a quantitative comparison with the planning strategy presented in Mellinger et al. [23] is made. In said work, the aim is also to make a multirotor vehicle reach a goal state with precision. Trajectories are designed by first integrating the vehicle dynamics backwards in time, but, in contrast to this paper, neither the relation between snap and body torques nor the geometry of SO(3) are exploited. Different controllers are proposed to track each segment that form the trajectory and iteratively refined through various trials to account for errors in the dynamic model and noise in the actuators and sensors.

To conduct a fair performance comparison, the nine test cases considered in the aforementioned work to validate the strategy there proposed are carried out in this work as well. Each of these test cases is detailed in Table 2. For all of them, the vehicle is commanded to reach a goal position $\boldsymbol{p}_G = [2\ 2\ 5]^\top$m with zero roll and yaw angles. The multirotor executed each manoeuvre in MATLAB after completing a total of 20 trials to iteratively refine velocity and pitch angle commands. Although iterating on the pitch angle command only in the first two trials revealed to be enough, it was necessary to iterate on velocity more than the four times that were

Table 2 The nine test cases used in Mellinger et al. [23] to validate the proposed planning and control strategy

Case	Description	$\boldsymbol{v}_G$ (v_x, v_y, v_z) (m/s)
1	Vertical window at 45°	(2, 0, 0)
2	Vertical window at 60°	(2, 0, 0)
3	Vertical window at 75°	(2, 0, 0)
4	Vertical window at 90°	(2, 0, 0)
5	Down through horizontal window at 90°	$(0, 0, -1.5)$
6	Up through horizontal window at 90°	(0, 0.4, 2.2)
7	Perch at 60°	$(0, 0.8\cos(\pi/6), -0.8\sin(\pi/6))$
8	Perch at 90°	(0, 0.8, 0)
9	Perch at 120°	$(0, 0.8\cos(\pi/6), 0.8\sin(\pi/6))$

considered in [23]. To observe a satisfactory improvement of the position error, 20 iterations were necessary.

Table 3 reports the results obtained in MATLAB for: (a) the strategy proposed in this work and (b) the strategy presented in Mellinger et al. [23]. While strategy (b) enables multirotors to reach the desired velocity and orientation with high precision in the great majority of test cases (exceptions are cases 6 and 9), it exhibits a considerable position error – mainly in the x and z axes – in most of the simulations conducted. It has been observed, in particular for the most demanding manoeuvres, that the vehicle is requested to reach the *launch* configuration with a too large velocity, which has hindered it in reaching successfully the goal configuration. For instance, for five out the nine test cases the z error was larger than 20 cm. As for the strategy in this work (a), the results show the multirotor consistently reaches the desired position, velocity, and orientation with minimal error. For all nine test cases, i) the position error was at most 1 cm; ii) the velocity error was at most 0.01 m s^{-1}; and iii) the pitch angle error was at most 2.88°. Even though strategy (b) provides marginally better orientation results for six out of the nine test cases, the strategy here presented (a) in addition to enabling the vehicle to achieve the desired orientation with precision, the position and velocity errors are significantly smaller than those obtained with strategy (b).

6.3 Loop Manoeuvre

Swinging a multirotor into a loop is a challenging manoeuvre because high thrust and angular rates, with sudden changes in these quantities, may be required. The time duration of each phase of the manoeuvre (*Phase 1* – go to launch point; *Phase 2* – go to the goal configuration; *Phase 3* – recover

Table 3 Position, velocity, and orientation (parameterised using ZXY Euler Angles) at the goal point G: (a) the results obtained using the strategy proposed in this work; (b) the results obtained using the strategy proposed in [23]

Case		x (m)	y (m)	z (m)	v_x (m/s)	v_y (m/s)	v_z (m/s)	ϕ (deg)	θ (deg)	ψ (deg)
1	a	**2.00**	**2.00**	**5.00**	**2.01**	0.00	**0.00**	0.00	46.56	0.00
	b	1.88	2.00	5.02	2.02	0.00	0.01	0.00	**44.78**	0.00
2	a	**2.00**	**2.00**	**5.00**	2.01	0.00	0.00	0.00	61.78	0.00
	b	1.88	2.00	5.12	**2.00**	0.00	0.00	0.00	**59.81**	0.00
3	a	**2.00**	**2.00**	**5.00**	**2.01**	**0.00**	0.00	0.00	75.92	0.00
	b	1.92	2.00	5.19	2.05	–0.01	0.00	0.00	**74.70**	0.00
4	a	**2.00**	**2.00**	**5.00**	2.01	0.00	0.00	**0.00**	**90.02**	0.00
	b	1.92	2.00	5.25	1.99	0.00	0.00	–0.01	89.62	0.00
5	a	**1.99**	**2.00**	**5.00**	**0.01**	0.00	**–1.50**	0.00	92.15	**0.00**
	b	1.98	2.00	5.23	0.07	0.00	–1.57	0.00	**89.54**	–0.01
6	a	1.99	2.00	**5.00**	**0.01**	**0.40**	**2.21**	**–0.01**	87.21	0.81
	b	1.99	2.00	5.28	0.18	0.38	1.84	0.45	**89.27**	**0.15**
7	a	**2.00**	**2.00**	**5.00**	0.01	0.70	–0.40	–0.67	60.73	1.21
	b	1.92	1.98	5.17	–0.01	**0.69**	–0.40	**–0.05**	**59.75**	**0.07**
8	a	**1.99**	**2.00**	**5.00**	**0.01**	0.81	**0.00**	**0.00**	**90.19**	1.62
	b	1.96	1.99	5.23	–0.07	**0.80**	–0.01	0.80	89.48	**0.28**
9	a	**1.99**	**2.00**	**5.00**	**0.01**	**0.70**	**0.40**	**0.70**	**119.69**	1.27
	b	2.10	2.01	5.23	0.64	0.62	–0.70	3.09	119.19	**0.79**

and finish at hover) is depicted in Fig. 4 and the waypoints established for this manoeuvre are the following:

- The vehicle starts at position $\boldsymbol{p}_S = \begin{bmatrix}0 & 2.5 & 5\end{bmatrix}^\top$ m;
- To perform the loop, the vehicle moves towards the goal configuration G:
  - G is defined by $\boldsymbol{p}_G = \begin{bmatrix}1 & 2.5 & 25\end{bmatrix}^\top$ m, $\boldsymbol{v}_G = \begin{bmatrix}-2 & 0 & 0\end{bmatrix}^\top$ m s^{-1}, and $\mathbf{R}_G = \mathbf{R}_2(-\pi)$, a rotation of $-\pi$ radians about $\boldsymbol{e}_2$;
- At the end of the manoeuvre, the vehicle is commanded to hover at position $\boldsymbol{p}_F = \begin{bmatrix}4 & 2.5 & 17\end{bmatrix}^\top$ m.

The detailed results for one of the executed trials picked randomly are presented in Fig. 5, 6, and 7. Histograms to inspect the distribution of the position and orientation error relative to the goal configuration are presented in Fig. 8. In Fig. 5, the position reference trajectories of the vehicle together with the trajectory it described in the xz plane during *Phase 2* and *3* of the manoeuvre are presented. Along the described trajectories, drawings of the vehicles are included at specific time instants, taking into account their actual configuration during the simulation in Gazebo.

Figure 6 presents the position; linear velocity; the orientation parameterised by ZXY Euler angles, ϕ, θ, $\psi \in [-180, 180]$ deg, and the angular velocity plots of the vehicle during the entire loop manoeuvre. The evolution of these state variables observed in MATLAB and SITL simulations is included in the results. As the multirotor dynamics are differentially flat, from the reference trajectory in the space of flat outputs, the reference trajectories for all state variables and inputs can be algebraically obtained. The orientation reference trajectory plot was computed considering that there was no drag to examine whether the proposed controller can make the vehicle track the desired orientation reference despite being subject to this force. The position and orientation tracking errors of the rotorcraft relative to the optimal reference trajectory are shown in Fig. 7 for a closer inspection of the obtained results in the SITL simulation. These plots demonstrate that, in the simulation carried out in Gazebo, both vehicles exhibit satisfactory position tracking performance over the entire manoeuvre.

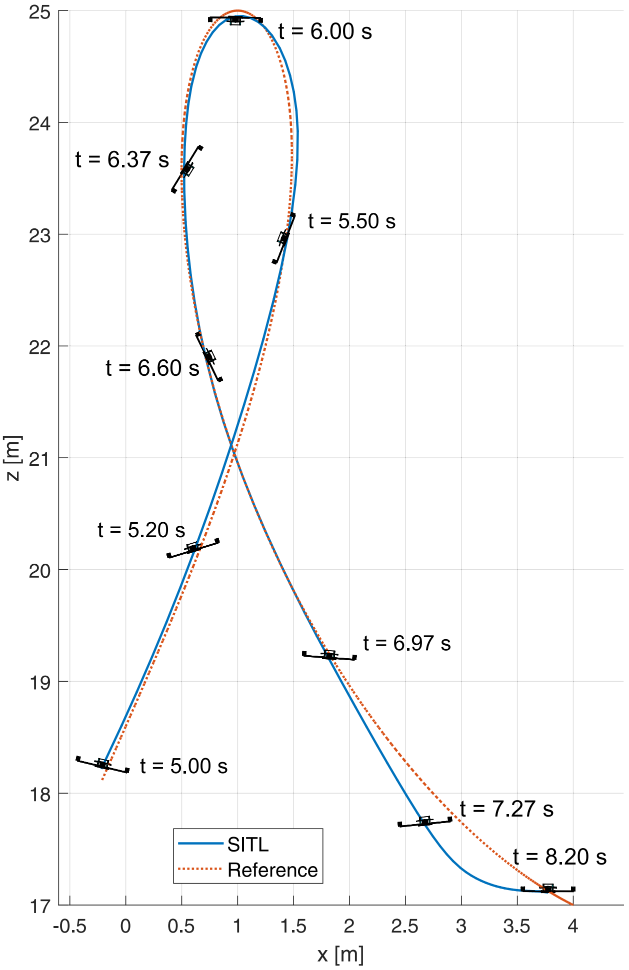


Fig. 5 Illustration of Phase 2 and 3 of the loop manoeuvre drawn using data observed during the software-in-the-loop (SITL) simulation

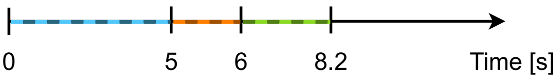
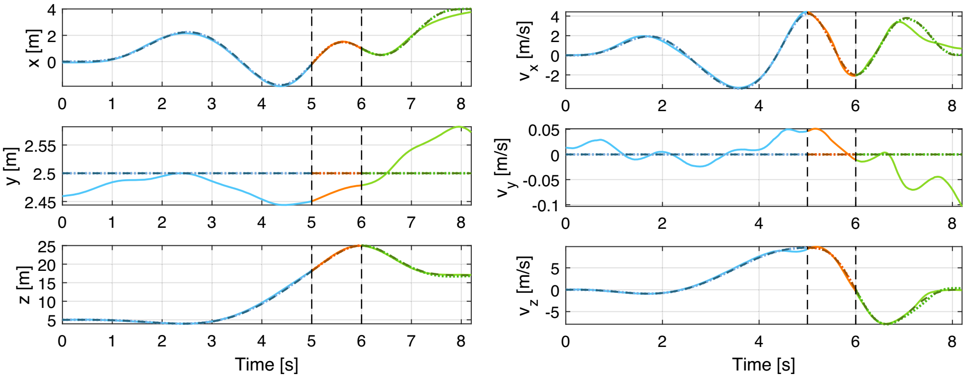


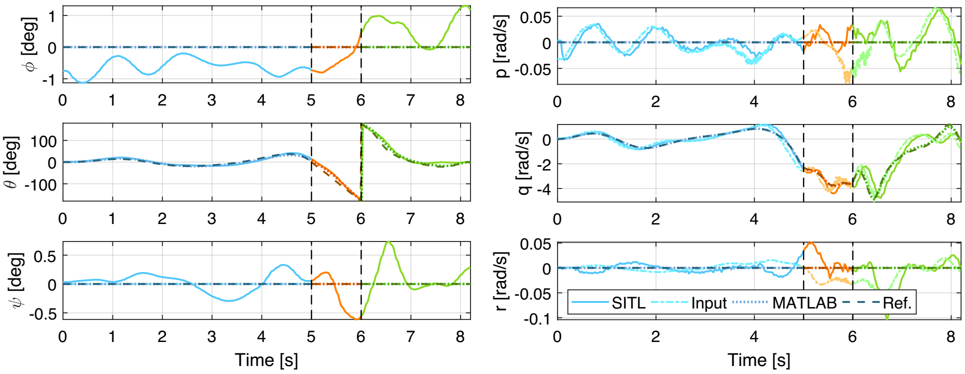
Fig. 4 Time axis representing the duration of the each phase of the loop manoeuvre

Further conclusions are drawn by examining the illustration in Fig. 5 and the position tracking error in Fig. 7a. The position tracking error, mainly in the x and z axes, is perceptible, as the high linear velocities in these axes that are required to launch the vehicle into the loop and to finish the manoeuvre at hover lead to a deterioration of the position tracking performance. For the SITL results, a slight oscillatory error behaviour is also visible – which is attributed to slight time synchronisation errors between reference and state values while exchanging messages through ROS – at higher travel velocities. It has been verified that there is a varying delay in the arrival times of the state messages that the process running the PX4 autopilot sends through ROS to the process running the controller. The analysis of the plots of Fig. 6b reveals that the vehicle reaches $\|\boldsymbol{v}\| = 10.3$ m s^{-1} at the end of *Phase 1*. In addition, it is seen that, during *Phase 3*, it visibly struggles to follow the velocity reference in x. Towards the end of *Phase 2*, however, the rotorcraft reaches the desired position reference with an RMSE = 3.85 cm. Finally, the sample distribution of the x and z components of the position error relative to goal configuration shown in Fig. 8a and b – that were obtained by performing the manoeuvre 1,000 times – demonstrate the vehicle reaches the desired position with precision and repeatability. The mean error and

Fig. 6 Position, linear velocity, orientation (expressed in ZXY Euler angles), and angular rates plots of the loop manoeuvre conducted in the Gazebo simulator and MATLAB

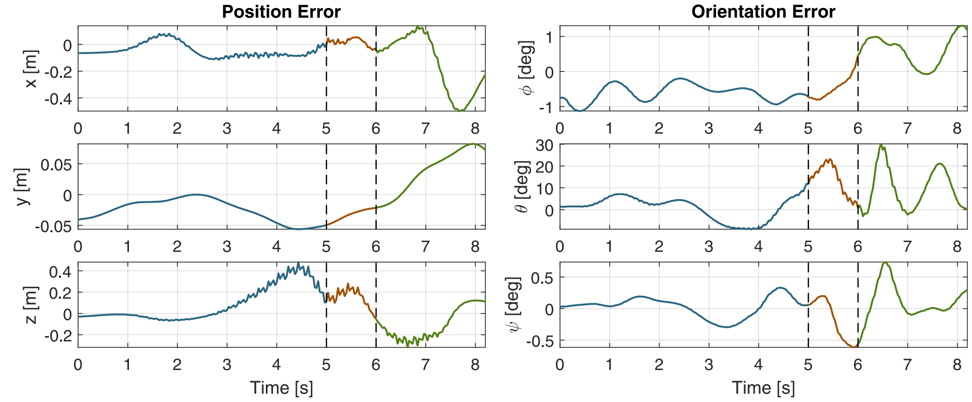


(a) Position. (b) Linear velocity.



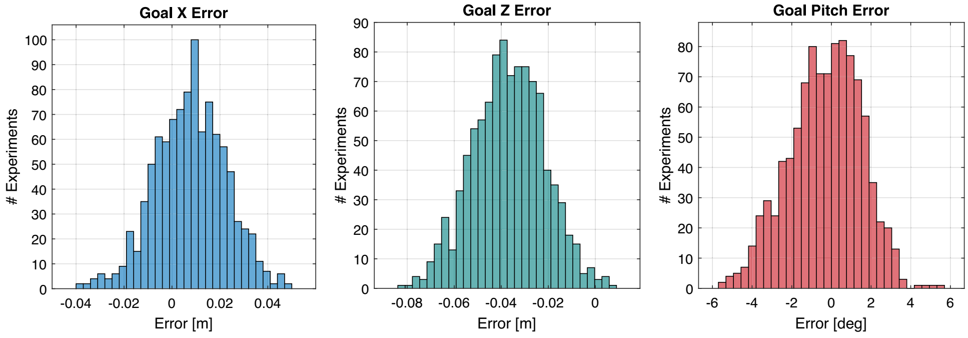
(c) Orientation. (d) Angular rates.

Fig. 7 Position and orientation (expressed in ZXY Euler angles) tracking errors during the loop manoeuvre conducted in the Gazebo simulator. These errors are relative to the optimal planned trajectories



(a) Position error. (b) Orientation error.

Fig. 8 Histograms of the position and pitch error at time $t = 6.00$ s relative to the desired goal configuration. The data shown was obtained by running the loop manoeuvre 1,000 times in the Gazebo simulator



(a) X. (b) Z. (c) Pitch.

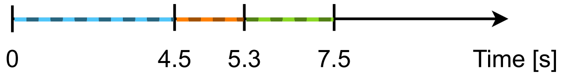


Fig. 9 Time axis representing the duration of the each phase of the parcel relay manoeuvre

standard deviation in x and z are, respectively, $\mu_x = 0.78$ cm and $\sigma_x = 1.47$ cm, and $\mu_z = -3.69$ cm and $\sigma_z = 1.52$ cm.

As for the attitude tracking performance, the angular velocity inputs to the autopilot, generally, follow closely the reference trajectory. Yet, almost halfway into *Phase 2* and during *Phase 3*, the orientation error of each vehicle relative to the reference trajectory is considerable, since the controller is aiming to drive the position tracking error to zero. For the randomly picked trial, the orientation of vehicle at time $t = 6.00$ s is parameterised by $\phi = 0.46$ deg, $\theta = -179.2$ deg, and $\psi = -0.56$ deg. The histogram shown in Fig. 8c reveals that the pitch error at $t = 6.00$ s follows approximately a normal distribution parameterised by the mean $\mu_\theta = -0.32°$ and standard deviation $\sigma_\theta = 1.81°$.

The tracking performance in the MATLAB simulation is fairly good, since the vehicle has access to its actual state, unlike SITL simulations where only a state estimate, obtained by fusing synthetic on-board sensors and motion capture system data, is available. Nevertheless, given that drag is also considered in the MATLAB simulations, the tracking performance is not perfect, as a slight tracking error is observed across all plots depicted in Fig. 6.

6.4 Parcel Relay Manoeuvre

To perform relay manoeuvres, a trajectory is generated for each of the vehicles involved using the methods described in Section 4. To ensure that the vehicles exchanging a parcel fly side-by-side, a simple synchronisation software is responsible for guaranteeing that the rotorcraft begin performing the manoeuvre simultaneously. After taking off, each of the vehicles is commanded to reach a certain position waypoint before following the trajectory. Whenever one of the multirotors reaches its own setpoint, it periodically ascertains whether the other one has reached its own as well. Once a positive response is obtained, the vehicle which posed the question requests the other vehicle to start tracking the trajectory together.

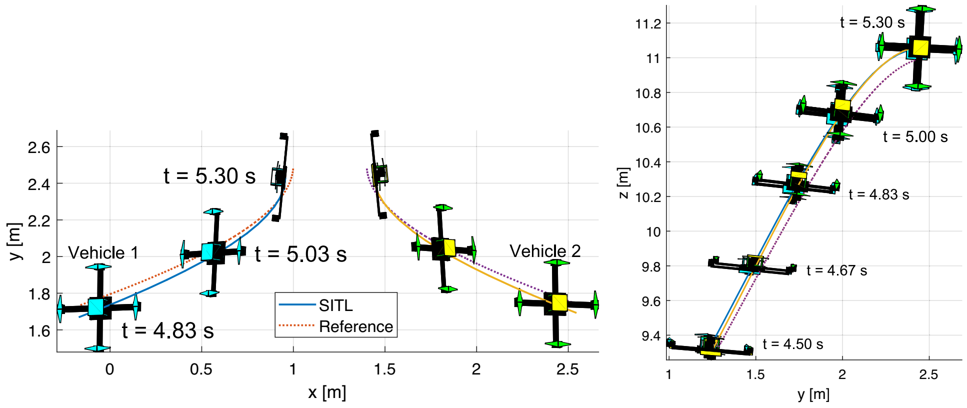
The proposed strategy to plan the relay manoeuvres is validated by analysing the simulation (Section 6.4.1) and experimental results (Section 6.4.2). Real-world experiments were carried out with only one INTEL® Aero Ready to Fly quadrotor for the dimensions of the indoor flying arena (7.0 × 4.0 × 2.5 m). Given the aim of the present paper, in the experimental setup, greater importance was given to performing manoeuvres that require large excursions from the hover condition and fast attitude changes.

6.4.1 Simulation Results

The time duration of each phase of the manoeuvre is depicted in Fig. 9 and the waypoints established for this manoeuvre are the following:

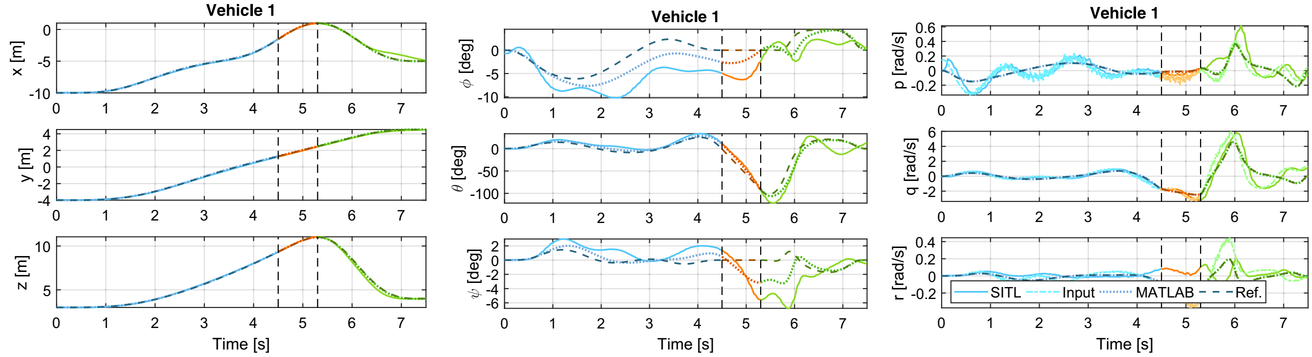
- Vehicle 1 starts at position $\boldsymbol{p}_{S1} = \begin{bmatrix}-10 & -4 & 3\end{bmatrix}^\top$ m; Vehicle 2, at position $\boldsymbol{p}_{S2} = \begin{bmatrix}12.4 & -4 & 3\end{bmatrix}^\top$ m;
- To exchange a parcel between the two rotorcraft, Vehicle 1 moves towards the goal configuration G_1, and Vehicle 2, towards the goal configuration G_2:
  - G_1 is defined by $\boldsymbol{p}_{G1} = \begin{bmatrix}1 & 2.5 & 11\end{bmatrix}^\top$ m, $\boldsymbol{v}_{G1} = \begin{bmatrix}0 & 1.5 & 0\end{bmatrix}^\top$ m s^{-1}, and $\mathbf{R}_{G1} = \mathbf{R}_2(-\pi/2)$;
  - G_2 is defined by $\boldsymbol{p}_{G2} = \begin{bmatrix}1.4 & 2.5 & 11\end{bmatrix}^\top$ m, $\boldsymbol{v}_{G2} = \begin{bmatrix}0 & 1.5 & 0\end{bmatrix}^\top$ m s^{-1}, and $\mathbf{R}_{G2} = \mathbf{R}_2(\pi/2)$;
- At the end of the manoeuvre, both vehicles are commanded to hover. Vehicle 1 at position $\boldsymbol{p}_{F1} = \begin{bmatrix}-5 & 4.5 & 4\end{bmatrix}^\top$ m, Vehicle 2 at position $\boldsymbol{p}_{F2} = \begin{bmatrix}7.4 & 4.5 & 4\end{bmatrix}^\top$ m.

Fig. 10 Illustration of Phase 2 of the parcel relay manoeuvre drawn using data observed during the software-in-the-loop (SITL) simulation. Note that the drawings of the vehicles are not to scale. The reference trajectory of Vehicle 1 is depicted in red 🟥, whereas the trajectory this vehicle described, in blue 🟦; the reference trajectory of Vehicle 2 is depicted in purple 🟪, the trajectory it described, in yellow 🟨

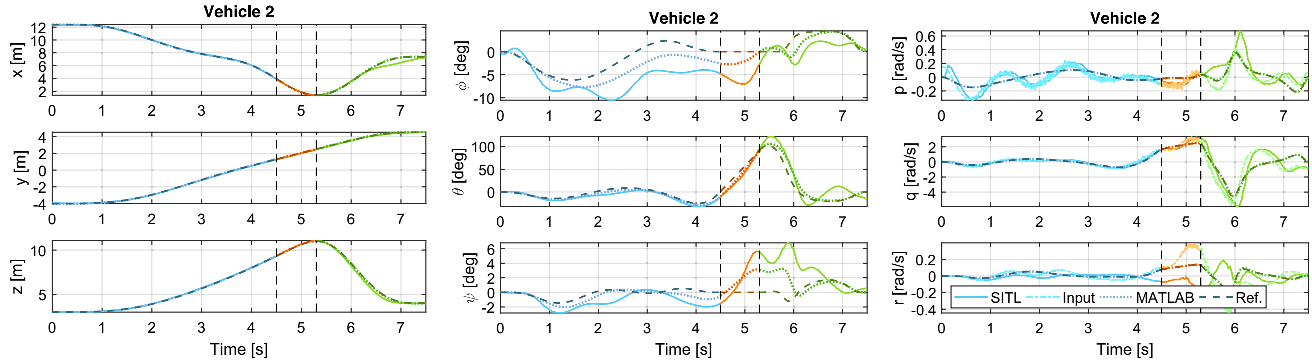


(a) xy plane view.

(b) yz plane view.



(a) Position of Vehicle 1. (b) Orientation of Vehicle 1. (c) Angular velocity of Vehicle 1.

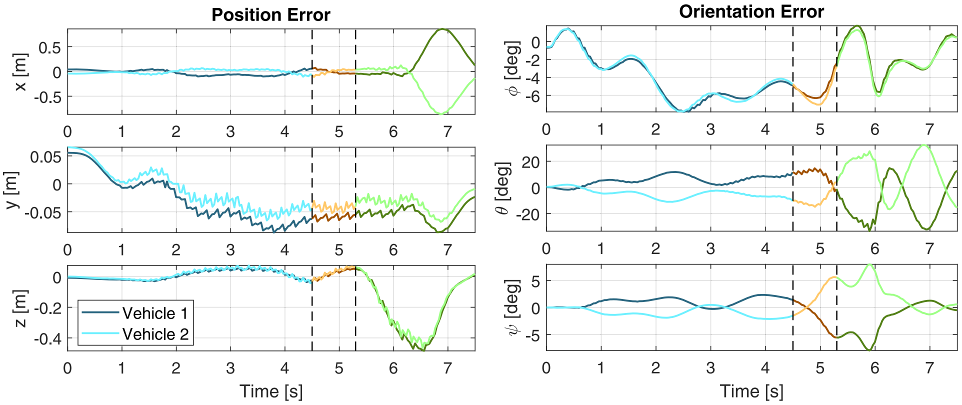


(d) Position of Vehicle 2. (e) Orientation of Vehicle 2. (f) Angular velocity of Vehicle 2.

Fig. 11 Position, orientation (expressed in ZXY Euler angles), and angular velocity plots of the parcel relay manoeuvre conducted in the Gazebo simulator and MATLAB

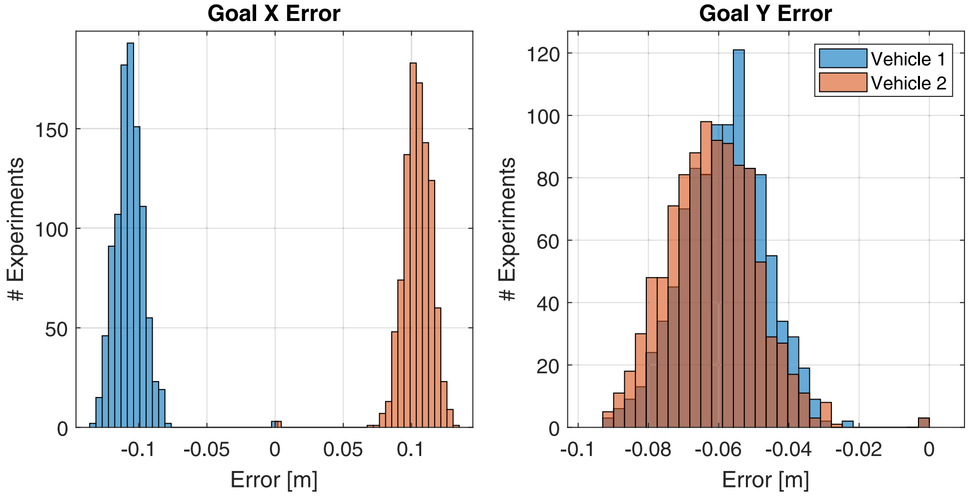
The detailed results for one of the executed trials picked randomly are presented in Fig. 10, 11, and 12. Histograms to inspect the distribution of the position and orientation error relative to the goal configuration are presented in Fig. 13 and 14. In Fig. 10, the position reference trajectories of both vehicles together with the described trajectories during the *Phase 2* of this manoeuvre are presented. Top and lateral views of the manoeuvre are depicted. Along the described trajectories, drawings of the vehicles are included at specific time instants, taking into account their actual configuration during the experiment in the Gazebo simulator.

Figure 11 presents the position, orientation, and angular velocity plots of both vehicles during the entire relay manoeuvre. The position and orientation tracking errors of the rotorcraft relative to the optimal reference trajectory are plotted in Fig. 12. As in the previous section, the orientation reference trajectory plots were computed considering that there was no drag to verify whether the proposed tracking

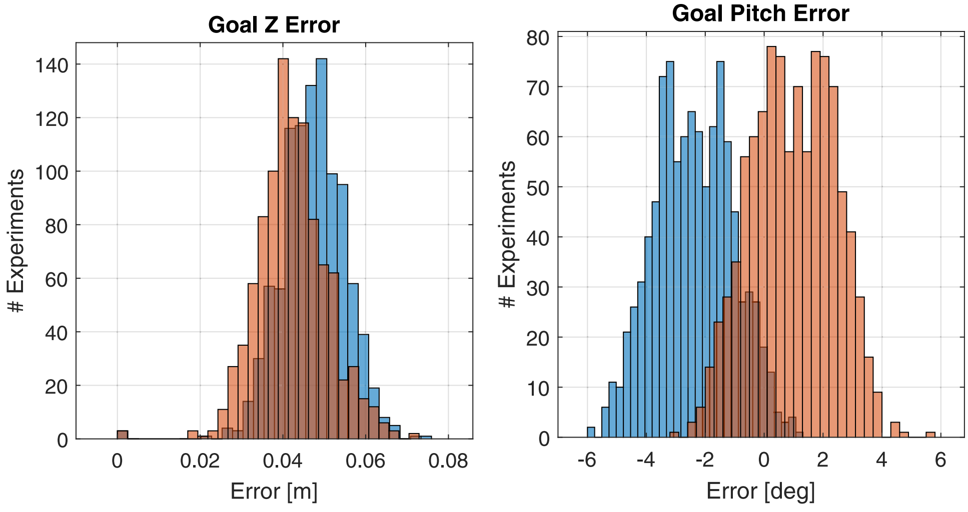


(a) Position error of the two vehicles. (b) Orientation error of the two vehicles.

Fig. 12 Position and orientation (expressed in ZXY Euler angles) tracking errors during the parcel relay manoeuvre conducted in the Gazebo simulator. These errors are relative to the optimal trajectories. The tracking error evolution across time of Vehicle 1 is depicted using dark colours, whereas the errors evolution of Vehicle 2, using lighter shades of those same colours

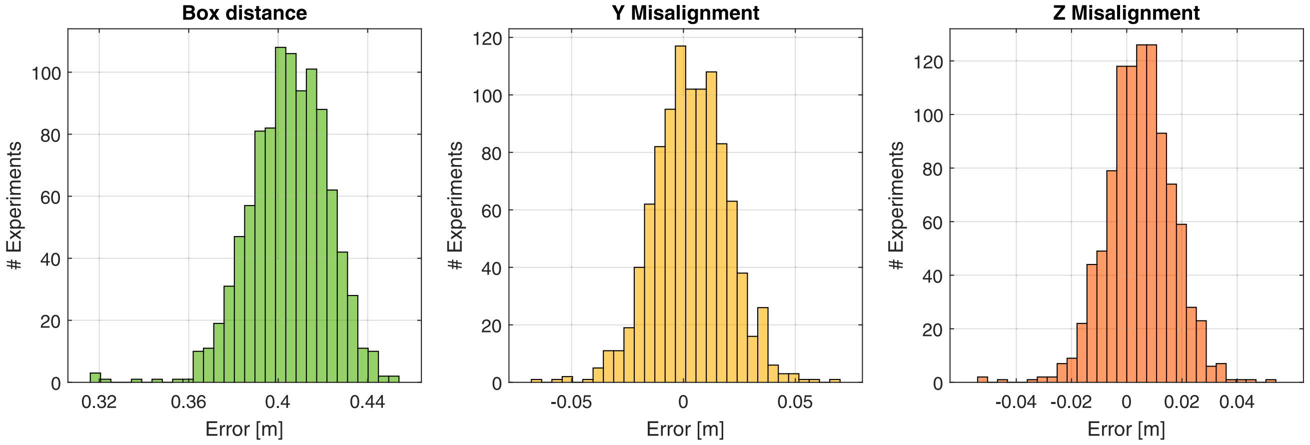


(a) X. (b) Y.



(c) Z. (d) Pitch.

Fig. 13 Histograms of the position and pitch error at time $t = 5.30\,\mathrm{s}$ relative to the desired goal configuration. The data shown was obtained by running the relay manoeuvre 1,000 times in the Gazebo simulator



(a) Box distance. (b) Misalignment in Y. (c) Misalignment in Z.

Fig. 14 Histograms of the box distance and misalignment between the two multirotors at time $t = 5.30\,\mathrm{s}$. The data shown was obtained by running the relay manoeuvre 1,000 times in the Gazebo simulator

Table 4 Mean μ and standard deviation σ that parametrise the approximate distribution of the position and pitch errors relative to the goal configuration at time $t = 5.30$ s

Vehicle 1		Vehicle 2	
$\mu_x = -10.7\,\text{cm}$	$\sigma_x = 0.97\,\text{cm}$	$\mu_x = 10.4\,\text{cm}$	$\sigma_x = 1.11\,\text{cm}$
$\mu_y = -5.84\,\text{cm}$	$\sigma_y = 1.16\,\text{cm}$	$\mu_y = -6.17\,\text{cm}$	$\sigma_y = 1.26\,\text{cm}$
$\mu_z = 4.74\,\text{cm}$	$\sigma_z = 0.74\,\text{cm}$	$\mu_z = 4.24\,\text{cm}$	$\sigma_z = 0.85\,\text{cm}$
$\mu_\theta = -2.40^\circ$	$\sigma_\theta = 1.31^\circ$	$\mu_\theta = 0.97^\circ$	$\sigma_\theta = 1.41^\circ$

controller can make the vehicle describe the desired manoeuvre despite being subject to this force.

The plots in the aforementioned figures demonstrate that the manoeuvre is successfully executed. Vehicle 1 reaches the goal configuration G_1 with a position RMSE of 4.96 cm and with orientation parameterised by $\phi_1 = -2.48$ deg, $\theta_1 = -91.2$ deg, $\psi_1 = -5.55$ deg; Vehicle 2 reaches the goal configuration G_2 with a position RMSE of 4.73 cm and with orientation parameterised by $\phi_2 = -2.79$ deg, $\theta_2 = 93.9$ deg, $\psi_2 = 5.55$ deg. The histograms depicted in Fig. 13 presenting the sample distribution of the position and pitch errors at time $t = 5.30$ s and the mean error and standard deviation computed over the dataset of 1,000 simulations reported in Table 4 allow to state – with 95% confidence – that the position and pitch errors of Vehicle 1 are $e_x^1 \in [-12.64, -8.76]$ cm, $e_y^1 \in [-8.16, -3.52]$ cm, $e_z^1 \in [3.26, 6.22]$ cm, and $e_\theta^1 \in [-5.02, 0.22]$ deg. As for the position and pitch errors of Vehicle 2, $e_x^2 \in [8.18, 12.62]$ cm, $e_y^2 \in [-8.69, -3.65]$ cm, $e_z^2 \in [2.54, 5.94]$ cm, and $e_\theta^2 \in [-1.85, 3.79]$ deg. The position error in x is noticeable, though, for the pitch error not converging to zero fast enough. To solve this issue, the desired position the vehicles should reach may be adjusted.

Furthermore, Fig. 12a reveals that during *Phase 3* the position tracking error, mainly in the x and z axes, is significant. It has been verified that the necessary body torques to transition from the goal to the final configuration are excessive, since it is demanded that rotorcraft finish performing the manoeuvre at hover. The vehicles would naturally keep rotating in the same direction, i.e. go past $|\theta| = 90$ deg, but they are forced to counteract that motion. Thus, the deterioration of the attitude tracking performance during *Phase 3* results in the observed position tracking error.

It is worth highlighting that the misalignment between the position of the vehicles is small. At the goal configuration, at time $t = 5.30$ s, the vehicles were off by 1.44 cm in the y axis and by 1.00 cm in the z axis. Considering boxes, with dimensions 50 × 50 × 15 cm, aligned with the body frame of each vehicle, the Euclidean distance between the two at the goal configuration is 34.4 cm. The sample distribution of these three quantities is presented in Fig. 14 and they reveal that the quantities in question follow approximately a normal distribution with the parameters reported in Table 5. The small mean and standard deviation for the misalignment in y and z provides further evidence to suggest that a package exchange at the goal configuration between the vehicles is possible.

Table 5 Mean μ and standard deviation σ that parametrise the approximate distribution of the box distance and the misalignment in y and z between the two vehicles at time $t = 5.30$ s

	μ (cm)	σ (cm)
Box distance	40.5	1.76
Y misalignment	0.35	1.64
Z misalignment	0.48	1.18

6.4.2 Experimental Results

The time duration of each phase of the manoeuvre is depicted in Fig. 15 and the waypoints established for this manoeuvre are the following:

- The vehicle starts at position $\boldsymbol{p}_S = \begin{bmatrix}-2.4 & -1.3 & 1.3\end{bmatrix}^\top$ m;
- To perform the loop, the vehicle moves towards the goal configuration G:
  - G is defined by $\boldsymbol{p}_G = \begin{bmatrix}1.2 & 1.9 & 2\end{bmatrix}^\top$ m, $\boldsymbol{v}_G = \begin{bmatrix}1.5 & 0.5 & 0\end{bmatrix}^\top$ m s^{-1}, and $\mathbf{R}_G = \mathbf{R}_1(7\pi/36)$, a rotation of $7\pi/36$ radians (35 deg) about $\boldsymbol{e}_1$;
- At the end of the manoeuvre, the vehicle is commanded to hover at position $\boldsymbol{p}_F = \begin{bmatrix}2.7 & -1.2 & 1.8\end{bmatrix}^\top$ m.

In Fig. 16, the position reference trajectory together with the described trajectory during the *Phase 2* of this manoeuvre is presented. Top and lateral views of the manoeuvre are depicted. Along the described trajectory, drawings of the vehicle are included at specific time instants, taking into account its actual configuration during the experiment.

Figure 17 presents the position, orientation, and angular velocity plots of the rotorcraft during the entire manoeuvre. The orientation reference trajectory plot was computed con-

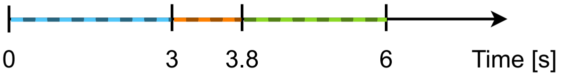
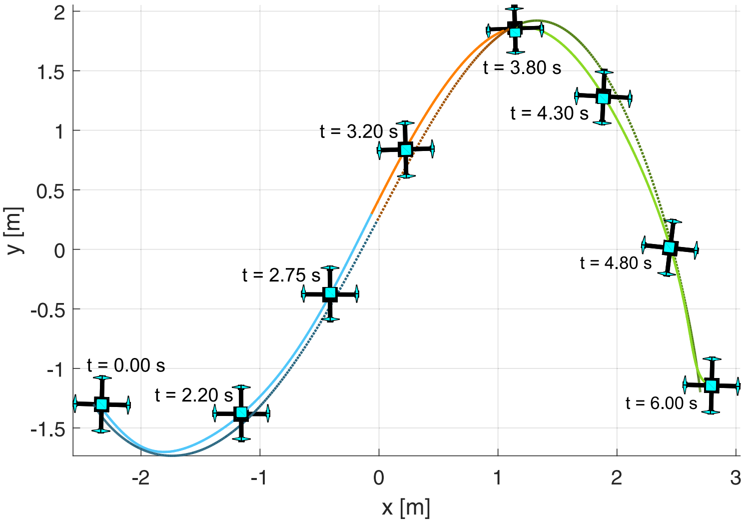
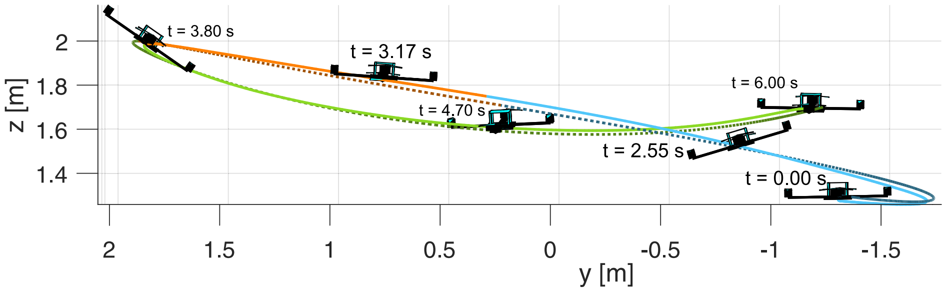


Fig. 15 Time axis representing the duration of the each phase of the experimental parcel relay manoeuvre



(a) xy plane view.



(b) yz plane view.

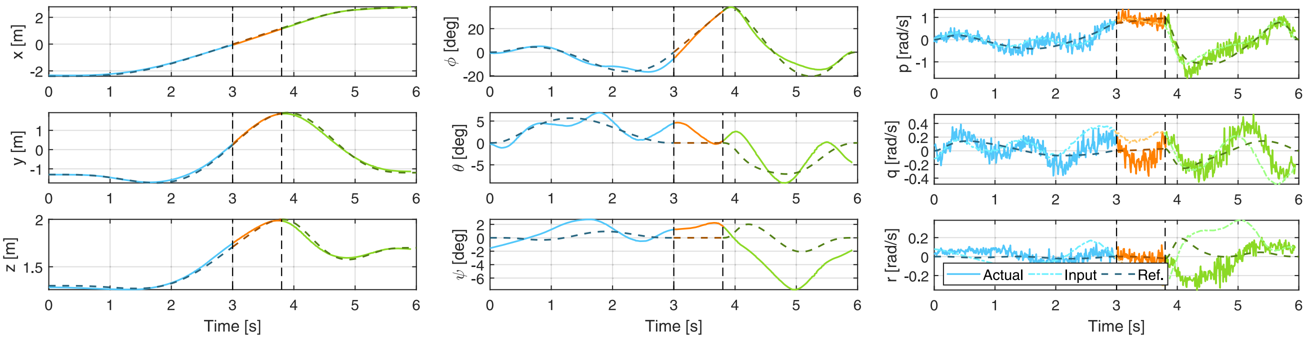
Fig. 16 Illustration of the parcel relay manoeuvre drawn using data observed during the experiment. Note that the drawings of the vehicle are not to scale. The planned (reference) trajectories are depicted using dashed lines and dark colours; the actual trajectories described by the vehicle, using continuous lines and lighter shades of those same colours

sidering that there was no drag to analyse whether the vehicle performs the manoeuvre accurately, though being subject to this force.

The plots in the aforementioned figures demonstrate that the manoeuvre is successfully executed. The position tracking error over the entire manoeuvre, in absolute value, reaches a maximum of 9.06 cm in the x axis, 16.7 cm in the y axis, and 5.07 cm in the z axis. The goal configuration G is reached with a position RMSE of 3.44 cm and with orientation parameterised by $\phi = 35.3$ deg, $\theta = 0.59$ deg, $\psi = 1.61$ deg.

7 Conclusion

This paper has presented a trajectory planning strategy that enables multirotors to achieve a desired configuration in state space. Minimum snap trajectories are generated from a



(a) Position of Vehicle 1. (b) Orientation of Vehicle 1. (c) Angular velocity of Vehicle 1.

Fig. 17 Position, orientation (expressed in ZXY Euler angles), and angular velocity plots of the experimental parcel relay manoeuvre quadratic program that includes linear constraints on the linear acceleration to coordinate the orientation of the vehicles. To design aggressive, precise manoeuvres, three different phases are considered. In the first phase, the vehicles fly from an initial to a launch configuration – to achieve the required momenta to reach the desired configuration in state space in the second phase of the manoeuvre. The vehicles are then driven to a final configuration in the third phase. So that the reference trajectories for the second phase are accurately tracked, the dynamical model of the multirotors is considered to design them by numerically integrating it backwards in time. By making the attitude subsystem of multirotors describe geodesic curves on SO(3) during the backwards integration, the relation between snap and body torques is exploited, as it enables planning minimum snap trajectories to the desired configuration that require small body torques to be tracked with accuracy and precision.

The SITL simulation results reveal a quadrotor successfully doing a loop-the-loop and two quadrotors performing a challenging manoeuvre that involves them being in close interaction and varying their orientation significantly, with the objective of exchanging a parcel during flight. Real-world experiments further demonstrate the suitability of the proposed trajectory planning strategy to perform aggressive drone manoeuvres.

Although considering geodesic curves on SO(3) as reference trajectories during the backwards integration allows low torque actuation to move to the goal configuration, for manoeuvres that involve large excursions from the hover condition, large linear and angular momenta are often necessary at the beginning of the second phase of the manoeuvre, at the *launch* configuration. A relevant line of research to follow would, therefore, be examining whether considering alternative curves on SO(3) or the special Euclidean group in three dimensions $\mathrm{SE}(3) := \mathrm{SO}(3) \times \mathbb{R}^3$ as reference trajectories for the backwards integration enables planning aggressive manoeuvres that require smaller momenta at the beginning of the second phase.

Acknowledgements The authors acknowledge José Tojeira for his valuable assistance in carrying out experimental tests.

Author Contributions All authors contributed to the study conception, manoeuvres design, and analysis. Coding and experiments were performed by João Pinto. The first draft of the manuscript was written by João Pinto and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding The work of João Pinto was supported by the PhD Grant 2022.12145.BD from Fundação para a Ciência e a Tecnologia (FCT), Portugal.
This work was supported by FCT through LARSyS funding (DOI: 10.54499/LA/P/0083/2020, 10.54499/UIDP/50009/2020, and 10.54499/UIDB/50009/2020), CTS funding (DOI: 10.54499/UIDP/00066/2020 and 10.54499/UIDB/00066/2020), and project CAPTURE (DOI: 10.54499/PTDC/EEI-AUT/1732/2020).

Code Availability The software environment developed to implement tracking controllers for multirotors is available at an online repository: https://github.com/dsor-isr/Quad_Ctrl_MAVROS.

Declarations

Competing Interests The authors have no relevant financial or non-financial interests to disclose.

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

References

1. LaValle, S.M.: Planning algorithms. Cambridge University Press, Cambridge, Cambridgeshire (2006). https://doi.org/10.1017/CBO9780511546877
2. Mellinger, D., Kumar, V.: Minimum snap trajectory generation and control for quadrotors. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 2520–2525. IEEE, (2011). https://doi.org/10.1109/ICRA.2011.5980409
3. Loianno, G., Brunner, C., McGrath, G., Kumar, V.: Estimation, control, and planning for aggressive flight with a small quadrotor with a single camera and IMU. IEEE Robot. Autom. Lett. **2**(2), 404–411 (2016). https://doi.org/10.1109/LRA.2016.2633290
4. Richter, C., Bry, A., Roy, N.: In: Inaba, M., Corke, P. (eds.) Polynomial Trajectory Planning for Aggressive Quadrotor Flight in Dense Indoor Environments, pp. 649–666. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-28872-7_37
5. Thomas, J., Pope, M., Loianno, G., Hawkes, E.W., Estrada, M.A., Jiang, H., Cutkosky, M.R., Kumar, V.: Aggressive flight with quadrotors for perching on inclined surfaces. J. Mech. Robot. **8**(5), 051007 (2016). https://doi.org/10.1115/1.4032250
6. Mao, J., Nogar, S., Kroninger, C.M., Loianno, G.: Robust active visual perching with quadrotors on inclined surfaces. IEEE Trans. Robot. **39**(3), 1836–1852 (2023). https://doi.org/10.1109/TRO.2023.3238911
7. Watterson, M., Liu, S., Sun, K., Smith, T., Kumar, V.: Trajectory optimization on manifolds with applications to quadrotor systems. Int. J. Robot. Res. **39**(2–3), 303–320 (2020). https://doi.org/10.1177/0278364919891775
8. Foehn, P., Romero, A., Scaramuzza, D.: Time-optimal planning for quadrotor waypoint flight. Sci. Robot. **6**(56), 1221 (2021). https://doi.org/10.1126/scirobotics.abh1221

9. Ryou, G., Tal, E., Karaman, S.: Multi-fidelity black-box optimization for time-optimal quadrotor maneuvers. Int. J. Robot. Res. **40**(12–14), 1352–1369 (2021). https://doi.org/10.1177/02783649211033317
10. Liu, Z., Cai, L.: Simultaneous planning and execution for quadrotors flying through a narrow gap under disturbance. IEEE Trans. Control Syst. Technol. **31**(6), 2644–2659 (2023). https://doi.org/10.1109/TCST.2023.3283446
11. Mueller, M.W., Hehn, M., D'Andrea, R.: A computationally efficient motion primitive for quadrocopter trajectory generation. IEEE Trans. Robot. **31**(6), 1294–1310 (2015). https://doi.org/10.1109/TRO.2015.2479878
12. Hehn, M., D'Andrea, R.: Real-time trajectory generation for quadrocopters. IEEE Trans. Robot. **31**(4), 877–892 (2015). https://doi.org/10.1109/TRO.2015.2432611
13. Liu, S., Atanasov, N., Mohta, K., Kumar, V.: Search-based motion planning for quadrotors using linear quadratic minimum time control. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2872–2879. IEEE, (2017). https://doi.org/10.1109/IROS.2017.8206119
14. Liu, S., Mohta, K., Atanasov, N., Kumar, V.: Search-based motion planning for aggressive flight in SE(3). IEEE Robot. Autom. Lett. **3**(3), 2439–2446 (2018). https://doi.org/10.1109/LRA.2018.2795654
15. Li, Y., Littlefield, Z., Bekris, K.E.: Asymptotically optimal sampling-based kinodynamic planning. Int. J. Robot. Res. **35**(5), 528–564 (2016). https://doi.org/10.1177/0278364915614386
16. Tang, Z., Chen, B., Lan, R., Li, S.: Vector field guided RRT* based on motion primitives for quadrotor kinodynamic planning. J. Intell. Robot. Syst. **100**, 1325–1339 (2020). https://doi.org/10.1007/s10846-020-01231-y
17. Ye, H., Pan, N., Wang, Q., Xu, C., Gao, F.: Efficient sampling-based multirotors kinodynamic planning with fast regional optimization and post refining. In: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3356–3363. IEEE, (2022). https://doi.org/10.1109/IROS47612.2022.9981707
18. Penicka, R., Scaramuzza, D.: Minimum-time quadrotor waypoint flight in cluttered environments. IEEE Robot. Autom. Lett. **7**(2), 5719–5726 (2022). https://doi.org/10.1109/LRA.2022.3154013
19. Kamel, M., Alexis, K., Achtelik, M., Siegwart, R.: Fast nonlinear model predictive control for multicopter attitude tracking on SO(3). In: 2015 IEEE Conference on Control Applications (CCA), pp. 1160–1166. IEEE, (2015). https://doi.org/10.1109/CCA.2015.7320769
20. Pereira, J.C., Leite, V.J., Raffo, G.V.: Nonlinear model predictive control on SE(3) for quadrotor aggressive maneuvers. J. Intell. Robot. Syst. **101**, 1–15 (2021). https://doi.org/10.1007/s10846-021-01310-8
21. Mayne, D.: An apologia for stabilising terminal conditions in model predictive control. Int. J. Control **86**(11), 2090–2095 (2013). https://doi.org/10.1080/00207179.2013.813647
22. Mayne, D.Q.: Model predictive control: Recent developments and future promise. Automatica **50**(12), 2967–2986 (2014). https://doi.org/10.1016/j.automatica.2014.10.128
23. Mellinger, D., Michael, N., Kumar, V.: In: Khatib, O., Kumar, V., Sukhatme, G. (eds.) Trajectory Generation and Control for Precise Aggressive Maneuvers with Quadrotors, pp. 361–373. Springer, Berlin, Heidelberg (2014). https://doi.org/10.1007/978-3-642-28572-1_25
24. Wang, L., Su, J.: Switching control of attitude tracking on a quadrotor UAV for large-angle rotational maneuvers. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 2907–2912. IEEE, (2014). https://doi.org/10.1109/ICRA.2014.6907277
25. Chaturvedi, N.A., Sanyal, A.K., McClamroch, N.H.: Rigid-body attitude control. IEEE Contr. Syst. Mag. **31**(3), 30–51 (2011). https://doi.org/10.1109/MCS.2011.940459
26. Haier, E., Lubich, C., Wanner, G.: Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations, 2nd edn. Springer, Berlin, Heidelberg (2006). https://doi.org/10.1007/3-540-30666-8
27. Iserles, A., Munthe-Kaas, H.Z., Nørsett, S.P., Zanna, A.: Lie-group methods. Acta Numerica **9**, 215–365 (2000). https://doi.org/10.1017/S0962492900002154
28. Pinto, J., Guerreiro, B.J., Cunha, R.: Planning parcel relay manoeuvres for quadrotors. In: 2021 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 137–145. IEEE, (2021). https://doi.org/10.1109/ICUAS51884.2021.9476757
29. Meier, L., Honegger, D., Pollefeys, M.: PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 6235–6240. IEEE, (2015). https://doi.org/10.1109/ICRA.2015.7140074
30. O'Neill, B.: Semi-Riemannian Geometry with Applications to Relativity, vol. 103. Academic Press, New York, NY (1983)
31. Do Carmo, M.P.: Riemannian Geometry, vol. 6. Birkhäuser, Boston, MA (1992)
32. Lee, J.M.: Introduction to smooth manifolds, 2nd edn. Graduate Texts in Mathematics. Springer, New York, NY (2013). https://doi.org/10.1007/978-1-4419-9982-5
33. Murray, R.M., Li, Z., Sastry, S.S.: A Mathematical Introduction to Robotic Manipulation. CRC Press, Boca Raton, FL (1994)
34. Olver, P.J.: Applications of Lie Groups to Differential Equations, 2nd edn. Springer, New York (1993). https://doi.org/10.1007/978-1-4612-4350-2
35. Crouch, P.E., Grossman, R.: Numerical integration of ordinary differential equations on manifolds. J. Nonlinear Sci. **3**(1), 1–33 (1993). https://doi.org/10.1007/BF02429858
36. Jackiewicz, Z., Marthinsen, A., Owren, B.: Construction of Runge-Kutta methods of Crouch-Grossman type of high order. Adv. Comput. Math. **13**(4), 405–415 (2000). https://doi.org/10.1023/A:1016645730465
37. Kai, J.-M., Allibert, G., Hua, M.-D., Hamel, T.: Nonlinear feedback control of quadrotors exploiting first-order drag effects. IFAC-PapersOnLine. **50**(1), 8189–8195 (2017). https://doi.org/10.1016/j.ifacol.2017.08.1267. 20th IFAC World Congress
38. Atkinson, K.E.: An Introduction to Numerical Analysis, 2nd edn. John Wiley & Sons, Nashville, TN (1989)
39. Faessler, M., Franchi, A., Scaramuzza, D.: Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories. IEEE Robot. Autom. Lett. **3**(2), 620–626 (2017). https://doi.org/10.1109/LRA.2017.2776353
40. Lee, T., Leok, M., McClamroch, N.H.: Geometric tracking control of a quadrotor UAV on SE(3). In: 49th IEEE Conference on Decision and Control (CDC), pp. 5420–5425. IEEE, (2010). https://doi.org/10.1109/CDC.2010.5717652
41. Open Robotics: Gazebo simulator. [Online] Available: https://gazebosim.org/. [Online; accessed 15-Apr-2023] (2023)
42. OptiTrack: Motion Capture Systems. https://optitrack.com. [Online; Accessed 15 Apr 2023] (2023)
43. McCormick, B.W.: Aerodynamics, Aeronautics, and Flight Mechanics, 2nd edn. John Wiley & Sons, New York, NY (1994)
44. PX4 Autopilot: PX4 SITL Gazebo-classic. [Online] Available: https://github.com/PX4/PX4-SITL_gazebo-classic/. [Online; Accessed 15-Nov-2023] (2023)
45. Martin, P., Salaün, E.: The true role of accelerometer feedback in quadrotor control. In: 2010 IEEE International Conference on Robotics and Automation (ICRA), pp. 1623–1629. IEEE, (2010). https://doi.org/10.1109/ROBOT.2010.5509980

João Pinto received the B.Sc. and M.Sc. degrees in Electrical and Computer Engineering from Instituto Superior Técnico, Universidade de Lisboa, Portugal, in 2018 and 2021, respectively, where he is currently pursuing the Ph.D. degree in the same field. His current research interests include trajectory planning and tracking, model predictive control, multi-agent systems, and controller design for autonomous aerial vehicles.

Bruno J. Guerreiro received the Licenciatura and Ph.D. degrees in electrical and computer engineering from Instituto Superior Técnico, Lisbon, Portugal, in 2004 and 2013, respectively. He is currently an Assistant Professor with the NOVA School of Science and Technology, NOVA University Lisbon (FCT/UNL), and also a Researcher with CTS/Uninova/LASI and ISR-Lisbon/LARSyS. His research interests include theoretical, practical, and technological challenges related to the use of aerospace robotics for civilian applications, with particular emphasis on sensor-based SLAM, model-based predictive control, trajectory planning and tracking, linear and nonlinear control, automatic LIDAR calibration, and modelling and instrumentation of aerospace and marine vehicles.

Rita Cunha received the Licenciatura degree in Information Systems and Computer Engineering and the Ph.D. degree in Electrical and Computer Engineering from the Instituto Superior Técnico (IST), Universidade de Lisboa, Portugal, in 1998 and 2007, respectively. She is currently an Associate Professor with the Department of Electrical and Computer Engineering of IST and a researcher with the Institute for Systems and Robotics, LARSyS, Lisbon. Her research interests include nonlinear dynamical systems and control, multi-agent systems, cooperative control, optimal control and model predictive control applied to multi-vehicle trajectory generation and vision-based control with application to autonomous aerial vehicles.