2021 International Conference on Unmanned Aircraft Systems (ICUAS)
Athens, Greece. June 15-18, 2021

# Planning Parcel Relay Manoeuvres for Quadrotors

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

*Abstract*— **In this paper, a strategy for planning aggressive collision-free parcel relay manoeuvres for quadrotors is proposed. The method relies on the generation of optimal polynomial trajectories with acceleration constraints in order to coordinate the attitude of the vehicles during the package exchange. The problem is formulated as a mixed-integer quadratic program where the integer constraints ensure collision avoidance. The manoeuvre is divided into three phases and the dynamical model of the robots is considered to ensure the vehicles keep a suitable relative orientation during the parcel transfer. Simulation results demonstrate the success of the presented strategy.**

## I. INTRODUCTION

In the past few years, the impressive technological developments in a vast array of fields of study, tied to the commitment of the scientific community to devise and build unmanned aerial vehicles (UAVs), triggered a steady, continuous growth of the application domain for these vehicles. Amongst the enormous list of applications suited for the use of drones, establishing a drone delivery service is definitely one of the hottest.

The versatility of drones, specially those which fall into the category of miniature aerial vehicles (MAVs), soon caught the eye of the logistics and e-commerce giants [1] and many other enterprises. Driven by the ambition of reducing the time and cost associated with parcel delivery, multiple MAVs have been developed. Numerous experimental tests, ranging from preliminary to several months long trials, have been conducted and some delivery services using UAVs have been established since the 2010s [2]–[5].

In this work, quadrotors are considered. So as to overcome the stringent limitations on the range and endurance of a single vehicle and thus enable an effective, expeditious package delivery, in-flight parcel exchange between drones is proposed, within the scope of project REPLACE [6]. The focus of this paper is on the generation of collision-free optimal trajectories which are adequate for these manoeuvres. The problem of controlling a mechanism responsible for holding and exchanging packages between vehicles is also addressed in the project, for which there is a patent pending, but falls outside the scope of this work.

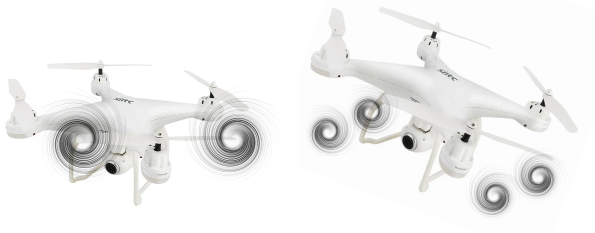


Fig. 1. Illustration of blade tip vortexes at different flight conditions. The figure to the left corresponds to the rotorcraft being in OGE (Out of the Ground Effect) hover; the figure to right, forward flight.

In order to perform the relay manoeuvre, beyond ensuring the two vehicles do not collide with one another, it is of the uttermost importance to account for the aerodynamic interactions between both vehicles. As it is known from rotor aerodynamics, when a rotor is at hover, vortexes form at the blade tips of the propellers that generate a turbulent downwash. If a smaller, or similar-sized quadcopter were to hover below the first, the turbulent airflow coming from above would very likely put the rotors of this aircraft in a condition named Vortex Ring State (VRS). In this situation, the rotors stall and, therefore, enough thrust cannot be produced to keep the drone hovering [7]. However, as illustrated in Fig. 1, when a rotor moves in forward flight the downwash becomes increasingly less turbulent, up to a point the blade tip vortexes are outrun by the aircraft, leaving the rotor operating in undisturbed air.

This suggests the relay manoeuvre should be aggressive in the sense that high manoeuvrability and high-speed navigation is demanded from the aircraft. In [8]–[10], the authors leverage on the *differential flatness* of the quadrotor dynamics to formulate convex optimisation problems which include constraints on the position, velocity, acceleration, and higher order derivatives. Acceleration constraints are also imposed in [11] to embed attitude constraints into the trajectory generation problem. In [12], aggressive manoeuvres are defined as a sequence of trajectory segments to reach some configuration in state space.

Regarding the problem of collision avoidance, mixed-integer programs have been proposed to impose convex constraints, by means of slack variables [9], [13], [14]. The hyperplane theorem is employed to formulate constraints in order to keep an autonomous ground vehicle (AGV) sufficiently far away from obstacles [15]. Non-convex avoidance constraints for a fleet of quadrotors were proposed in [16].

Building upon the seminal works [8], [9], [12], the main

*This work was partially funded by the FCT projects CAPTURE (PTDC/EEI-AUT/1732/2020), REPLACE (LISBOA-01-0145-FEDER-032107), and HARMONY (LISBOA-01-0145-FEDER-031411) which include Lisboa 2020 and PIDDAC funds, and also the research units projects LARSYS (UIDB/EEA/50009/2020) and CTS (UIDB/EEA/00066/2020).

[1] João Pinto and Rita Cunha are with Institute for Systems and Robotics (ISR), LARSYS, Instituto Superior Técnico, Universidade de Lisboa, 1049-001 Lisbon, Portugal. joao.s.pinto@tecnico.ulisboa.pt, rita@isr.tecnico.ulisboa.pt.

[2] Bruno J. Guerreiro is with ISR and also Department of Electrical and Computer Engineering and CTS-Uninova, Universidade Nova de Lisboa, 2829-516 Caparica, Portugal. bj.guerreiro@fct.unl.pt.

2021 International Conference on Unmanned Aircraft Systems (ICUAS) | 978-1-6654-1535-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICUAS51884.2021.9476757

contribution of this paper is the proposal of a strategy for generating aggressive relay manoeuvres reference trajectories from a centralised mixed-integer quadratic problem (MIQP), where the binary variables are responsible for ensuring collision-avoidance. The conducted experiments in simulation environment demonstrate the viability of such strategy. The generated trajectories avert inter-vehicle collisions and enable the quadrotors to be properly oriented to perform the parcel transfer.

The remainder of this paper is structured as follows. In Section II, the model of the quadrotor is presented. Section III describes trajectory planning algorithms for one and two vehicles with the imposition of collision avoidance constraints, while the proposed relay manoeuvre strategy is introduced in Section IV. With the objective of validating the generated trajectories, Section V presents summarily a controller drawn from the literature; Section VI, the analysis of simulation experiments; and, finally, some concluding remarks are made in Section VII.

## II. QUADROTOR DYNAMICS MODEL

As shown in Fig. 2, let $\mathcal{W}$ denote the world reference frame, defined by the axes $x_W = \begin{bmatrix}1 & 0 & 0\end{bmatrix}^T$, $y_W = \begin{bmatrix}0 & 1 & 0\end{bmatrix}^T$, and $z_W = \begin{bmatrix}0 & 0 & 1\end{bmatrix}^T$, and $\mathcal{B}$ denote the body frame attached to the vehicle with origin at the centre of mass. The axes $\boldsymbol{x}_B$ and $\boldsymbol{y}_B$ lie in plane defined by the centre of the four rotors; $\boldsymbol{z}_B$ is perpendicular to this plane and points upwards, in the direction of total thrust.

The configuration of the quadrotor is defined by the location of the centre of mass $\boldsymbol{p} = \begin{bmatrix}x & y & z\end{bmatrix}^T$ and its attitude with respect to $\mathcal{W}$, represented by a rotation matrix $\boldsymbol{R}$ from $\mathcal{B}$ to $\mathcal{W}$. Hence, the system evolves on the Special Euclidean Group $\mathrm{SE}(3)$, which is the semi-direct product of $\mathbb{R}^3$ and the Special Orthogonal Group $\mathrm{SO}(3) := \{\boldsymbol{R} \in \mathbb{R}^{3\times 3} \mid \boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{I},\ \det(\boldsymbol{R}) = 1\}$. Let the linear velocity of the centre of mass in the inertial frame be denoted by $\boldsymbol{v} = \begin{bmatrix}v_x & v_y & v_z\end{bmatrix}^T$ and the angular velocity of $\mathcal{B}$, expressed in $\mathcal{B}$, by $\boldsymbol{\omega} = \begin{bmatrix}p & q & r\end{bmatrix}^T$. The dynamics of the system is described by the following equations

$$\dot{\boldsymbol{p}} = \boldsymbol{v}, \tag{1}$$
$$m\dot{\boldsymbol{v}} = -mg\boldsymbol{z}_W + u_1\boldsymbol{z}_B, \tag{2}$$
$$\dot{\boldsymbol{R}} = \boldsymbol{R}\boldsymbol{S}(\boldsymbol{\omega}), \tag{3}$$
$$\boldsymbol{J}\dot{\boldsymbol{\omega}} + \boldsymbol{S}(\boldsymbol{\omega})\boldsymbol{J}\boldsymbol{\omega} = \boldsymbol{\tau}, \tag{4}$$

where the control input to the system $\boldsymbol{u} \in \mathbb{R}^4$ is comprised of the total thrust $u_1$ and the body torques $\boldsymbol{\tau} = \begin{bmatrix}u_2 & u_3 & u_4\end{bmatrix}^T$ produced by the four rotors, $m$ and $\boldsymbol{J}$ represent the mass and the inertia tensor (about the frame $\mathcal{B}$) of the aircraft, respectively. The constant $g = 9.8\ \mathrm{m\,s^{-2}}$ denotes the Earth gravity and $\boldsymbol{S}(\cdot)$ is a map from $\mathbb{R}^3$ to the space of three-by-three skew-symmetric matrices $\mathfrak{so}(3)$ defined by

$$\boldsymbol{S}(\boldsymbol{a}) = \begin{bmatrix}0 & -a_3 & a_2\\ a_3 & 0 & -a_1\\ -a_2 & a_1 & 0\end{bmatrix},\ \boldsymbol{a} = \begin{bmatrix}a_1 & a_2 & a_3\end{bmatrix}^T. \tag{5}$$

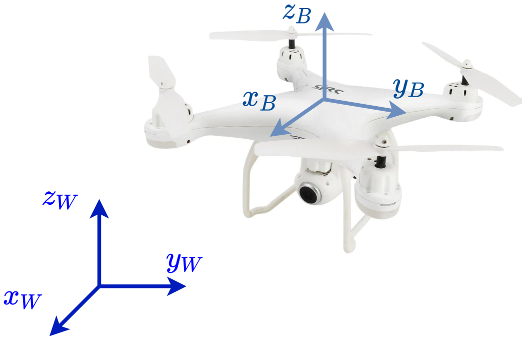


Fig. 2. Illustration of the employed reference frames.

Note that $\mathfrak{so}(3)$ and $\mathbb{R}^3$ are isomorphic. This isomorphism is defined by the condition $\boldsymbol{S}(\boldsymbol{a})\boldsymbol{b} = \boldsymbol{a} \times \boldsymbol{b}$ for all $\boldsymbol{a},\ \boldsymbol{b} \in \mathbb{R}^3$.

### A. Actuation Dynamics

In this work, the thrust and torque actuation dynamics are modelled as first-order systems

$$\dot{\boldsymbol{u}} = -\boldsymbol{K}_u(\boldsymbol{u} - \boldsymbol{u}^*), \tag{6}$$

where $\boldsymbol{K}_u$ is a diagonal matrix with non-negative entries corresponding to the inverse of the time constant of each input variable dynamics and $\boldsymbol{u}^*$ denotes the desired input value. These entries should be large positive values to reflect that the rotor dynamics are much faster than the rigid body dynamics.

## III. TRAJECTORY PLANNING

In this section, a method for planning reference trajectories for the relay manoeuvres is presented. Polynomial trajectory generation algorithms for one and two quadcopters are introduced. Inter-vehicle collision avoidance constraints are included in the trajectory generation problem as binary constraints in the same fashion as in [9], [14].

### A. Single Vehicle

As demonstrated in [8], the quadrotor dynamics are *differentially flat*. The choice of the position $\boldsymbol{p}$ and the yaw angle $\psi$ of the vehicle as *flat outputs* allows the states and inputs of the system to be expressed algebraically in terms of these outputs and their derivatives.

In this manner, piecewise polynomials of order $N$ over $M$ time intervals are selected to serve as basis functions for generating smooth trajectories $\boldsymbol{\sigma}(t)$ in the space of flat outputs,

$$\boldsymbol{\sigma}(t) = \begin{cases}\sum_{k=0}^{N}\boldsymbol{\sigma}_{k1}t^k & t_0 \le t < t_1,\\ \sum_{k=0}^{N}\boldsymbol{\sigma}_{k2}t^k & t_1 \le t < t_2,\\ \quad\vdots & \\ \sum_{k=0}^{N}\boldsymbol{\sigma}_{kM}t^k & t_{M-1} \le t \le t_M,\end{cases} \tag{7}$$

where the vector $\boldsymbol{\sigma}_{kn} = \begin{bmatrix}x_{kn} & y_{kn} & z_{kn} & \psi_{kn}\end{bmatrix}^T$ represents the set of constants associated with the polynomial of degree $k$ of the trajectory segment $\boldsymbol{\sigma}_n(t),\ n = 1, \ldots, M$. In a more compact form, (7) can be written as

$$\boldsymbol{\sigma}(t) = \boldsymbol{\sigma}_n(t) = \boldsymbol{\Sigma}_n \boldsymbol{\eta}(t),\ t_{n-1} \leq t \leq t_n, \tag{8}$$

with $\boldsymbol{\Sigma}_n = \begin{bmatrix} \boldsymbol{\sigma}_{0n} & \boldsymbol{\sigma}_{1n} & \cdots & \boldsymbol{\sigma}_{Nn} \end{bmatrix}$ and $\boldsymbol{\eta}(t) = \begin{bmatrix} 1 & t & \cdots & t^N \end{bmatrix}^T$, $n = 0, \ldots, M$.

An optimisation problem is formulated in order to determine the polynomial coefficients $\sigma_{kn}$ which define the trajectory that passes through a set of waypoints $\boldsymbol{w}_n$ at times $t_n,\ n = 0, \ldots, M$. Since restricting the attitude of the vehicle is required during the parcel exchange manoeuvre, the desired orientation $\boldsymbol{R}_{des}$ is associated with each of the waypoints. As $\boldsymbol{z}_B$ points in the direction of thrust, the attitude of the vehicle, apart from the yaw angle, is determined by the direction of this axis. In this manner, from $\boldsymbol{R}_{des}$, the desired $\boldsymbol{z}_B$ is obtained,

$$\boldsymbol{z}_B(t_n) = \boldsymbol{R}_{des}(t_n)\boldsymbol{z}_W, \tag{9}$$

and the linear velocity dynamics equation presented in (2) can be used for formulating the following linear constraint on the linear acceleration of the vehicle which imposes a desired direction of thrust,

$$\left(\boldsymbol{I}_3 - \boldsymbol{z}_B(t_n)\boldsymbol{z}_B^T(t_n)\right)\left(\dot{\boldsymbol{v}}(t_n) + g\boldsymbol{z}_W\right) = \boldsymbol{0}, \tag{10}$$

where $\boldsymbol{I}_3$ corresponds to the three-by-three identity matrix. The cost function of the proposed optimisation problem is such that it minimises the integral of the squared norm of the snap (the fourth derivative of position) of the trajectory and the second derivative of the yaw angle,

$$\begin{aligned}
\text{minimise} \quad & \int_{t_0}^{t_M} \mu_p \left\| \frac{d^4\boldsymbol{p}}{dt^4} \right\|^2 + \mu_\psi \frac{d^2\psi}{dt^2}^2 \, dt \\
\text{s.t.} \quad & \boldsymbol{\sigma}(t_n) = \boldsymbol{w}_n,\ \text{for } n = 0, \ldots, M, \\
& \dot{\boldsymbol{\sigma}}(t_n) = \boldsymbol{v}_n,\ \text{for } n = 0, \ldots, M, \\
& \ddot{\boldsymbol{\sigma}}(t_M) = \boldsymbol{0} \text{ or free}, \\
& \text{Equation (10)},\ \text{for } n = 0, \ldots, M,
\end{aligned} \tag{11}$$

where $\mu_p,\ \mu_\psi \in \mathbb{R}$, $\boldsymbol{w}_n \in \mathbb{R}^3 \times (-\pi, \pi]$ and $\boldsymbol{v}_n \in \mathbb{R}^4$ correspond, respectively, to the position and yaw waypoint and to the desired linear velocity and yaw angle time derivative $\dot{\psi}$ at time $t_n$. The meaningfulness of this cost function 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 [8].

So as to guarantee the smoothness of the generated trajectory, constraints are introduced to ensure the continuity of the first four derivatives of $\boldsymbol{p}$ and the first two derivatives of $\psi$. In particular, to achieve the desired orientation at each waypoint $\boldsymbol{w}_n$, it is necessary to impose the continuity of acceleration and lateral jerk (the third derivative of $x$ and $y$) to ensure the continuity of attitude and roll and pitch rate trajectories. However, the continuity of these position derivatives alone is not sufficient to make sure a quadrotor is able to follow accurately every single trajectory. Saturation and rate limit constraints may impede the execution of some manoeuvres.

Even though the objective function implicitly minimises the necessary input body torques $\boldsymbol{\tau}$ to track the generated trajectory, one may also enforce bound constraints on the aforementioned derivatives of position to incorporate saturation and rate limit constraints into (11) with the objective of further ensuring the feasibility of trajectories.

In the context of this work, it was opted not to include these constraints in (11) for the reasons discussed in Section III-C.

In what concerns the determination of the coefficients $\sigma_{kn}$, the optimisation problem is readily solved by reformulating (11) as a quadratic program (QP),

$$\begin{aligned}
\underset{\boldsymbol{c}}{\text{minimise}} \quad & \boldsymbol{c}^T \mathcal{H} \boldsymbol{c} + \boldsymbol{f}^T \boldsymbol{c} \\
\text{s.t.} \quad & \boldsymbol{A}\boldsymbol{c} = \boldsymbol{b}.
\end{aligned} \tag{12}$$

The vector $\boldsymbol{c}$ contains $F = 4(N+1)M$ optimisation variables which correspond to the polynomial coefficients that characterise the whole trajectory, while the constants $\mathcal{H}$, $\boldsymbol{f}$, $\boldsymbol{A}$, and $\boldsymbol{b}$ can be obtained from (11).

### *B. Planning for Two Vehicles*

Building upon (11), another optimisation problem is formulated for planning trajectories in a centralised fashion for two quadrotors while avoiding inter-vehicle collisions.

Consider two vehicles each with position $\boldsymbol{p}_i(t_k) = \begin{bmatrix} x_i(t_k) & y_i(t_k) & z_i(t_k) \end{bmatrix}^T$, $i \in \{1, 2\}$. The inter-collision avoidance constraints are as follows,

$$\begin{gathered}
x_1(t_k) - x_2(t_k) \geq d_{x12} - b_{1k}V, \\
x_2(t_k) - x_1(t_k) \geq d_{x21} - b_{2k}V, \\
y_1(t_k) - y_2(t_k) \geq d_{y12} - b_{3k}V, \\
y_2(t_k) - y_1(t_k) \geq d_{y21} - b_{4k}V, \\
z_1(t_k) - z_2(t_k) \geq d_{z12} - b_{5k}V, \\
z_2(t_k) - z_1(t_k) \geq d_{z21} - b_{6k}V, \\
\sum_{j=1}^{6} b_{jk} \leq 5.
\end{gathered} \tag{13}$$

The first six inequalities define two exclusion cuboid-shaped regions oriented with the inertial frame around each vehicle. In other words, around quadrotor $i$ a rectangular prism is considered within which the quadrotor $j,\ j \neq i$, cannot enter. The dimensions of these polyhedron are provided by the values specified by the $d_{xij}$, $d_{yij}$ and $d_{zij}$ constants. These constants should be large enough for the vehicle to be at any attitude and to remain inside the box.

As for the binary variables, these are denoted by $b_{jk},\ 1 \leq j \leq 6$, and are introduced as slack variables; the constant $V$ is large positive number. In this manner, when $b_{jk} = 1$, the $j^{\text{th}}$ constraint is relaxed; should be it zero, this constraint is enforced. This last constraint is fundamental since if the slack variables were all equal to zero, it would be only necessary for one of the six constraints to be satisfied in order to ensure the vehicles were not in collision.

For the generation of trajectories for two quadrotors including the collision avoidance constraints, the vector $\boldsymbol{c}$

is augmented so as to include the polynomial coefficients of the two vehicles and the binary variables. The collision avoidance constraints are enforced at $K$ intermediate time steps, that is, at every time step except for those which are associated with a waypoint. It may be required the vehicles to be closer than permitted by the binary collision avoidance constraints to effectively exchange the parcel at a concrete waypoint.

### C. Computational Complexity and Limitations

The MIQP formulated for the generation of reference trajectories for two quadrotors with inter-vehicle collision avoidance constraints involves $2F + 6K$ decision variables. Out of these, $6K$ are binary and are associated with the collision avoidance constraints. In the worst-case scenario, a branch-and-bound solver may require $2^{6K}$ quadratic programs to be solved [17]. The authors in [9] highlight the short amount of time required to obtain sub-optimal integer solutions. These are nevertheless, not at all times, adequate trajectories, as for a quadcopter to track them high thrust and body torques may be demanded from its rotors. To overcome this problem, one may consider imposing bounds on the derivatives of position for all time steps, however, the sheer amount of inequality constraints introduced would cause numerical issues to be more prone to arise.

The time complexity of this method for generating collision-free trajectories may render it unsuitable for planning long missions. One may weaken the optimality criteria of the mixed-integer program solver with the objective of reducing the time spent on solving the optimisation problem.

Additionally, as also emphasised by the authors of [9], the choice of the spanning polynomial basis functions is of the uttermost importance. For the polynomial basis chosen, it was observed that numerical issues arise when attempting to generate long trajectories with high degree polynomials.

## IV. Relay Planning Strategy

In spite of the addition of (10) to (11), 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 and keeps an adequate orientation for the parcel transfer.

Since the quadcopters performing the manoeuvre must achieve some concrete configuration in state space, a strategy, based on the work published in [12], is devised so that the vehicles are driven to a goal state $G$.

It is established that the trajectory for each vehicle is composed of three segments, as illustrated in Fig. 3. In order to generate these trajectory segments, $G$ is provided by specifying the desired position $\boldsymbol{p}_G$, velocity $\boldsymbol{v}_G$, attitude $\boldsymbol{\lambda}_G = \begin{bmatrix}\phi_G & \theta_G & \psi_G\end{bmatrix}^T$ parametrised by $Z-X-Y$ Euler angles, and an angular velocity $\boldsymbol{\omega}_G$ of zero. The reason for expressing the desired attitude using Euler angles rather than rotation matrices is explained soon.

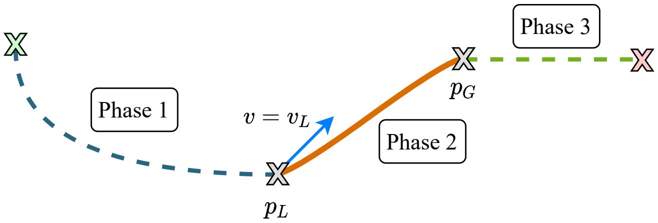


Fig. 3. Diagram illustrating the segments of the trajectory in order to reach the goal state $G$.

- 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. 3) to a launching position $\boldsymbol{p}_L$, from where the vehicle departs for the goal position $\boldsymbol{p}_G$ with a launch velocity $\boldsymbol{v}_L$, and with adequate roll, $\phi$, and pitch, $\theta$, angles 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 orientation parameterised by $\boldsymbol{\lambda}_G$. During this segment, the vehicles are to exchange the parcel;
- 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. 3). It may be imposed that the aircraft finishes at hover. Moreover, it should be ensured that the vehicles move apart from one another during this phase.

All segments of the relay manoeuvre are generated using the method described in the previous section, where collision avoidance constraints are enforced. Orientation and linear velocity constraints are imposed at every waypoint.

In order to obtain the launch point $\boldsymbol{p}_L$ and the launch velocity $\boldsymbol{v}_L$ necessary for *Phase 2*, the quadrotor dynamics coupled with an attitude control law are integrated backwards in time, starting from the desired state for the relay manoeuvre, $G$. As this backwards integration requires a control law, $u_1$ is kept constant during this phase, whereas a simple proportional derivative (PD) control law computes the body torques input $\boldsymbol{\tau}$,

$$\boldsymbol{\tau} = \boldsymbol{K}_\lambda \boldsymbol{e}_\lambda - \boldsymbol{K}_\omega \boldsymbol{e}_\omega, \tag{14}$$

where $\boldsymbol{K}_\lambda$ and $\boldsymbol{K}_\omega$ are diagonal gain matrices. The attitude and angular velocity errors are denoted by $\boldsymbol{e}_\lambda$ and $\boldsymbol{e}_\omega$, respectively,

$$\boldsymbol{e}_\lambda = \boldsymbol{\lambda}_{des} - \boldsymbol{\lambda}, \tag{15}$$

$$\boldsymbol{e}_\omega = \boldsymbol{\omega}_{des} - \boldsymbol{\omega}. \tag{16}$$

The variable $\boldsymbol{\lambda}$ represents the aircraft orientation parametrised by $Z-X-Y$ Euler angles. The Euler angles reference $\boldsymbol{\lambda}_{des}: \ [0\ T) \to \mathbb{R}^3$ for *Phase 2* is defined as the following map,

$$\boldsymbol{\lambda}_{des}(t) = \begin{cases} \begin{bmatrix}0 & 0 & \psi_G\end{bmatrix}^T & 0 \leq t < t_b, \\ \\ \begin{bmatrix}\phi_G & \theta_G & \psi_G\end{bmatrix}^T & t_b \leq t < T. \end{cases} \tag{17}$$

Provided that $T$ denotes the duration of *Phase 2*, the choice of $t_b$ establishes the time window for which the vehicles are in a suitable orientation to exchange the parcel. As the dynamics are integrated starting from $G$ and the thrust input $u_1$ is constant, the position of gripping mechanism attached to vehicles should be taken into account to decide whether $t_b$ should be closer to zero or closer to $T$. Despite the definition of $\boldsymbol{\lambda}_{des}(\cdot)$ in (17) imposing the roll and pitch angles to be equal to zero at the beginning of *Phase 2*, the attitude control law presented in (14) is able to track the desired Euler angles step reference, as long as the singularity at $\phi = \pm\pi/2$ is avoided.

For the computation of the desired body rate $\boldsymbol{\omega}_{des}$, a first-order numerical differentiation method is employed to approximate the time derivative of $\boldsymbol{\lambda}_{des}$. The desired body rate, expressed in $\mathcal{B}$, is then obtained by

$$\boldsymbol{\omega}_{des} = \begin{bmatrix} \cos\theta & 0 & -\cos\phi\sin\theta \\ 0 & 1 & \sin\phi \\ \sin\theta & 0 & \cos\phi\cos\theta \end{bmatrix} \dot{\boldsymbol{\lambda}}_{des}. \tag{18}$$

If the desired attitude reference for *Phase 2* were parametrised by rotation matrices, one could attempt to use the orientation dynamics equation presented in (3) to determine $\boldsymbol{\omega}_{des}$, since

$$\boldsymbol{S}(\boldsymbol{\omega}_{des}) = \boldsymbol{R}_{des}^T \dot{\boldsymbol{R}}_{des}. \tag{19}$$

However, it was verified that using finite-difference methods to find the time derivative of $\boldsymbol{R}_{des}$ led to numerical errors.

The three trajectory segments are obtained by formulating separate optimisation problems. This revealed to be necessary as, even though a solution for the relay manoeuvre reference was achieved, in some cases, one could verify that some constraints regarding the continuity of the trajectory were partially violated. Lowering the constraint violation tolerance would render the problem infeasible. Furthermore, having in mind the large number of binary variables which might be involved in the trajectory generation problem, splitting this problem into multiple is an advantage.

The continuity of position and velocity is enforced between the three segments.

## V. TRACKING CONTROLLER

In the spirit of evaluating whether two quadrotors are indeed able to follow the trajectories generated using the proposed relay strategy and to exchange a parcel during flight, in the following section simulation results are presented for two vehicles equipped with the geometric controller on SE(3) presented in [8].

For the sake of completeness, the overall controller structure is as follows. The outer loop of the controller is responsible for tracking a trajectory defined by the reference position $\boldsymbol{p}_T$, linear velocity $\boldsymbol{v}_T$ and linear acceleration $\boldsymbol{a}_T$. This loop computes the desired thrust vector $\boldsymbol{F}_{des}$ which is used by the inner loop, along with the reference yaw angle $\psi_T$ and its time derivative $\dot{\psi}_T$, and also the reference jerk $\boldsymbol{j}_T$ to define the desired body frame orientation $\boldsymbol{R}_{des}$ and angular velocity $\boldsymbol{\omega}_{des}$. In this manner, the three control torques $\boldsymbol{\tau}$ are computed.

For tracking the reference trajectory the position and linear velocity errors are defined as

$$\boldsymbol{e}_p = \boldsymbol{p} - \boldsymbol{p}_T, \tag{20}$$

$$\boldsymbol{e}_v = \boldsymbol{v} - \boldsymbol{v}_T, \tag{21}$$

which are used to define a PD control law, with the inclusion of the reference linear acceleration $\boldsymbol{a}_T$ as feedforward term,

$$\boldsymbol{F}_{des} = -\boldsymbol{K}_p\boldsymbol{e}_p - \boldsymbol{K}_v\boldsymbol{e}_v + mg\boldsymbol{z}_W + m\boldsymbol{a}_T. \tag{22}$$

The matrices $\boldsymbol{K}_p$ and $\boldsymbol{K}_v$ are positive definite and correspond to the position and velocity controller gains. The control input $u_1$ is obtained from the projection of the desired thrust $\boldsymbol{F}_{des}$ onto the body frame axis $\boldsymbol{z}_B$,

$$u_1 = \boldsymbol{F}_{des}^T \boldsymbol{z}_B. \tag{23}$$

So as to compute the other three inputs, the input body torques $\boldsymbol{\tau}$, one begins by determining the desired orientation of the body frame $\boldsymbol{R}_{des}$. By the assumption that $\boldsymbol{z}_B$ points in the direction of total thrust, given that $\|\boldsymbol{F}_{des}\| \neq 0$, the desired $\boldsymbol{z}_B$ is

$$\boldsymbol{z}_{B,des} = \frac{\boldsymbol{F}_{des}}{\|\boldsymbol{F}_{des}\|}. \tag{24}$$

Given $\boldsymbol{z}_{B,des}$, as the reference yaw angle $\psi_T$ is provided, the desired orientation $\boldsymbol{R}_{des}$ is easily computed. Finally, the orientation error $\boldsymbol{e}_R$ and the angular velocity error $\boldsymbol{e}_\omega$ are defined as

$$\boldsymbol{e}_R = \frac{1}{2}\boldsymbol{S}^{-1}\left(\boldsymbol{R}_{des}^T\boldsymbol{R} - \boldsymbol{R}^T\boldsymbol{R}_{des}\right), \tag{25}$$

$$\boldsymbol{e}_\omega = \boldsymbol{\omega} - \boldsymbol{\omega}_{des}, \tag{26}$$

where $\boldsymbol{S}^{-1} : \mathfrak{so}(3) \mapsto \mathbb{R}^3$ corresponds to the inverse of the map defined in (5). The control moments to be input to the quadcopter is computed using the established attitude errors,

$$\boldsymbol{\tau} = -\boldsymbol{K}_R\boldsymbol{e}_R - \boldsymbol{K}_\omega\boldsymbol{e}_\omega, \tag{27}$$

where $\boldsymbol{K}_R$ and $\boldsymbol{K}_\omega$ are positive definite matrices representing the orientation and angular rates gains.

## VI. SIMULATION RESULTS

As the binary collision avoidance constraints described previously do not consider the attitude of the vehicles, in order to monitor the possibility of collision, in a more precise manner, the distance between them is computed taking into account their orientation. To this end, two polyhedrons are defined around each aircraft. Each of these polyhedrons is a collection of planar constraints expressed in the body frame $\mathcal{B}$ of the respective vehicle. Using the position and orientation information of the vehicles, the polyhedrons are then expressed in the world frame $\mathcal{W}$.

In this manner, an optimisation problem is formulated so as to determine two points, one in each of the polyhedrons, that minimises the square of the $\ell_2$ norm between them,

$$\begin{aligned} \underset{\boldsymbol{x}_1,\boldsymbol{x}_2}{\text{minimise}} \quad & \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2 \\ \text{s.t.} \quad & \boldsymbol{x}_1 \in \mathcal{P}_1(\boldsymbol{p}_1, \boldsymbol{R}_1), \\ & \boldsymbol{x}_2 \in \mathcal{P}_2(\boldsymbol{p}_2, \boldsymbol{R}_2), \end{aligned} \tag{28}$$

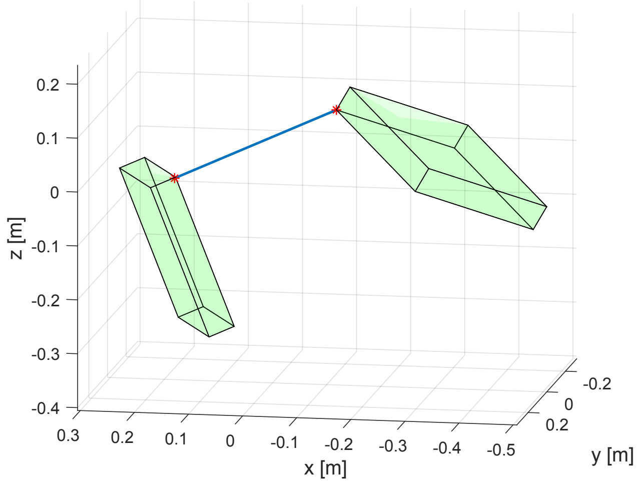


Fig. 4. Illustration of the solution of the optimisation problem that allows to compute the distance between two polyhedrons. Two rectangular parallelipipeds were considered for this example.

where $\mathcal{P}_j(\boldsymbol{p}_j, \boldsymbol{R}_j)$ denotes the polyhedron wrapped around quadrotor $j$, located at position $\boldsymbol{p}_j$ with attitude $\boldsymbol{R}_j$. This problem is convex as the cost function is a semi-definite positive quadratic function and the constraints are linear inequalities. The optimal value of (28) establishes the squared distance function between two polyhedrons,

$$D(\mathcal{P}_1, \mathcal{P}_2) = \inf \left\{ \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2 \mid \boldsymbol{x}_1 \in \mathcal{P}_1, \boldsymbol{x}_2 \in \mathcal{P}_2 \right\}. \quad (29)$$

Despite not being possible to express (29) in closed-form, it can be deduced that $D(\cdot,\cdot)$ is non-convex both in position $\boldsymbol{p}_1$, $\boldsymbol{p}_2$ and orientation $\boldsymbol{R}_1$, $\boldsymbol{R}_2$ of the polyhedrons, since polyhedra are shapes with flat faces and sharp corners. Be as it may, the squared distance function (29) is a handy analysis tool as it allows to compute with relative ease the Euclidean distance between any two polytopes living in the same dimension.

An illustration of the solution of (28) is presented in Fig. 4. The red points correspond to the optimal solution $\{\boldsymbol{x}_1^*, \boldsymbol{x}_2^*\}$ and the length of the blue line linking those points is the square root of $D(\mathcal{P}_1, \mathcal{P}_2)$ evaluated at the optimum.

The simulation results presented in this section were obtained from a computer with the Windows 10 operating system installed running MATLAB, equipped with an Intel Core i7-6700HQ CPU @2.60GHz, and 16,00GB of RAM, where the MIQPs were solved using the branch-and-bound solver provided by CPLEX [18].

For this simulation, two equal drones with $m = 0.152\,\text{kg}$ and $\boldsymbol{J} = \text{diag}\left(\begin{bmatrix}0.0138 & 0.0110 & 0.0082\end{bmatrix}\right)\text{kg}\,\text{m}^2$ were considered.

### A. Collision Avoidance

In this section, the proposed method for generating trajectories for two vehicles with the collision avoidance constraints is evaluated. For this experiment, the established waypoints $\boldsymbol{w}_{ij} = \begin{bmatrix}x_{ij} & y_{ij} & z_{ij} & \psi_{ij}\end{bmatrix}^T$ are schematically represented in Fig. 5. The green crosses correspond to waypoints of the first vehicle, the magenta ones, to the waypoints of the second. The vector $\boldsymbol{w}_{ij}$ denotes the $i^{\text{th}}$ waypoint of vehicle $j$. In this diagram, a time scale is included so as to associate each waypoint with a time instant. The generated trajectories consider that both vehicles start tracking the reference trajectory with zero linear velocity and finish at hover.

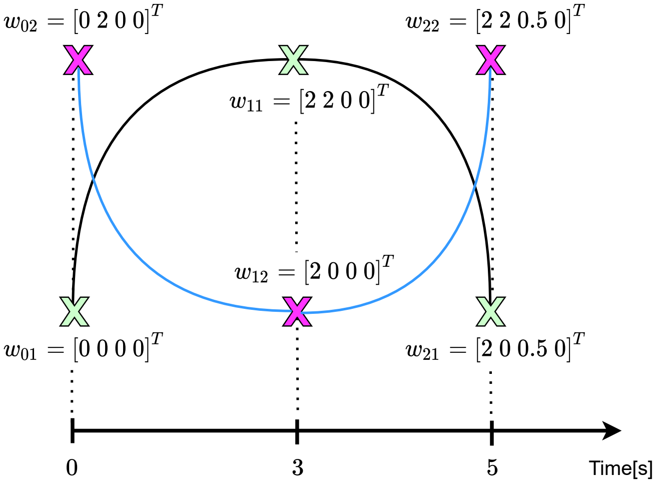

$w_{02} = [0\ 2\ 0\ 0]^T$
$w_{22} = [2\ 2\ 0.5\ 0]^T$
$w_{11} = [2\ 2\ 0\ 0]^T$
$w_{12} = [2\ 0\ 0\ 0]^T$
$w_{01} = [0\ 0\ 0\ 0]^T$
$w_{21} = [2\ 0\ 0.5\ 0]^T$
0
3
5
Time[s]

Fig. 5. Schematic representation of the established waypoints.

TABLE I
Performance comparison of the results obtained.

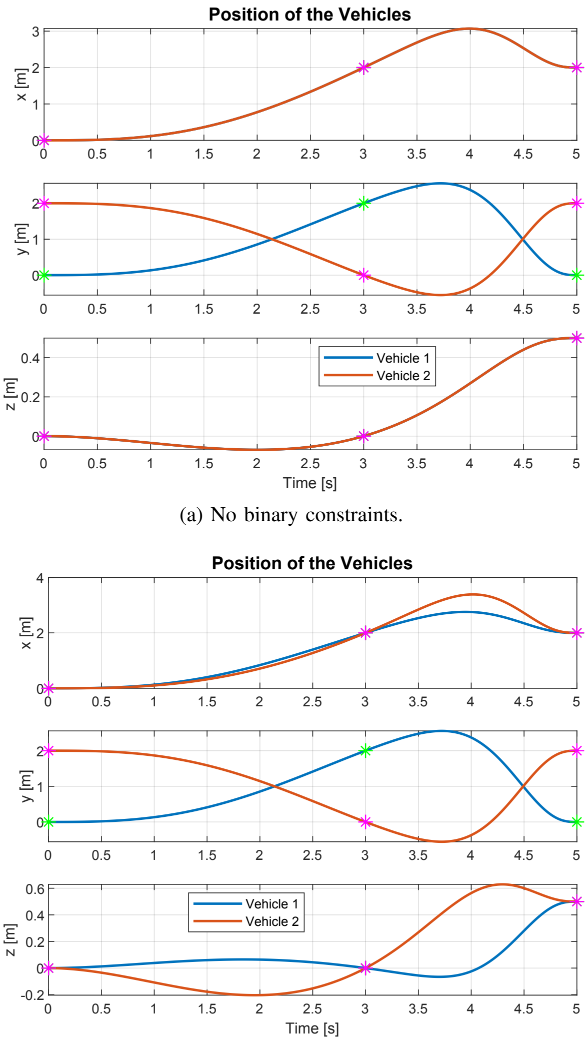
| Figure | $n_c$ | $n_b$ | Cost | Time [s] |
|---|---|---|---|---|
| 6a | 208 | – | 7.3334 | 0.1090 |
| 6b | 208 | 2988 | 7.8698 | 43.1090 |

The generated position reference trajectories excluding and including the collision avoidance binary constraints are plotted in Fig. 6a and 6b, respectively. It is observed that the binary constraints enforce the vehicles to be further apart by performing a comparison between these figures. The trajectory in the $x$ and $z$ axes are modified in order to ensure the two vehicles are never in the same point throughout the trajectory. This result is further confirmed by computing the distance between the vehicles using the orientation reference trajectories and employing (29). The orientation reference can be obtained from the linear acceleration and yaw angle generated trajectories. Fig. 7 reveals that, in fact, the binary constraints effectively avert collisions between the quadrotors.

Table I details the performance results in this section. Let $n_c = 2F$ and $n_b = 6K$ denote the number of continuous and binary variables involved in an optimisation problem, respectively. Notwithstanding the fact that, nowadays, there is a plethora of solvers which are able to achieve an optimal solution of a large convex QP in a fraction of a section, one can verify that the inclusion of the binary variables may increase the computational time significantly.

### B. Relay Manoeuvre

The proposed strategy for planning relay manoeuvres is evaluated, in this section, by presenting and analysing



(a) No binary constraints.

(b) Inclusion of binary constraints.

Fig. 6. Position reference trajectories generation excluding and including binary constraints.

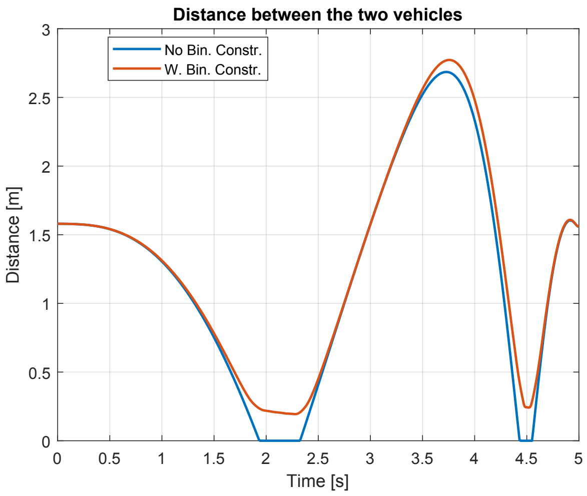


Fig. 7. Distance between the two vehicles comparison.

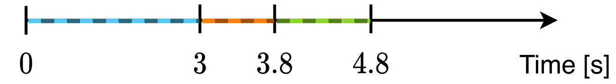


Fig. 8. Time axis representing the duration of the every phase of the relay trajectory.

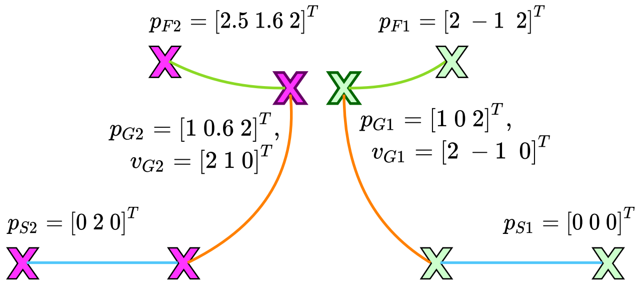


Fig. 9. Illustration of the relay manoeuvre.

the simulation results of a manoeuvre in which the two vehicles vary their roll angle until the desired orientation $\boldsymbol{\lambda}_G$ is achieved, while simultaneously shortening the lateral distance between them.

It is assumed that some sort of gripping mechanism is attached to the bottom of each vehicle to establish the desired orientation of the vehicle at the goal state $G$ and the value of $t_b$.

The time duration of each phase of the manoeuvre is depicted in Fig. 8. Throughout this section, the colour scheme adopted in Section IV is used: *Phase 1* is associated with the colour blue; *Phase 2*, with the colour orange; and *Phase 3*, with the colour green.

In Fig. 9, a schematic representation of the established waypoints for this manoeuvre is provided. Let $\boldsymbol{p}_{Sj}$, $\boldsymbol{p}_{Gj}$, $\boldsymbol{p}_{Fj} \in \mathbb{R}^3$ denote the start, goal, and final positions of vehicle $j$, respectively. During *Phase 2*, one vehicle rolls 84° to left, and the other, to right, so as to avoid the singularity $\phi = \pi/2$. The velocity of each quadcopter at the goal point $\boldsymbol{v}_{Gj}$ was selected in accordance to the defined final position and $\boldsymbol{\lambda}_G$. Both vehicles start with zero linear velocity and finish at hover.

Fig. 14a and 14d present the position; Fig. 14b and 14e the velocity; and Fig. 14c and 14f the attitude trajectory plots of both vehicles, respectively. These graphs indicate that both aircraft are able to track, with minimal error, the position, velocity and attitude references while not colliding into one another, as Fig. 10 evinces. One can also observe that the imposed orientation constraints are satisfied.

So as to visualise whether the vehicles are indeed able to exchange a parcel while tracking the planned trajectory, Fig. 11 presents the the position reference trajectories of both vehicles together with the described trajectories during *Phase 2*. This figure reveals that the quadrotors keep a suitable orientation for the relay manoeuvre while simultaneously shortening the distance between them. The analysis of the attitude (Fig. 14c and 14f) and inter-vehicle distance (Fig. 10) plots indicates that the quadrotors keep a suitable orientation for the relay manoeuvre while simultaneously shortening the

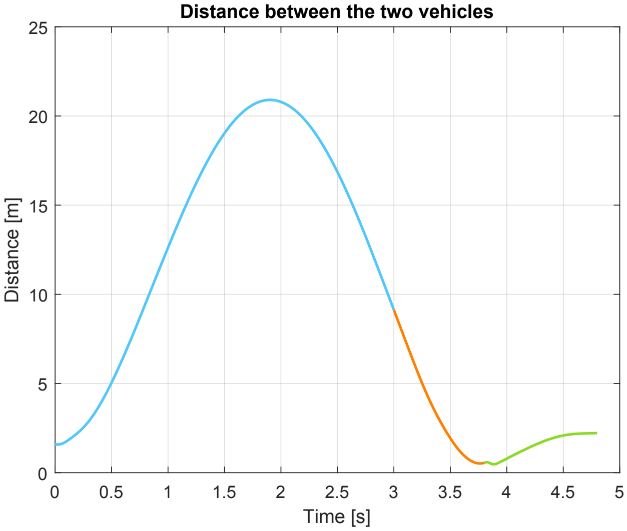


Fig. 10. Distance between the quadcopters computed using the square root of (29).

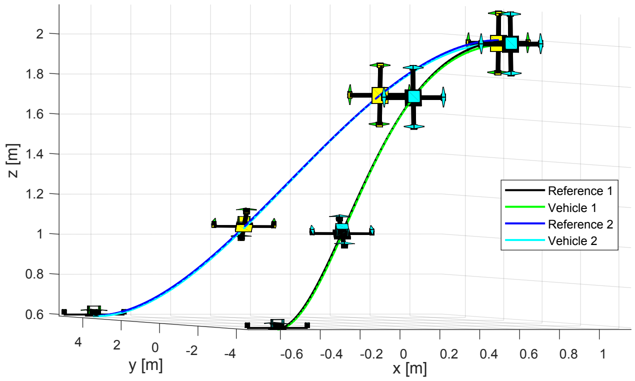


Fig. 11. Composite illustration of the vehicles tracking the reference of *Phase 2* (an animation is available at https://youtu.be/0xPtaoa6kfY).

distance between them.

Fig. 12 reveals the position reference trajectories of both vehicles together with the described trajectories during *Phase 3*. The aircraft swiftly move apart from each other.

The control inputs provided to the vehicles throughout the trajectory are represented in Fig. 13. The lines in lighter colours correspond to the inputs provided to the first vehicle, whereas the darker, to the second. It is observed that both track the reference trajectories without demanding high input values.

Finally, Table II arranges the value of the cost function and the time to attain an optimal solution for the three phases of the manoeuvre. Although the performance results of the relay manoeuvre indicate it can be planned in a fair short amount of time, when there is a chance of the two vehicles being in the course of collision at any point of the trajectory, the solver may take significantly longer to provide an optimal solution.

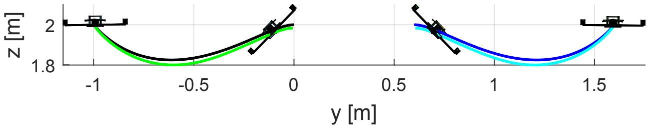


Fig. 12. Composite illustration of the vehicles tracking the reference of *Phase 3* (an animation is available at https://youtu.be/0xPtaoa6kfY).

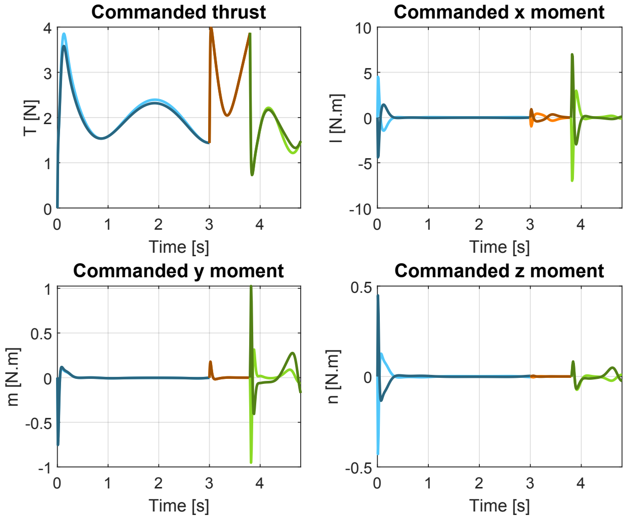


Fig. 13. Commanded thrust and body torques.

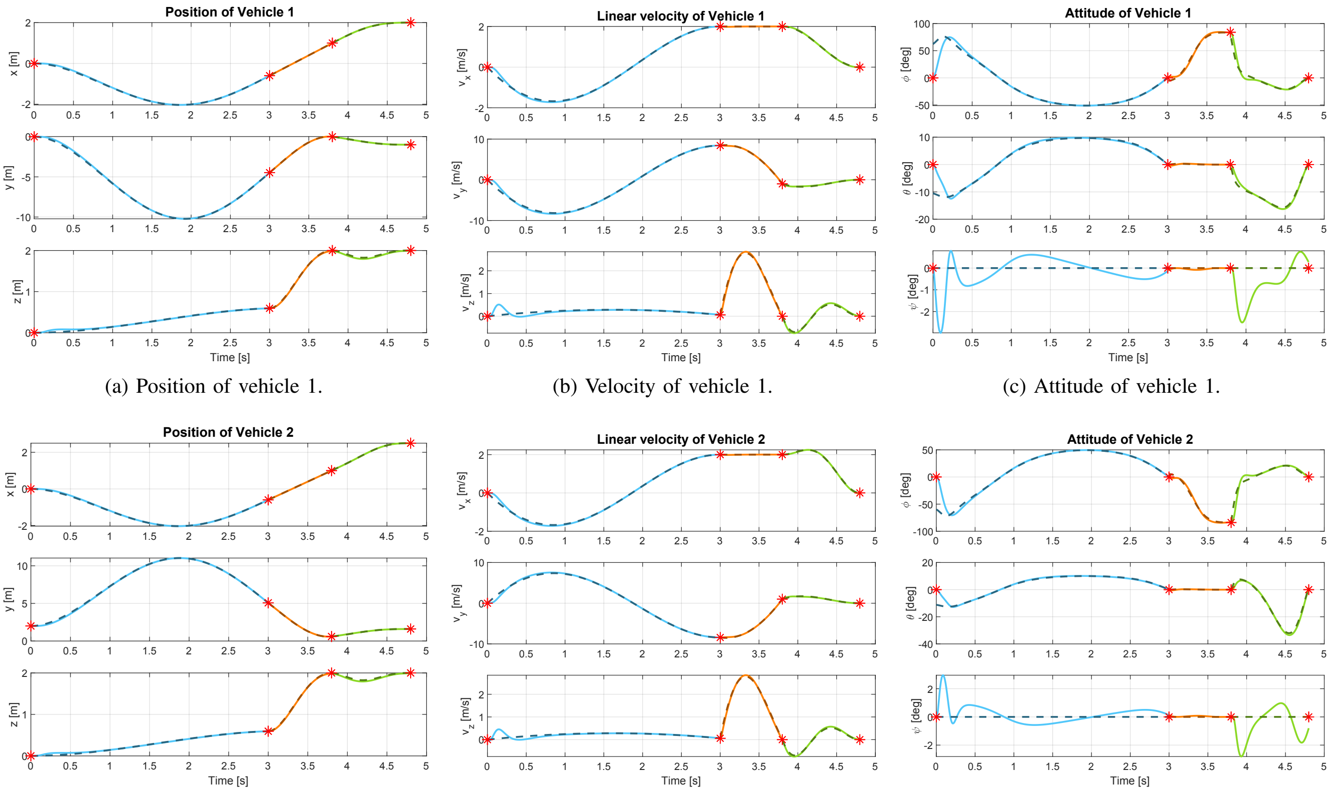
## VII. Conclusions

The proposed strategy for planning relay manoeuvres using a mixed-integer quadratic program (MIQP) proved to be successful, ensuring the generation of polynomial trajectories which not only avoid collisions between the vehicles, but also enable the aircraft to be in a suitable orientation to perform the package exchange. It was verified that the imposition of linear constraints on the linear acceleration effectively coordinates the orientation of the vehicles. Regarding the possibility that a tracking controller may not be able to follow the desired trajectories, the relay manoeuvre was spilt into three phases. The second phase of the manoeuvre is when the vehicles are to transfer the package to one another and, therefore, the trajectory generated for this phase builds upon the dynamical model of the robots.

In terms of future work, experimental tests to further study the aerodynamics interactions between the vehicles are being considered. Furthermore, it is relevant to evaluate the robustness of the proposed backwards integration strategy under disturbances, namely wind, and its performance when the vehicle dynamics are coupled with different controller types.

TABLE II
Relay Manoeuvre - Performance results.

| | *Phase 1* | | *Phase 2* | | *Phase 3* | |
|---|---|---|---|---|---|---|
| | Cost | T [s] | Cost | T [s] | Cost | T [s] |
| Roll Relay | 1545 | 2.860 | 14407 | 1.235 | 52717 | 0.5990 |



(a) Position of vehicle 1.

(b) Velocity of vehicle 1.

(c) Attitude of vehicle 1.

(d) Position of vehicle 2.

(e) Velocity of vehicle 2.

(f) Attitude of vehicle 2.

Fig. 14. Position, velocity, and attitude actual and reference trajectories for the relay manoeuvre. The references are depicted using dashed lines and darker colours; the actual trajectories, using continuous lines and lighter colours. The red stars correspond to the established waypoints.

The flexibility of the algorithm would be enhanced if each quadrotor considered its initial and launch configurations to set the time duration of its own trajectory segment of *Phase 1* of the manoeuvre. Finally, it is worth pursuing sensor-based cooperative control strategies which fully ensure a faithful tracking of the proposed optimal trajectories and a successful parcel exchange.

## References

[1] R. D'Andrea, "Guest Editorial - Can Drones Deliver?" *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 3, pp. 647–648, 2014.

[2] "Deutsche Post DHL Group — DHL Parcelcopter," [Online] Available: https://www.dpdhl.com/en/media-relations/specials/dhl-parcelcopter.html.

[3] M. Giles, "Zipline launches the world's fastest commercial delivery drone," *MIT Technology Review*, April 3, 2018, [Online] Available: https://www.technologyreview.com/2018/04/03/144144/zipline-launches-the-worlds-fastest-commercial-delivery-drone/.

[4] "Amazon — Amazon Prime Air," [Online] Available: https://www.amazon.com/b?node=8037720011l.

[5] S. Burns, "Drone meets delivery truck," *UPS Longitudes*, February 22, 2017, [Online] Available: https://www.ups.com/us/en/services/knowledge-center/article.page?kid=cd18bdc2.

[6] "Project REPLACE," [Online] Available: http://replace.isr.tecnico.ulisboa.pt.

[7] J. Leishman, *Principles of Helicopter Aerodynamics*, 2nd ed. Cambridge University Press, 2006.

[8] D. Mellinger and V. Kumar, "Minimum Snap Trajectory Generation and Control for Quadrotors," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2520–2525, 2011.

[9] D. Mellinger, A. Kushleyev, and V. Kumar, "Mixed-Integer Quadratic Program Trajectory Generation for Heterogeneous Quadrotor Teams," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 477–483, 2012.

[10] G. Loianno, C. Brunner, G. McGrath, and V. Kumar, "Estimation, control, and planning for aggressive flight with a small quadrotor with a single camera and imu," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 404–411, 2016.

[11] G. Yu, D. Cabecinhas, R. Cunha, and C. Silvestre, "Quadrotor trajectory generation and tracking for aggressive maneuvers with attitude constraints," *IFAC-PapersOnLine*, vol. 52, no. 12, pp. 55–60, 2019.

[12] D. Mellinger, N. Michael, and V. Kumar, "Trajectory Generation and Control for Precise Aggressive Maneuvers with Quadrotors," in *Int, Symposium on Experimental Robotics*, 2010.

[13] A. Richards and J. P. How, "Aircraft Trajectory Planning With Collision Avoidance Using Mixed Integer Linear Programming," in *Proceedings of the American Control Conference (IEEE Cat. No. CH37301)*, vol. 3. Anchorage, AK: IEEE, 2002, pp. 1936–1941.

[14] T. Schouwenaars, B. De Moor, E. Feron, and J. How, "Mixed Integer Programming for Multi-Vehicle Path Planning," in *2001 European control conference (ECC)*. Porto, Portugal: IEEE, 2001, pp. 2603–2608.

[15] T. Mercy, R. Van Parys, and G. Pipeleers, "Spline-based motion planning for autonomous guided vehicles in a dynamic environment," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 6, pp. 2182–2189, 2017.

[16] F. Augugliaro, A. P. Schoellig, and R. D'Andrea, "Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach," *IEEE International Conference on Intelligent Robots and Systems*, pp. 1917–1922, 2012.

[17] L. A. Wolsey and G. L. Nemhauser, *Integer and Combinatorial Optimization*, 1st ed. New York: John Wiley & Sons, 1988.

[18] International Business Machines Corporation, *IBM ILOG CPLEX Optimization Studio Getting Started with CPLEX for MATLAB V12.8*, 2017.