Time-optimal trajectory planning for multicopter lateral flight

Juan Augusto Paredes Salazar

1 Motivation

As autonomous vehicles become more prevalent in society, a bigger emphasis is given on performance and ways of gauging it. One of these vehicles are multicopters, who are used in photography, precision agriculture, 3D reconstruction, monitoring, and are being investigated for possible uses in medicine transportation for example. For that purpose, a baseline for multicopter trajectories is required to assess the possibility of flight maneuvers and the flight performance of new multicopter designs. This motivates the derivation of time-optimal input and state trajectories for multicopter flight. This document considers the simplified case of lateral flight, in which the multicopter dynamics can be simplified into bicopter dynamics. Furthermore, in this document, the optimization algorithm featured in [1] is compared against a more general trajectory optimization algorithm, featured in the code attached to this document.

This document is organized as follows. Section 2 introduces the multicopter lateral flight dynamics without rotational dynamics. Section 3 explains the derivation of the time-optimal input and state trajectories using Pontryagin's optimality conditions, which includes the formulation of the optimization problem and stating the minimum principle necessary conditions for optimality; a description of the algorithm proposed by [1] to solve the proposed optimization problem is also provided. Section 4 shows the numerical results obtained from solving the optimization problem introduced in Section 3 applied to the system dynamics shown in Section 2. Section 5 formulates the time-optimal trajectory planning problem as a nonlinear program (NLP) by discretizing the nonlinear dynamics using Euler discretization, optimizing over a user-defined number of time steps, and making the time between steps an optimization variable. Section 6 shows the numerical results obtained from solving the NLP introduced in Section 5 and compares the resulting trajectories to those obtained in Section 4. Section 7 include rotational dynamics to the multicopter lateral flight dynamics introduced in Section 2. Section 8 shows the numerical results obtained from solving the NLP introduced in Section 5 applied to the system dynamics shown in Section 7. Finally, Section 9 briefly states the conclusions.

2 Multicopter lateral flight dynamics without rotational dynamics

Consider the multicopter shown in Figure 1. Let $x, z, \theta \in \mathbb{R}$ be the horizontal position, vertical position and pitch angle of the multicopter respectively. Let $f_1, f_2 \geq 0$ be the thrust exerted by each of the rotors and let $f_{\mathrm{T}} \triangleq f_1 + f_2 \geq 0$ be the total thrust. Suppose that the pitch rate $\omega \triangleq \dot{\theta} \in \mathbb{R}$ can be directly modulated, which is an assumption made in [1]. Furthermore, suppose that f_{T} and ω are constrained, such that $f_{\mathrm{T}} \in [\underline{f}_{\mathrm{T}}, \overline{f}_{\mathrm{T}}]$ and $\omega \in [-\overline{\omega}, \overline{\omega}]$, where $\underline{f}_{\mathrm{T}}, \overline{f}_{\mathrm{T}}, \overline{\omega} > 0$; the constraints on f_{T} represent the minimum and maximum thrust all rotors are able to provide, while the constraints on ω represent the limits of the gyroscopic measurements obtained from onboard sensors. Then, the dynamics for the lateral flight of a multicopter are given by

$$\ddot{x} = \frac{f_{\mathrm{T}}}{m} \sin\theta, \tag{1}$$

$$\ddot{z} = \frac{f_{\mathrm{T}}}{m} \cos\theta - g, \tag{2}$$

$$\dot{\theta} = \omega, \tag{3}$$

where m is the mass of the multicopter and g denotes the gravitational acceleration constant. Let $u_{\mathrm{T}} \triangleq f_{\mathrm{T}}/m$ be the normalized thrust control input and $u_{\mathrm{R}} \triangleq \omega$ be the normalized rotational control input. Define the state vector

$$\mathbf{x} \triangleq [x,\ \dot{x},\ z,\ \dot{z},\ \theta]^{\mathrm{T}} \in \mathbb{R}^5, \tag{4}$$

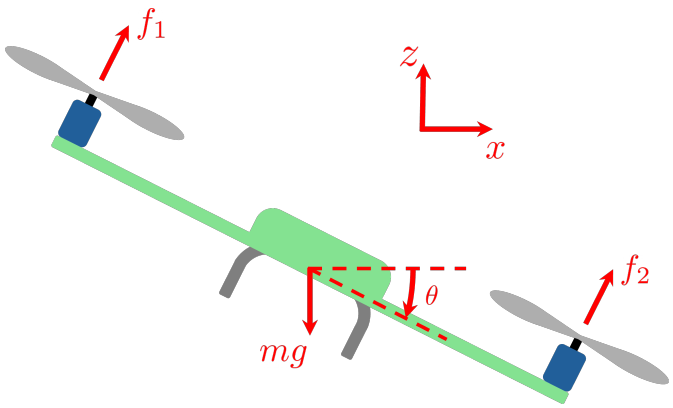

Figure 1: Multicopter lateral flight diagram.

and the input vector

$$\mathbf{u} \triangleq [u_{\mathrm{T}},\ u_{\mathrm{R}}]^{\mathrm{T}} \in \mathbb{R}^2. \tag{5}$$

Then the system dynamics (1),(2), (3) can be written as

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \dot{x} \\ u_{\mathrm{T}} \sin\theta \\ \dot{z} \\ u_{\mathrm{T}} \cos\theta - g \\ u_{\mathrm{R}} \end{bmatrix}. \tag{6}$$

Note that while [1] derives non-dimensional dynamics, this document uses the dynamics given by (6).

The model described by (6) is a simplified version of the full multicopter dynamics. In practical applications, thrust and pitch rate cannot be directly modulated since the dynamics of these depend on motor dynamics and the moment of inertia of the multicopter body, which would result in delayed changes. However, in the case of multicopters capable of very fast maneuvers, the changes in thrust and pitch rate are fast enough for this assumption to hold. Furthermore, several other effects, such as blade flapping and the drag of the multicopter body are neglected. While these effects are not negligible during fast maneuvers, these assumptions simplify the optimization problem and yield state and input trajectories that can be used for benchmarking purposes.

3 Derivation of the time-optimal input and state trajectories using Pontryagin's optimality conditions

In this section, the time-optimal input and state trajectories of the dynamic system introduced in Section 2 are derived using Pontryagin's optimality conditions. This section is organized as follows. Subsection 3.1 formulates the optimization problem whose solution yields the time-optimal input and state trajectories for a general nonlinear dynamical system. Subsection 3.2 introduces the conditions stated by Pontryagin's Minimum Principle (PMP) for the time-optimal input and state trajectories and applies it to the dynamics introduced in Section 2. Subsections 3.3 and 3.4 derive expressions for the optimal trajectories of u_{R} and u_{T}, respectively. Subsection 3.5 introduces the augmented system dynamics and a Boundary Value Problem (BVP), whose solution satisfies the PMP conditions and thus yields time-optimal input and state trajectories. Subsection 3.6 reformulates the BVP introduced in Section 3.5 to consider normalized time to simplify the optimization problem. Finally, Subsection 3.7 briefly explains the algorithm proposed in [1] to solve the BVP presented in Subsection 3.5. Note that most of the content in this section is rewritten from the formulation presented in [1].

3.1 Optimization problem formulation

For a general nonlinear dynamical system, let $\mathbf{x} \in \mathbb{R}^{\ell_{\mathrm{x}}}$, be the state, and $\mathbf{u} \in \mathbb{R}^{\ell_{\mathrm{u}}}$ be the input. Furthermore, for a given trajectory, let $T > 0$ be the final time, $\mathbf{x}_{\mathrm{i}} \in \mathbb{R}^{\ell_{\mathrm{x}}}$ be the initial state, and $\mathbf{x}_{\mathrm{f}} \in \mathbb{R}^{\ell_{\mathrm{x}}}$ be the final state. Then,

the time-optimal trajectory optimization problem can be formulated as

$$\min_{\mathbf{u}\in\mathbf{U}} T = \min_{\mathbf{u}\in\mathbf{U}} \int_0^T dt, \tag{7}$$

subject to

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}), \tag{8}$$
$$\mathbf{x}(0) = \mathbf{x}_{\mathrm{i}}, \tag{9}$$
$$\mathbf{x}(T) = \mathbf{x}_{\mathrm{f}}, \tag{10}$$

where $f\colon \mathbb{R}^{\ell_{\mathrm{x}}} \times \mathbb{R}^{\ell_{\mathrm{u}}} \to \mathbb{R}^{\ell_{\mathrm{x}}}$ is a function that encodes the nonlinear dynamics of the system, and $\mathbf{U} \subseteq \mathbb{R}^{\ell_{\mathrm{u}}}$ is a subset that encodes the input constraints. Note that, for the multicopter lateral flight dynamics shown in Section 2, $\ell_{\mathrm{x}} = 5$, $\ell_{\mathrm{u}} = 2$, f is given by (6), and $\mathbf{U} \triangleq \{\mathbf{u} \in \mathbb{R}^2\colon [1\ 0]\ \mathbf{u} \in [\underline{F}_{\mathrm{T}}/m, \overline{F}_{\mathrm{T}}/m],\ [0\ 1]\ \mathbf{u} \in [-\overline{\omega}, \overline{\omega}]\}$.

3.2 Pontryagin's Minimum Principle for time-optimal maneuvers

For all $t \in [0, T]$, let $\mathbf{p}(t) \triangleq [p_1(t),\ p_2(t),\ p_3(t),\ p_4(t),\ p_5(t)]^{\mathrm{T}} \in \mathbb{R}^5$ be the adjoint variable vector. Considering the dynamics given by (6), let the Hamiltonian function $H\colon \mathbb{R}^5 \times \mathbb{R}^2 \times \mathbb{R}^5 \to \mathbb{R}$ corresponding to the optimal-time trajectory optimization problem formulated in (7)–(10) be given by

$$H(\mathbf{x}, \mathbf{u}, \mathbf{p}) \triangleq 1 + \mathbf{p}^{\mathrm{T}} f(\mathbf{x}, \mathbf{u}) = p_0 + p_1\dot{x} + p_2 u_{\mathrm{T}} \sin\theta + p_3\dot{z} + p_4(u_{\mathrm{T}}\cos\theta - g) + p_5 u_{\mathrm{R}}. \tag{11}$$

Let $\mathbf{u}^* \triangleq \begin{bmatrix} u_{\mathrm{T}}^* & u_{\mathrm{R}}^* \end{bmatrix}^{\mathrm{T}} \in \mathbf{U} \subset \mathbb{R}^2$ and $\mathbf{x}^* \triangleq \begin{bmatrix} x^* & \dot{x}^* & z^* & \dot{z}^* & \theta^* \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^5$ be the time-optimal control and state trajectories obtained by solving the optimization problem given by (7)–(10), respectively, and suppose that, for all $t \in [0, T]$,

$$\dot{\mathbf{p}}(t) = -\nabla_{\mathbf{x}} H(\mathbf{x}^*(t), \mathbf{u}^*(t), \mathbf{p}(t)) = \begin{bmatrix} 0 \\ -p_1(t) \\ 0 \\ -p_3(t) \\ -p_2(t)u_{\mathrm{T}}^*(t)\cos\theta^*(t) + p_4(t)u_{\mathrm{T}}^*(t)\sin\theta^*(t) \end{bmatrix}, \tag{12}$$

which implies that

$$\dot{p}_5(t) = -p_2(t)u_{\mathrm{T}}^*(t)\cos\theta^*(t) + p_4(t)u_{\mathrm{T}}^*(t)\sin\theta^*(t), \tag{13}$$

and there exist $c_1, c_2, c_3, c_4 \in \mathbb{R}$ such that

$$p_1(t) = c_1, \tag{14}$$
$$p_2(t) = -c_1 t + c_2, \tag{15}$$
$$p_3(t) = c_3, \tag{16}$$
$$p_4(t) = -c_3 t + c_4, \tag{17}$$

For all $t \in [0, T]$, write $\mathbf{x}(t) = f_{\mathrm{x}}(\mathbf{x}(0), \mathbf{u}(t))$ and $\mathbf{p}(t) = f_{\mathrm{p}}(t, \mathbf{p}(0), \mathbf{x}(0), \mathbf{u}(t))$, where functions f_{x} and f_{p} encode the solutions of $\mathbf{x}(t)$ and $\mathbf{p}(t)$ given by $\mathbf{u}(t)$. Then, it follows from PMP that the time-optimal control trajectory $\mathbf{u}^*$ minimizes H, that is, for all $t \in [0, T]$,

$$\mathbf{u}^*(t) = \operatorname*{argmin}_{\mathbf{u}(t)\in\mathbf{U}} H(\mathbf{x}(t), \mathbf{u}(t), \mathbf{p}(t)) = \operatorname*{argmin}_{\mathbf{u}\in\mathbf{U}} H(f_{\mathrm{x}}(\mathbf{x}(0), \mathbf{u}(t)), \mathbf{u}(t), f_{\mathrm{p}}(t, \mathbf{p}(0), \mathbf{x}(0), \mathbf{u}(t))). \tag{18}$$

Furthermore, since the terminal time is free (T is not fixed), H is always zero along the optimal trajectory, that is, for all $t \in [0, T]$,

$$H(\mathbf{x}^*(t), \mathbf{u}^*(t), \mathbf{p}(t)) = 0. \tag{19}$$

Since $\frac{\partial}{\partial u_{\mathrm{T}} \partial u_{\mathrm{R}}} H(\mathbf{x}, \mathbf{u}, \mathbf{p}) = \frac{\partial}{\partial u_{\mathrm{R}} \partial u_{\mathrm{T}}} H(\mathbf{x}, \mathbf{u}, \mathbf{p}) = 0$, the Hamiltonian can be minimized separately for u_{T} and u_{R}.

3.3 Derivation of optimal control input u_{R}^*

Minimizing (11) with respect to u_{R} yields

$$u_{\mathrm{R}}^* = \underset{u_{\mathrm{R}} \in [-\overline{\omega}, \overline{\omega}]}{\operatorname{argmin}} \; p_5 u_{\mathrm{R}}. \tag{20}$$

In the case where $p_5 \neq 0$, it follows from (20) that

$$u_{\mathrm{R}}^* = -\overline{\omega} \operatorname{sign}(p_5). \tag{21}$$

Suppose there exist $t_1, t_2 \in [0, T]$ such that $t_1 < t_2$ and, for all $t \in [t_1, t_2]$, $p_5(t) = 0$, that is, a singular arc occurs for the nontrivial interval of time $[t_1, t_2]$. This in turn, implies that, for all $t \in [t_1, t_2]$, $\dot{p}_5(t) = 0$, and thus, it follows from (13) that

$$-p_2 u_{\mathrm{T}}^* \cos\theta^* + p_4 u_{\mathrm{T}}^* \sin\theta^* = 0. \tag{22}$$

Since $u_{\mathrm{T}}^* > 0$, it follows from (15), (17) and (22) that, for all $t \in [t_1, t_2]$,

$$\theta^* = \arctan\left(\frac{p_2}{p_4}\right) = \arctan\left(\frac{-c_1 t + c_2}{-c_3 t + c_4}\right). \tag{23}$$

Taking the time derivative of (23) yields

$$\dot{\theta}^* = u_{\mathrm{R}}^* = \frac{c_2 c_3 - c_1 c_4}{(c_1^2 + c_3^2)t^2 - 2(c_1 c_2 + c_3 c_4)t + c_2^2 + c_4^2} \tag{24}$$

in the case where $p_5 = 0$. Define

$$u_{\mathrm{R,sing}}^* = \frac{c_2 c_3 - c_1 c_4}{(c_1^2 + c_3^2)t^2 - 2(c_1 c_2 + c_3 c_4)t + c_2^2 + c_4^2}.$$

Then, it folows from (21) and (24) that u_{R}^* is given by

$$u_{\mathrm{R}}^* = \begin{cases} -\overline{\omega} \operatorname{sign}(p_5), & p_5 \neq 0, \\ u_{\mathrm{R,sing}}^*, & p_5 = 0. \end{cases} \tag{25}$$

3.4 Derivation of optimal control input u_{T}^*

Minimizing (11) with respect to u_{T} yields

$$u_{\mathrm{T}}^* = \underset{u_{\mathrm{T}} \in [\underline{F}_{\mathrm{T}}/m, \overline{F}_{\mathrm{T}}/m]}{\operatorname{argmin}} \; (p_2 \sin\theta^* + p_4 \cos\theta^*) u_{\mathrm{T}}. \tag{26}$$

Define $p_{\mathrm{T}} \triangleq p_2 \sin\theta^* + p_4 \cos\theta^* = (-c_1 t + +c_2)\sin\theta^* + (-c_3 t + +c_4)\cos\theta^*$. In the case where $p_{\mathrm{T}} \neq 0$, it follows from (26) that

$$u_{\mathrm{T}}^* = \begin{cases} \overline{F}_{\mathrm{T}}/m, & p_{\mathrm{T}} < 0, \\ \underline{F}_{\mathrm{T}}/m, & p_{\mathrm{T}} > 0. \end{cases} \tag{27}$$

Suppose there exist $t_1, t_2 \in [0, T]$ such that $t_1 < t_2$ and, for all $t \in [t_1, t_2]$, $p_{\mathrm{T}}(t) = 0$, that is, a singular arc occurs for the nontrivial interval of time $[t_1, t_2]$. Then, it follows from (15) and (17) that, for all $t \in [t_1, t_2]$,

$$\theta^* = \arctan\left(-\frac{p_4}{p_2}\right) = \arctan\left(\frac{c_3 t - c_4}{-c_1 t + c_2}\right). \tag{28}$$

Since $u_{\mathrm{R}}^* = \dot{\theta}^*$, it follows from (25) that, in the case where $p_5 \neq 0$, $\dot{\theta}^* = -\overline{\omega} \operatorname{sign}(p_5)$, which implies that θ^* is piecewise-affine over t, which contradicts (28). In the case where $p_5 = 0$, it follows from (23) and (28) that

$$\theta^* = \arctan\left(\frac{-c_1 t + c_2}{-c_3 t + c_4}\right) = \arctan\left(\frac{c_3 t - c_4}{-c_1 t + c_2}\right), \tag{29}$$

which implies that

$$(c_1^2 + c_3^2)t^2 - 2(c_1 c_2 + c_3 c_4)t + c_2^2 + c_4^2 = 0. \tag{30}$$

The equality shown in (30) only holds for all $t \in [t_1, t_2]$ in the case where $c_1 = c_2 = c_3 = c_4 = 0$, which results in a contradiction since (23) and (28) become indefinite. Hence, no singular arc occurs for a nontrivial interval of time and it follows from (27) that u_{T}^* can be written as

$$u_{\mathrm{T}}^* = \begin{cases} \overline{F}_{\mathrm{T}}/m, & p_{\mathrm{T}} \leq 0, \\ \underline{F}_{\mathrm{T}}/m, & p_{\mathrm{T}} > 0. \end{cases} \tag{31}$$

3.5 Augmented system and boundary value problem

The system of equations (6) is augmented to include the dynamics of p_5 since it is the only variable that does not have an explicit solution in time. Define the augmented state vector $\mathbf{x}_\mathrm{a} \triangleq [\mathbf{x}^*,\ p_5]^\mathrm{T}$. Then, it follows from (6), (13), (15), (17), (25), and (31) that the augmented system dynamics are given by

$$\dot{\mathbf{x}}_\mathrm{a} = f_\mathrm{a}(t, \mathbf{x}_\mathrm{a}) = \begin{bmatrix} \dot{x}^* \\ u_\mathrm{T}^* \sin\theta^* \\ \dot{z}^* \\ u_\mathrm{T}^* \cos\theta^* - g \\ u_\mathrm{T}^* \\ (c_1 t - c_2) u_\mathrm{T}^* \cos\theta^* + (c_4 - c_3 t) u_\mathrm{T}^* \sin\theta^* \end{bmatrix}, \tag{32}$$

where u_T^* and u_R^* are given by

$$u_\mathrm{T}^* = \begin{cases} \overline{F}_\mathrm{T}/m, & (-c_1 t + + c_2)\sin\theta^* + (-c_3 t + + c_4)\cos\theta^* \leq 0, \\ \underline{F}_\mathrm{T}/m, & (-c_1 t + + c_2)\sin\theta^* + (-c_3 t + + c_4)\cos\theta^* > 0, \end{cases} \tag{33}$$

$$u_\mathrm{R}^* = \begin{cases} -\overline{\omega}\ \mathrm{sign}(p_5), & p_5 \neq 0, \\ \frac{c_2 c_3 - c_1 c_4}{(c_1^2 + c_3^2)t^2 - 2(c_1 c_2 + c_3 c_4)t + c_2^2 + c_4^2}, & p_5 = 0. \end{cases} \tag{34}$$

Hence, a multicopter maneuver from $\mathbf{x}_\mathrm{i}$ to $\mathbf{x}_\mathrm{f}$ that satisfies the PMP conditions solves the following BVP:

$$\dot{\mathbf{x}}_\mathrm{a} = f_\mathrm{a}(t, \mathbf{x}_\mathrm{a}), \tag{35}$$

$$\mathbf{x}^*(0) = \mathbf{x}_\mathrm{i}, \tag{36}$$

$$\mathbf{x}^*(T) = \mathbf{x}_\mathrm{f}. \tag{37}$$

The BVP given by (35), (36), (37) has 6 unknown variables: the final time T, $p_5(0)$, and the four unknown constants c_1, c_2, c_3, c_4. Note that the value of $p_5(0)$ has to meet the constraint given by (19) with $t = 0$.

3.6 Augmented system and boundary value problem with normalized time

The BVP formulated in Subsection 3.5 may require several iterations over the final time T during the numerical optimization procedure. An alternative formulation is introduced by replacing the time variable t with the normalized time variable $t_\mathrm{n} \triangleq t/T$. Defining the new augmented state $\mathbf{x}_\mathrm{a,n} \triangleq [\mathbf{x}_\mathrm{a},\ T]^\mathrm{T}$, the augmented system dynamics over t_n are given by

$$\frac{\mathrm{d}\mathbf{x}_\mathrm{a,n}}{\mathrm{d}t_\mathrm{n}} = \frac{\mathrm{d}\mathbf{x}_\mathrm{a,n}}{\mathrm{d}t}\frac{\mathrm{d}t}{\mathrm{d}t_\mathrm{n}} = T\ \frac{\mathrm{d}\mathbf{x}_\mathrm{a,n}}{\mathrm{d}t} = f_\mathrm{a,n}(t_\mathrm{n}, \mathbf{x}_\mathrm{a,n}) = T \begin{bmatrix} \dot{x}^* \\ u_\mathrm{T}^* \sin\theta^* \\ \dot{z}^* \\ u_\mathrm{T}^* \cos\theta^* - g \\ u_\mathrm{T}^* \\ (c_1 T t_\mathrm{n} - c_2) u_\mathrm{T}^* \cos\theta^* + (c_4 - c_3 T t_\mathrm{n}) u_\mathrm{T}^* \sin\theta^* \\ 0 \end{bmatrix}, \tag{38}$$

where u_T^* and u_R^* are given by (33) and (34) with $t = T t_\mathrm{n}$, respectively. Hence, the new BVP is given by (36), (37), (38) and must be solved using the same initial and final states for $t_\mathrm{n} \in [0, 1]$. Since T is an element of the optimization state vector, this BVP may be easier to solve than the BVP given in the previous subsection. The BVP introduced in this subsection is not solved in this document, although a similar idea is used to formulate the optimization problem in Section 5.

3.7 Algorithm for calculation of time-optimal maneuvers

In [1], an algorithm is proposed to solve the BVP given by (35), (36), (37) and presented in Subsection 3.5. A brief description of this algorithm will be given in this subsection.

The first steps of the algorithm focus on obtaining an initial guess of the optimal trajectory to minimize the time the BVP solver takes to obtain a solution. The methodology used depends on whether the user chooses to allow

bang-bang maneuvers (no singular arcs) or bang-singular maneuvers. In this document, bang-singular maneuvers are chosen. The trajectory is initially optimized using Switching Time Optimization (STO), which exploits the structure of the control inputs shown in (33) and (34) and parametrizes the optimal trajectory using the terminal time, the switching times between the possible control inputs and the time spent on a singular arc. The weighted sum of the square errors at the terminal time of the trajectory is used by STO as an objective function. Note that the result from this optimization may not yield an optimal trajectory in the sense of PMP. Since the singular arc control input depends on c_1, c_2, c_3 and c_4, these parameters must also be optimized by STO. Furthermore, this implementation of STO requires a guess of the terminal time to iterate through possible terminal times while performing optimization. Once a terminal time iteration yields a terminal state close enough to the target final state, the BVP is solved using the trajectory optimized by the STO as an initial guess. The full code for algorithm implementation in Matlab is available in `https://link.springer.com/article/10.1007/s10514-012-9282-3`. More details are available in the Appendix of [1].

4 Numerical optimization results using Switching Time Optimization and solving the Boundary Value Problem

In this section, the STO algorithm is used to obtain the optimal state and input trajectories to perform a vertical displacement with a flip under the multicopter flight dynamics shown in Section 2. The results in this section are obtained using the code available in `https://link.springer.com/article/10.1007/s10514-012-9282-3`, which are similar to those shown in [1].

4.1 Vertical displacement with flip

The optimal trajectories for a vertical displacement with flip are obtained by solving an optimal-time problem with initial conditions

$$x(0) = 0 \text{ m}, \dot{x}(0) = 0 \text{ m/s}, z(0) = 0 \text{ m}, \dot{z}(0) = 0 \text{ m/s}, \theta(0) = 0 \text{ rad}, \tag{39}$$

and final conditions

$$x(T) = 0 \text{ m}, \dot{x}(T) = 0 \text{ m/s}, z(T) = 2.7 \text{ m}, \dot{z}(T) = 0 \text{ m/s}, \theta(T) = 2\pi \text{ rad}, \tag{40}$$

which encodes the objective of moving 2.7 m upwards and performing a flip in the middle of the trajectory. Note that these conditions do not explicitly state at which point along the trajectory the multicopter needs to perform the flip, so the optimization algorithm needs to determine this. For the considered multicopter dynamics, we consider $g = 9.81 m/s^2$, $\underline{F}_{\mathrm{T}}/m = 1$ m/s^2, $\overline{F}_{\mathrm{T}}/m = 20$ m/s^2, and $\overline{\omega} = 10$ rad/s.

The results from implementing STO to obtain an initial solution and subsequently solving the BVP given by (35), (36), (37) are shown in Figures 2 and 3, which show the obtained optimal state and input trajectories, respectively, with a final time of $T = 1.0499$ s. Figure 2 shows that the resulting optimal trajectory meets the requested initial and final conditions for all states. Furthermore, Figure 3 shows that $u_{\mathrm{T}}^* \in \{\underline{F}_{\mathrm{T}}/m,\ \overline{F}_{\mathrm{T}}/m\}$ and $u_{\mathrm{R}}^* \in \{-\underline{\omega},\ \overline{\omega}, 0\}$, since the solver yields $c_2 c_3 - c_1 c_4 = 0$, as shown in (33) and (34), respectively.

5 Time-optimal trajectory planning using nonlinear programming and discretized nonlinear dynamics

In this section, a NLP is formulated to solve the time-optimal trajectory planning problem considering a general nonlinear system. Hence, the states, inputs and dynamics considered in this section do not necessarily correspond to a multicopter.

Let $\mathbf{x} \in \mathbb{R}^{\ell_{\mathrm{x}}}$ be the state vector and $\mathbf{u} \in \mathbb{R}^{\ell_{\mathrm{u}}}$ be the input vector. Then, suppose the continuous-time, nonlinear dynamics of the system are given by

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}), \tag{41}$$

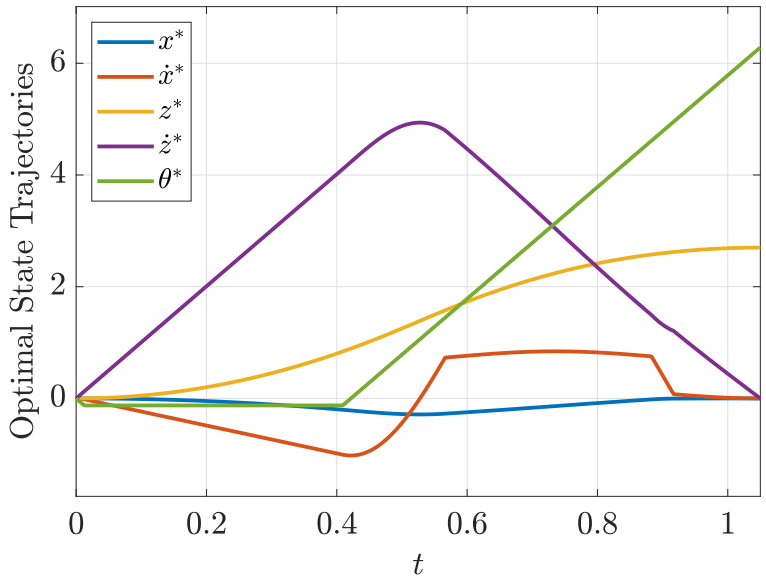

Figure 2: Optimal state trajectories for vertical displacement with flip maneuver considering the multicopter lateral flight dynamics shown in Section 2 obtained by using STO with $z(T) = 2.7$ m and $\theta(T) = 2\pi$ rad.

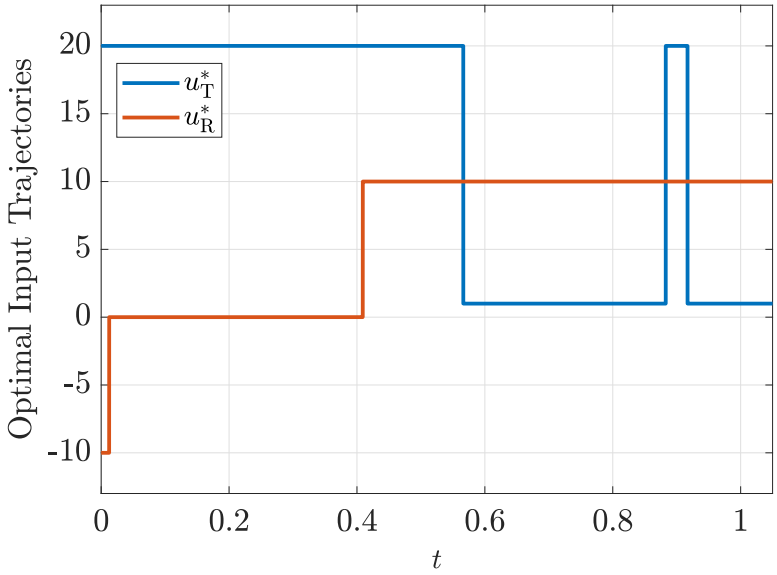

Figure 3: Optimal input trajectories for vertical displacement with flip maneuver considering the multicopter lateral flight dynamics shown in Section 2 obtained by using STO with $z(T) = 2.7$ m and $\theta(T) = 2\pi$ rad. The constraint parameters are given by $\underline{F}_{\mathrm{T}}/m = 1$ m/s^2, $\overline{F}_{\mathrm{T}}/m = 20$ m/s^2, and $\overline{\omega} = 10$ rad/s.

where $f\colon \mathbb{R}^{\ell_{\mathrm{x}}} \times \mathbb{R}^{\ell_{\mathrm{x}}} \to \mathbb{R}^{\ell_{\mathrm{x}}}$. Next, consider the discrete-time, nonlinear dynamics resulting from discretizing the system shown in (41) using the Euler-method given by

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \; f(\mathbf{x}_k, \mathbf{u}_k), \tag{42}$$

where $k \in \{0, 1, \ldots\}$ is the iteration step, $\Delta t > 0$ is the time between iterations, and $\mathbf{x}_k \in \mathbb{R}^{\ell_{\mathrm{x}}}, \mathbf{u}_k \in \mathbb{R}^{\ell_{\mathrm{u}}}$ are the discrete-time state and input vectors, respectively, at iteration step k, which corresponds to the time $t = k\Delta t$. For all $i \in \mathbb{R}^{\ell_{\mathrm{x}}}$, let $\mathbf{x}_{k,(i)}$ be the i-th component of $\mathbf{x}_k$, and, for all $j \in \mathbb{R}^{\ell_{\mathrm{u}}}$, let $\mathbf{u}_{k,(j)}$ be the j-th component of $\mathbf{u}_k$. Then, suppose that, for all $k \geq 0$, $\mathbf{x}_k$ and $\mathbf{u}_k$ are constrained, such that, for all $i \in \{1, \ldots, \ell_{\mathrm{x}}\}$, let $\mathbf{x}_{k,(i)} \in [\mathbf{x}_{(i),\min},\ \mathbf{x}_{(i),\max}]$, where $\mathbf{x}_{(i),\min}, \mathbf{x}_{(i),\max} \in \mathbb{R}$ denote the lower and upper bounds of the i-th component of $\mathbf{x}_k$, and , for all $j \in \{1, \ldots, \ell_{\mathrm{u}}\}$, let $\mathbf{u}_{k,(j)} \in [\mathbf{u}_{(j),\min},\ \mathbf{u}_{(j),\max}]$, where $\mathbf{u}_{(j),\min}, \mathbf{u}_{(j),\max} \in \mathbb{R}$ denote the lower

and upper bounds of the j-th component of $\mathbf{u}_k$.

For a given trajectory, let $T > 0$ be the final time, $\mathbf{x}_\mathrm{i} \in \mathbb{R}^{\ell_\mathrm{x}}$ be the initial state, $\mathbf{x}_\mathrm{f} \in \mathbb{R}^{\ell_\mathrm{x}}$ be the final state, and $\mathbf{u}_\mathrm{f} \in \mathbb{R}^{\ell_\mathrm{u}}$ be the final input. Then, for a user-chosen maximum number of iteration steps, the time-optimal trajectory optimization problem can be formulated using the discretized dynamics shown in (42) as

$$\min_{(\mathbf{x}_k)_{k=0}^{N},(\mathbf{u}_k)_{k=0}^{N-1},T} T + (\mathbf{x}_\mathrm{f} - \mathbf{x}_N)^\mathrm{T} Q_\mathrm{x} (\mathbf{x}_\mathrm{f} - \mathbf{x}_N) + (\mathbf{u}_\mathrm{f} - \mathbf{u}_{N-1})^\mathrm{T} Q_\mathrm{u} (\mathbf{u}_\mathrm{f} - \mathbf{u}_{N-1}), \tag{43}$$

subject to

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{T}{N} f(\mathbf{x}_k, \mathbf{u}_k) \text{ for all } k \in \{0, \ldots, N-1\}, \tag{44}$$

$$\mathbf{x}_0 = \mathbf{x}_\mathrm{i}, \tag{45}$$

$$\mathbf{x}_N = \mathbf{x}_\mathrm{f}, \tag{46}$$

$$\mathbf{x}_{k,(i)} \in [\mathbf{x}_{(i),\min},\ \mathbf{x}_{(i),\max}] \text{ for all } k \in \{0, \ldots, N\},\ \text{for all } i \in \{1, \ldots, \ell_\mathrm{x}\} \tag{47}$$

$$\mathbf{u}_{k,(j)} \in [\mathbf{u}_{(j),\min},\ \mathbf{u}_{(j),\max}] \text{ for all } k \in \{0, \ldots, N-1\},\ \text{for all } j \in \{1, \ldots, \ell_\mathrm{u}\}, \tag{48}$$

where $Q_\mathrm{x} \in \mathbb{R}^{\ell_\mathrm{x} \times \ell_\mathrm{x}}$ is the final state error weighting matrix and $Q_\mathrm{u} \in \mathbb{R}^{\ell_\mathrm{u} \times \ell_\mathrm{u}}$ is the final input error weighting matrix While it is redundant to include a final state error term in the cost function and stating the final state equality constraint (46), in practice, this arrangement increases the likelihood of the solver reaching the final state values; similarly, the final input error term in the cost function increases the likelihood of the solver reaching the final input values. Note that the optimization variables of the NLP given by (43)–(48) are the final time T, the discrete-time trajectory states $\mathbf{x}_0, \mathbf{x}_1 \ldots, \mathbf{x}_N$, and the discrete-time input states $\mathbf{u}_0, \mathbf{u}_1 \ldots, \mathbf{u}_{N-1}$. The number of discrete-time iterations N is chosen by the user and is fixed throughout the optimization procedure, which implies that T determines the time between iterations $\Delta t = T/N$. Hence, increasing N may yield smaller values of Δt, which may allow smoother optimal trajectories to be obtained and to increase the feasibility of solving the NLP. This approach is similar to the one shown in Subsection 3.6 since fixing the final time in continuous-time is analogous to fixing the number of iteration in discrete-time.

6 Numerical optimization results by solving a nonlinear program with CasADi

In this section, the NLP shown in Section 5 is solved to obtain the optimal state and input trajectories to perform a vertical displacement with a flip under the multicopter flight dynamics shown in Section 2 and the results are compared to those obtained in Section 4 using the STO algorithm. Furthermore, the NLP is also solved for a maneuver consisting of a horizontal displacement with a flip. The NLP is solved in the program `Trajectory_Optimization_for_Multicopter.m`, which uses the CasADi software toolbox [2].

6.1 Vertical displacement with flip

The optimal trajectories for a vertical displacement with flip are obtained by solving an optimal-time problem with initial conditions

$$\mathbf{x}_\mathrm{i} = \begin{bmatrix} x(0) & \dot{x}(0) & z(0) & \dot{z}(0) & \theta(0) \end{bmatrix}^\mathrm{T} = 0, \tag{49}$$

and final conditions

$$\mathbf{x}_\mathrm{f} = \begin{bmatrix} x(T) & \dot{x}(T) & z(T) & \dot{z}(T) & \theta(T) \end{bmatrix}^\mathrm{T} = \begin{bmatrix} 0 & 0 & 2.7 & 0 & 2\pi \end{bmatrix}^\mathrm{T}, \tag{50}$$

which encodes the objective of moving 2.7 m upwards and performing a flip in the middle of the trajectory. For the considered multicopter dynamics, we consider $g = 9.81 m/s^2$, and, for all $k \geq 0$, the following input constraints

$$\mathbf{u}_{k,(1)} = u_{\mathrm{T},k} \in [\underline{F}_\mathrm{T}/m,\ \overline{F}_\mathrm{T}/m] = [1,\ 20] \text{ m/s}^2, \tag{51}$$

$$\mathbf{u}_{k,(2)} = u_{\mathrm{R},k} \in [-\overline{\omega},\ \overline{\omega}] = [-10,\ 10] \text{ rad/s}. \tag{52}$$

No state constraints are considered for this maneuver. For the NLP setup, $N = 200$, $Q_\mathrm{x} = 100 I_5$, and $Q_\mathrm{u} = 0$, that is, no final input error cost is considered.

The results from solving the NLP given by (43)–(48) are shown in Figures 4 and 5, which show the obtained optimal state and input trajectories, respectively, with a final time of $T = 1.0477$ s. Figure 4 shows that the resulting optimal trajectory meets the requested initial and final conditions for all states, and Figure 5 shows that the input constraints are met by the optimal trajectory. Note that the results shown in Figures 4 and 5 are very similar to those shown in Figures 2 and 3, respectively, which validates the NLP approach. Finally, the optimal trajectory is illustrated in Figure 6.

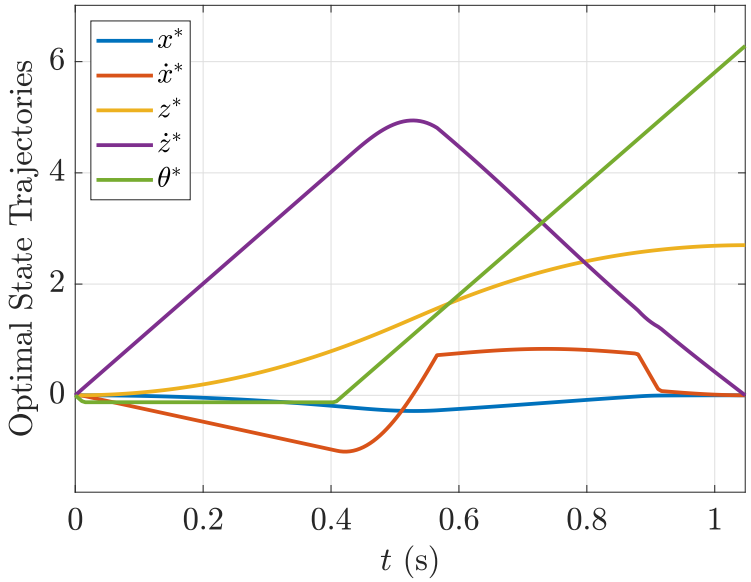

Figure 4: Optimal state trajectories for vertical displacement with flip maneuver considering the multicopter lateral flight dynamics shown in Section 2 obtained by solving the NLP shown in Section 5 with $z(T) = 2.7$ m and $\theta(T) = 2\pi$ rad, with a final time of $T = 1.0477$ s.

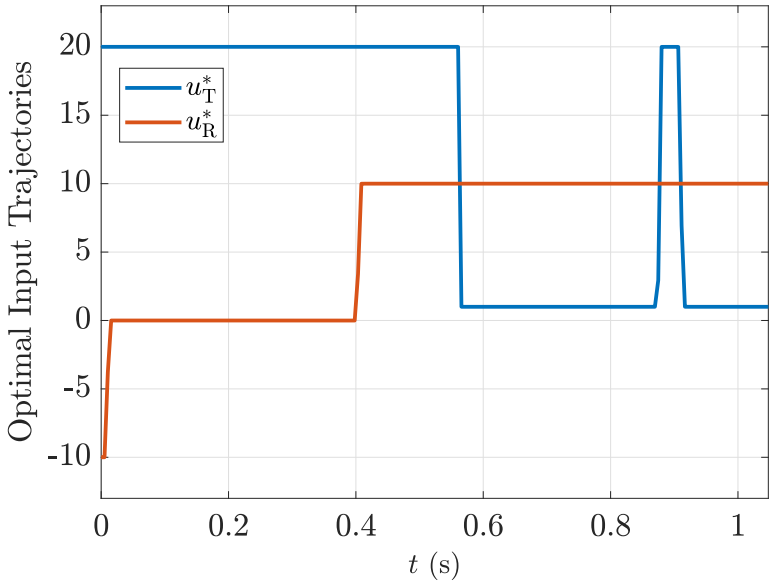

Figure 5: Optimal input trajectories for vertical displacement with flip maneuver considering the multicopter lateral flight dynamics shown in Section 2 obtained by solving the NLP shown in Section 5 with $z(T) = 2.7$ m and $\theta(T) = 2\pi$ rad, with a final time of $T = 1.0477$ s. The constraint parameters are given by $\underline{F}_{\mathrm{T}}/m = 1$ m/s^2, $\overline{F}_{\mathrm{T}}/m = 20$ m/s^2, and $\overline{\omega} = 10$ rad/s.

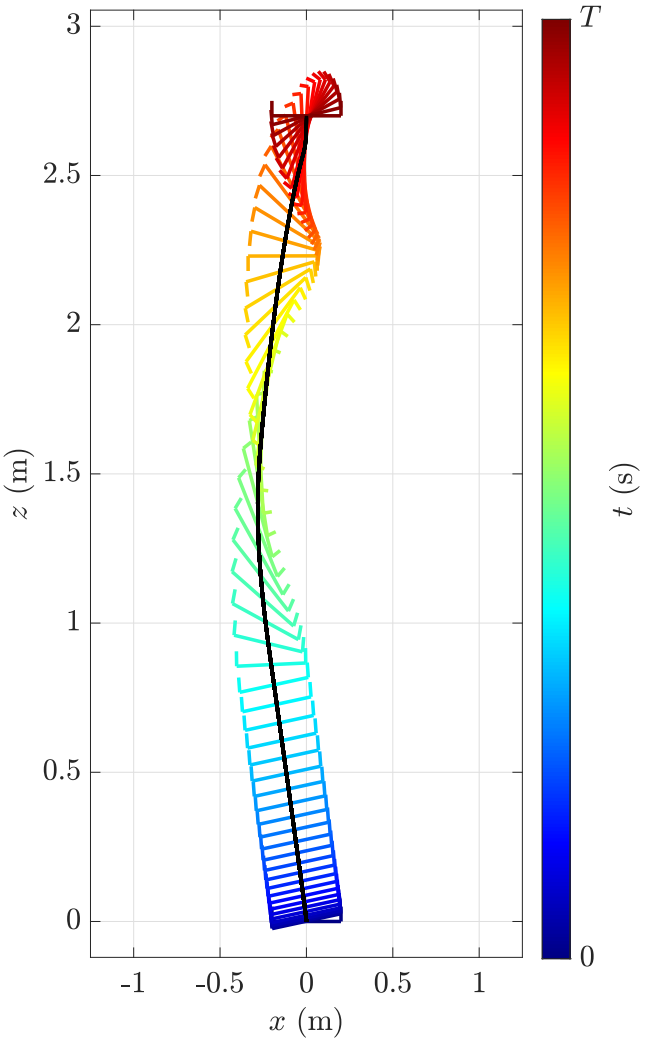

Figure 6: Illustration of the optimal trajectory for vertical displacement with flip maneuver considering the multicopter lateral flight dynamics shown in Section 2 obtained by solving the NLP shown in Section 5 with $z(T) = 2.7$ m and $\theta(T) = 2\pi$ rad, with a final time of $T = 1.0477$ s. The illustration is created using the `Create_Multicopter_vertical_motion_animation.m` program.

6.2 Horizontal displacement with flip

The optimal trajectories for a horizontal displacement with flip are obtained by solving an optimal-time problem with initial state conditions

$$\mathbf{x}_{\mathrm{i}} = \begin{bmatrix} x(0) & \dot{x}(0) & z(0) & \dot{z}(0) & \theta(0) \end{bmatrix}^{\mathrm{T}} = 0, \tag{53}$$

and final state conditions

$$\mathbf{x}_{\mathrm{f}} = \begin{bmatrix} x(T) & \dot{x}(T) & z(T) & \dot{z}(T) & \theta(T) \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} 0 & 0 & 12 & 0 & 2\pi \end{bmatrix}^{\mathrm{T}}, \tag{54}$$

which encodes the objective of moving 12 m rightwards and performing a flip in the middle of the trajectory. For the considered multicopter dynamics, we consider $g = 9.81 m/s^2$, and, for all $k \geq 0$, the following input constraints

$$\mathbf{u}_{k,(1)} = u_{\mathrm{T},k} \in [\underline{F}_{\mathrm{T}}/m,\ \overline{F}_{\mathrm{T}}/m] = [1,\ 20], \tag{55}$$

$$\mathbf{u}_{k,(2)} = u_{\mathrm{R},k} \in [-\overline{\omega},\ \overline{\omega}] = [-10,\ 10] \text{ rad/s}. \tag{56}$$

No state constraints are considered for this maneuver. For the NLP setup, $N = 200$, $Q_{\mathrm{x}} = 100 I_5$, and $Q_{\mathrm{u}} = 0$, that is, no final input error cost is considered.

The results from solving the NLP given by (43)–(48) are shown in Figures 7 and 8, which show the obtained optimal state and input trajectories, respectively, with a final time of $T = 1.8132$ s. Figure 7 shows that the resulting optimal trajectory meets the requested initial and final conditions for all states, and Figure 8 shows that the input constraints are met by the optimal trajectory. Finally, the optimal trajectory is illustrated in Figure 9.

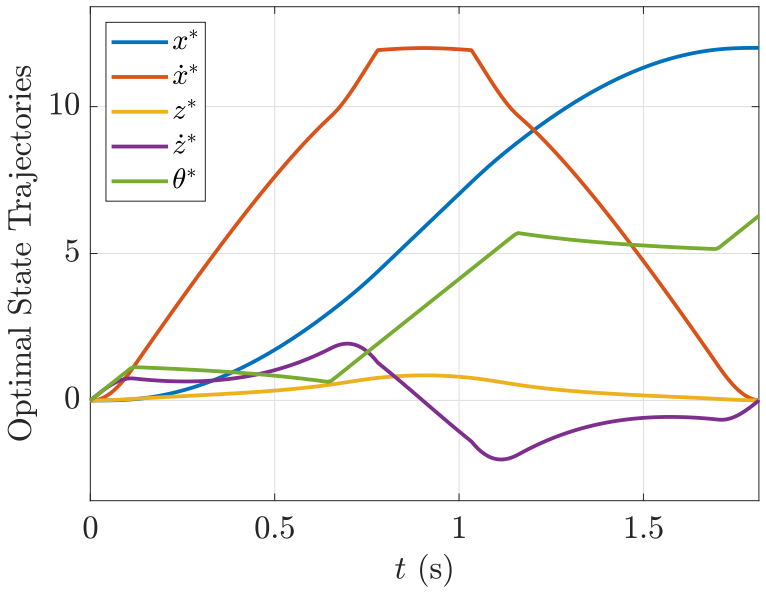

Figure 7: Optimal state trajectories for horizontal displacement with flip maneuver considering the multicopter lateral flight dynamics shown in Section 2 obtained by solving the NLP shown in Section 5 with $x(T) = 12$ m and $\theta(T) = 2\pi$ rad, with a final time of $T = 1.8132$ s.

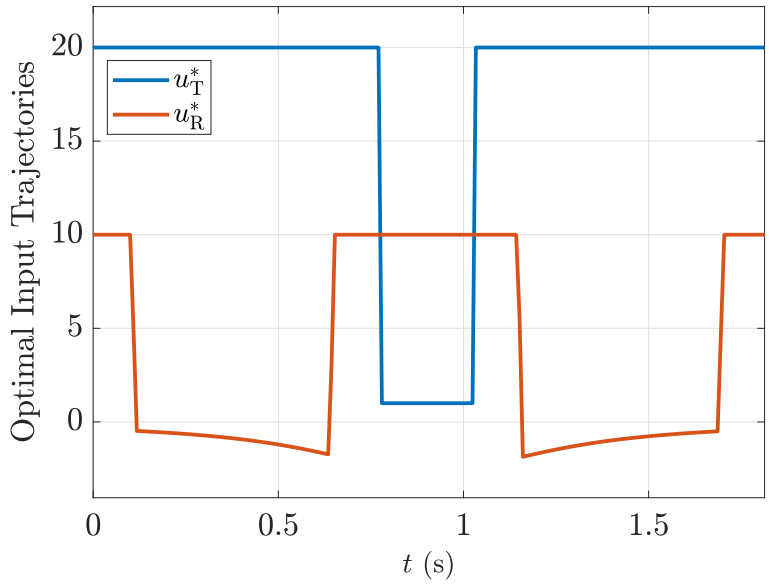

Figure 8: Optimal input trajectories for horizontal displacement with flip maneuver considering the multicopter lateral flight dynamics shown in Section 2 obtained by solving the NLP shown in Section 5 with $x(T) = 12$ m and $\theta(T) = 2\pi$ rad, with a final time of $T = 1.8132$ s. The constraint parameters are given by $\underline{F}_{\mathrm{T}}/m = 1$ m/s^2, $\overline{F}_{\mathrm{T}}/m = 20$ m/s^2, and $\overline{\omega} = 10$ rad/s.

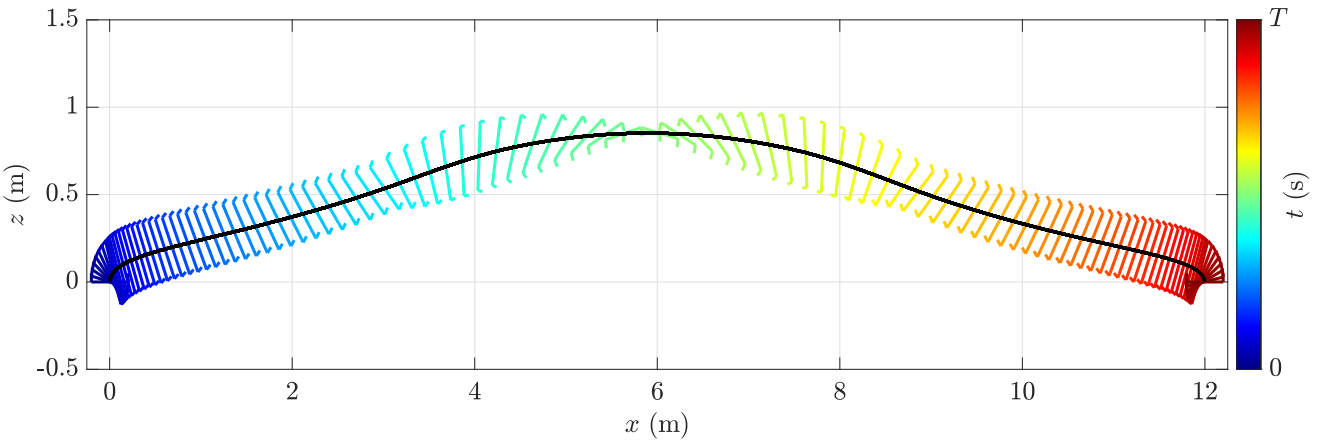

Figure 9: Illustration of optimal trajectory for horizontal displacement with flip maneuver considering the multicopter lateral flight dynamics shown in Section 2 obtained by solving the NLP shown in Section 5 with $x(T) = 12$ m and $\theta(T) = 2\pi$ rad, with a final time of $T = 1.8132$ s. The illustration is created using the `Create_Multicopter_horizontal_motion_animation.m` program.

7 Multicopter lateral flight dynamics with rotational dynamics

Consider the case in which the pitch rate ω cannot be directly modulated. Let $d > 0$ be the distance between the rotors and let $\tau \triangleq \frac{f_1 - f_2}{d} \in \mathbb{R}$ be the moment exerted by the rotors on the multicopter. Furthermore, suppose that τ is constrained, such that $\tau \in [-\overline{\tau}, \overline{\tau}]$, where $\overline{\tau} > 0$ represents the maximum thrust difference between f_1 and f_2. Then, the dynamics for the lateral flight of a multicopter shown in (1),(2), (3) can be rewritten as

$$\ddot{x} = \frac{f_{\mathrm{T}}}{m} \sin\theta, \tag{57}$$

$$\ddot{z} = \frac{f_{\mathrm{T}}}{m} \cos\theta - g, \tag{58}$$

$$\ddot{\theta} = \frac{\tau}{J_{\mathrm{I}}}, \tag{59}$$

where J_{I} is the moment of lateral inertia of the multicopter in the direction of the axis perpendicular to the plane of flight. Redefine $u_{\mathrm{R}} \triangleq \tau / J_{\mathrm{I}}$ as the normalized rotational control input, and the state vector

$$\mathbf{x} \triangleq [x,\ \dot{x},\ z,\ \dot{z},\ \theta,\ \dot{\theta}]^{\mathrm{T}} \in \mathbb{R}^6 \tag{60}$$

Then the system dynamics (57), (58), (59) can be written as

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \dot{x} \\ u_{\mathrm{T}} \sin\theta \\ \dot{z} \\ u_{\mathrm{T}} \cos\theta - g \\ \dot{\theta} \\ u_{\mathrm{R}} \end{bmatrix}. \tag{61}$$

Note that (61) can replace (6) in the NLP formulation shown in (43)–(48), although the problem becomes more complex due to the inclusion of the rotational dynamics.

8 Numerical optimization results by solving a nonlinear program with CasADi and considering rotational dynamics

In this section, the NLP shown in Section 5 is solved to obtain the optimal state and input trajectories to perform a vertical displacement with a flip and a horizontal displacement with a flip, both under the multicopter flight dynamics shown in Section 7. The NLP is solved in the program `Trajectory_Optimization_for_Multicopter_w_Rot_Dyn.m`, which uses the CasADi software toolbox [2].

8.1 Vertical displacement with flip

The optimal trajectories for a vertical displacement with flip are obtained by solving an optimal-time problem with initial state conditions

$$\mathbf{x}_{\mathrm{i}} = \begin{bmatrix} x(0) & \dot{x}(0) & z(0) & \dot{z}(0) & \theta(0) & \dot{\theta}(0) \end{bmatrix}^{\mathrm{T}} = 0, \tag{62}$$

and final state conditions

$$\mathbf{x}_{\mathrm{f}} = \begin{bmatrix} x(T) & \dot{x}(T) & z(T) & \dot{z}(T) & \theta(T) & \dot{\theta}(T) \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} 0 & 0 & 3 & 0 & 2\pi & 0 \end{bmatrix}^{\mathrm{T}}, \tag{63}$$

which encodes the objective of moving 3 m upwards and performing a flip in the middle of the trajectory. Furthermore, the following final input conditions are considered

$$\mathbf{u}_{\mathrm{f}} = \begin{bmatrix} u_{\mathrm{T}}(T) & u_{\mathrm{R}}(T) \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} g & 0 \end{bmatrix}^{\mathrm{T}}, \tag{64}$$

which, along the final state conditions, encode the requirement of the multicopter to be in hover conditions at the end of the trajectory. For the considered multicopter dynamics, we consider $g = 9.81m/s^2$, and, for all $k \geq 0$, the following input constraints

$$\mathbf{u}_{k,(1)} = u_{\mathrm{T},k} \in [\underline{F}_{\mathrm{T}}/m,\ \overline{F}_{\mathrm{T}}/m] = [1,\ 20]\ \mathrm{m/s^2}, \tag{65}$$

$$\mathbf{u}_{k,(2)} = u_{\mathrm{R},k} \in [-\overline{\tau}/J_{\mathrm{I}},\ \overline{\tau}/J_{\mathrm{I}}] = [-15,\ 15]\ \mathrm{rad/s^2}. \tag{66}$$

and the following state constraints

$$\mathbf{x}_{k,(1)} = x_k \in [-1,\ 1]\ \mathrm{m}, \tag{67}$$

$$\mathbf{x}_{k,(3)} = z_k \in [0,\ 3]\ \mathrm{m}. \tag{68}$$

$$\tag{69}$$

Note that the state constraints are set to prevent the optimization algorithm from seeking trajectories that are too far away from the line trajectory between the initial and final state conditions. Consider relaxing these if the optimization algorithm reaches a point of instability. For the NLP setup, $N = 400$, $Q_{\mathrm{x}} = 100I_6$, and $Q_{\mathrm{u}} = 100I_2$. The results from solving the NLP given by (43)–(48) are shown in Figures 10 and 11, which show the obtained optimal state and input trajectories, respectively, with a final time of $T = 1.6432$ s. Figure 10 shows that the resulting optimal trajectory meets the requested initial and final conditions for all states, and Figure 11 shows that the input constraints are met by the optimal trajectory. Finally, the optimal trajectory is illustrated in Figure 12.

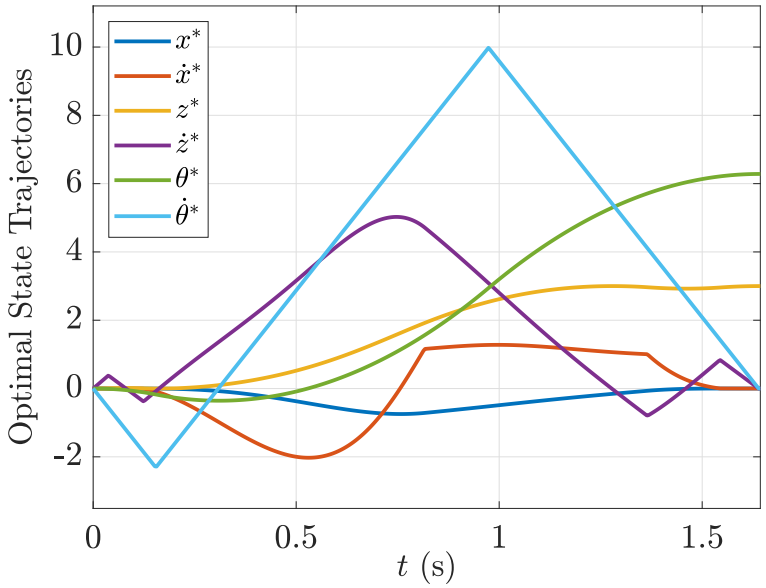

Figure 10: Optimal state trajectories for vertical displacement with flip maneuver considering the multicopter lateral flight dynamics shown in Section 7 obtained by solving the NLP shown in Section 5 with $z(T) = 3$ m and $\theta(T) = 2\pi$ rad, with a final time of $T = 1.6432$ s.

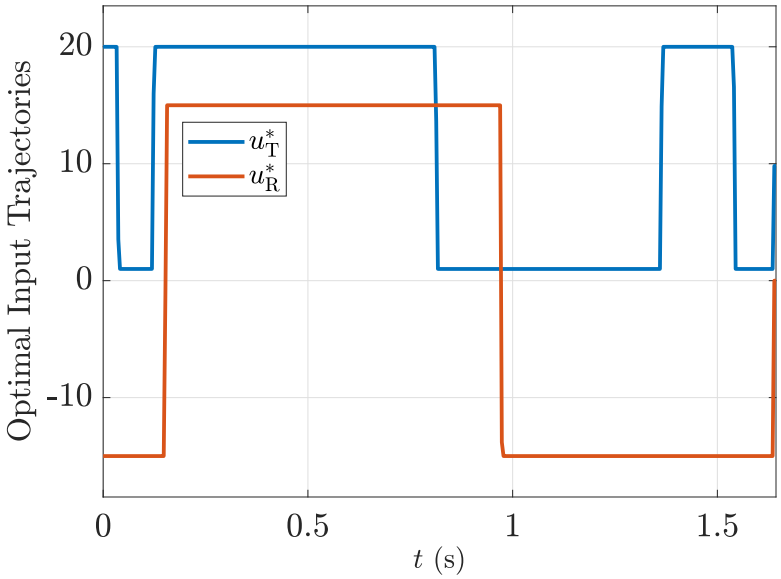

Figure 11: Optimal input trajectories for vertical displacement with flip maneuver considering the multicopter lateral flight dynamics shown in Section 7 obtained by solving the NLP shown in Section 5 with $z(T) = 3$ m and $\theta(T) = 2\pi$ rad, with a final time of $T = 1.6432$ s. The constraint parameters are given by $\underline{F}_{\mathrm{T}}/m = 1$ m/s^2, $\overline{F}_{\mathrm{T}}/m = 20$ m/s^2, and $\overline{\tau}/J_{\mathrm{I}} = 15$ rad/s.

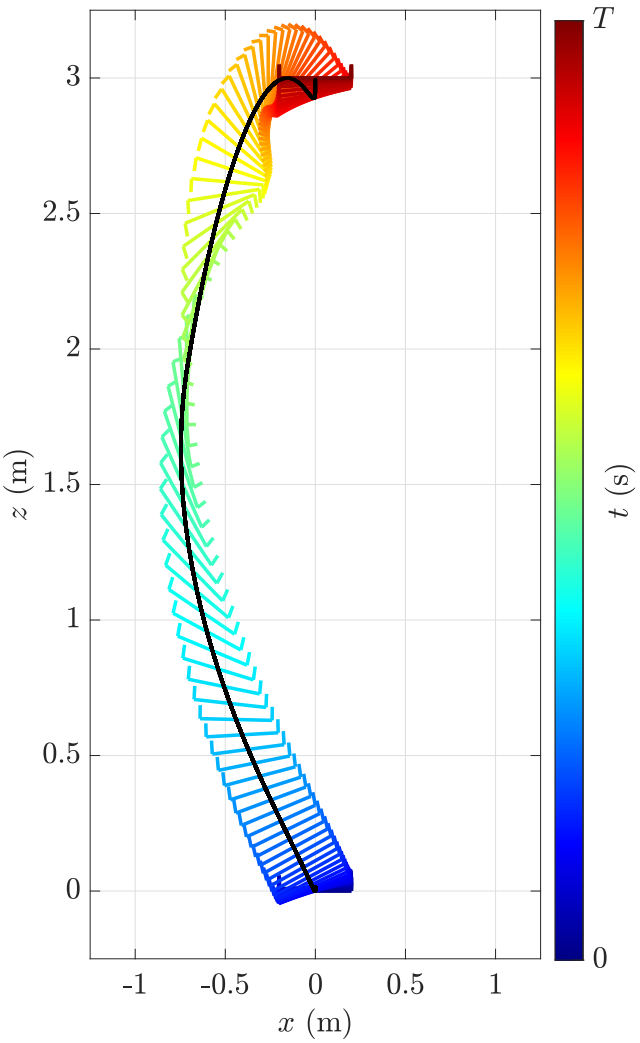

Figure 12: Illustration of the optimal trajectory for vertical displacement with flip maneuver considering the multicopter lateral flight dynamics shown in Section 7 obtained by solving the NLP shown in Section 5 with $z(T) = 3$ m and $\theta(T) = 2\pi$ rad, with a final time of $T = 1.6432$ s. The illustration is created using the `Create_Multicopter_vertical_motion_animation.m` program.

8.2 Horizontal displacement with flip

The optimal trajectories for a horizontal displacement with flip are obtained by solving an optimal-time problem with initial state conditions

$$\mathbf{x}_{\mathrm{i}} = \begin{bmatrix} x(0) & \dot{x}(0) & z(0) & \dot{z}(0) & \theta(0) & \dot{\theta}(0) \end{bmatrix}^{\mathrm{T}} = 0, \tag{70}$$

and final state conditions

$$\mathbf{x}_{\mathrm{f}} = \begin{bmatrix} x(T) & \dot{x}(T) & z(T) & \dot{z}(T) & \theta(T) & \dot{\theta}(T) \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} 12 & 0 & 0 & 0 & 2\pi & 0 \end{bmatrix}^{\mathrm{T}}, \tag{71}$$

which encodes the objective of moving 12 m rightwards and performing a flip in the middle of the trajectory. Furthermore, the following final input conditions are considered

$$\mathbf{u}_{\mathrm{f}} = \begin{bmatrix} u_{\mathrm{T}}(T) & u_{\mathrm{R}}(T) \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} g & 0 \end{bmatrix}^{\mathrm{T}}, \tag{72}$$

which, along the final state conditions, encode the requirement of the multicopter to be in hover conditions at the end of the trajectory. For the considered multicopter dynamics, we consider $g = 9.81m/s^2$, and, for all $k \geq 0$, the following input constraints

$$\mathbf{u}_{k,(1)} = u_{\mathrm{T},k} \in [\underline{F}_{\mathrm{T}}/m,\ \overline{F}_{\mathrm{T}}/m] = [1,\ 20]\ \mathrm{m/s^2}, \tag{73}$$

$$\mathbf{u}_{k,(2)} = u_{\mathrm{R},k} \in [-\overline{\tau}/J_{\mathrm{I}},\ \overline{\tau}/J_{\mathrm{I}}] = [-15,\ 15]\ \mathrm{rad/s^2}. \tag{74}$$

and the following state constraints

$$\mathbf{x}_{k,(1)} = x_k \in [0,\ 12]\ \mathrm{m}, \tag{75}$$

$$\mathbf{x}_{k,(3)} = z_k \in [-2,\ 5]\ \mathrm{m}. \tag{76}$$

$$\tag{77}$$

Note that the state constraints are set to prevent the optimization algorithm from seeking trajectories that are too far away from the line trajectory between the initial and final state conditions. Consider relaxing these if the optimization algorithm reaches a point of instability. For the NLP setup, $N = 400$, $Q_{\mathrm{x}} = 100I_6$, and $Q_{\mathrm{u}} = 100I_2$. The results from solving the NLP given by (43)–(48) are shown in Figures 13 and 14, which show the obtained optimal state and input trajectories, respectively, with a final time of $T = 2.1811$ s. Figure 13 shows that the resulting optimal trajectory meets the requested initial and final conditions for all states, and Figure 14 shows that the input constraints are met by the optimal trajectory. Finally, the optimal trajectory is illustrated in Figure 15.

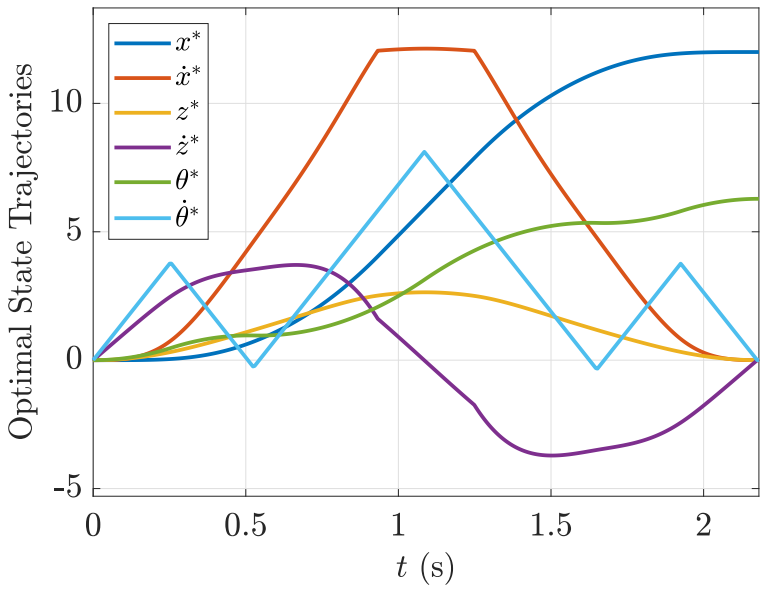

Figure 13: Optimal state trajectories for horizontal displacement with flip maneuver considering the multicopter lateral flight dynamics shown in Section 7 obtained by solving the NLP shown in Section 5 with $x(T) = 12$ m and $\theta(T) = 2\pi$ rad, with a final time of $T = 2.1811$ s.

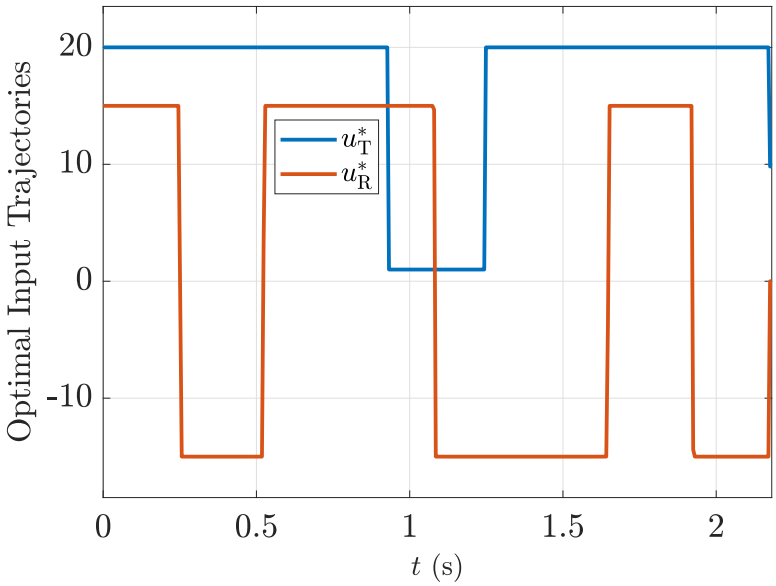

Figure 14: Optimal input trajectories for vertical displacement with flip maneuver considering the multicopter lateral flight dynamics shown in Section 7 obtained by solving the NLP shown in Section 5 with $x(T) = 12$ m and $\theta(T) = 2\pi$ rad, with a final time of $T = 2.1811$ s. The constraint parameters are given by $\underline{F}_{\mathrm{T}}/m = 1$ m/s^2, $\overline{F}_{\mathrm{T}}/m = 20$ m/s^2, and $\overline{\tau}/J_{\mathrm{I}} = 15$ rad/s.

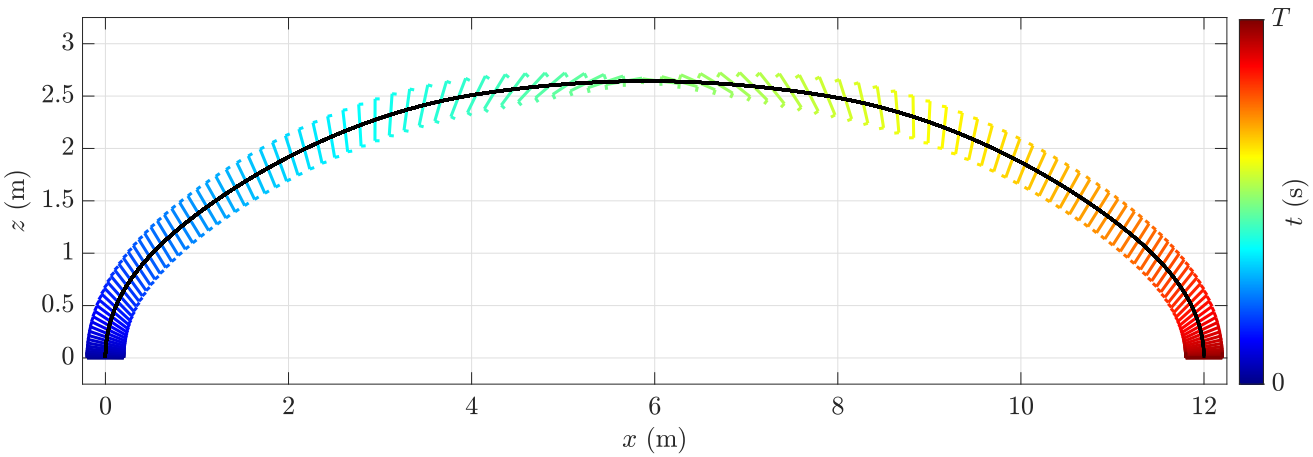

Figure 15: Illustration of the optimal trajectory for horizontal displacement with flip maneuver considering the multicopter lateral flight dynamics shown in Section 7 obtained by solving the NLP shown in Section 5 with $x(T) = 12$ m and $\theta(T) = 2\pi$ rad, with a final time of $T = 2.1811$ s. The illustration is created using the `Create_Multicopter_horizontal_motion_animation.m` program.

9 Conclusions

In this document, two methodologies for calculating the optimal-time state and input trajectories for a multicopter in lateral flight were reviwed. For the first approach obtained from [1], necessary conditions for optimality were derived using PMP, which resulted in a BVP that is preliminarily solved using the STO algorithm. For the second approach, the multicopter dynamics were discretized using Euler's method and a NLP was formulated, which can be solved by using the CasADi software tool. The NLP formulation allows for a more general case of problems to be solved without having to derive PMP optimality conditions and yield very similar results. These trajectories can be used as references in model predictive control applications.

References

[1] M. Hehn, R. Ritz, and R. D'Andrea, "Performance benchmarking of quadrotor systems using time-optimal control," *Autonomous Robots*, vol. 33, no. 1-2, pp. 69–88, 2012.

[2] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.