

# Practical and Accurate Generation of Energy-Optimal Trajectories for a Planar Quadrotor

Fabio Morbidi<sup>1</sup>, Dominik Pisarski<sup>2</sup>

**Abstract**—Motivated by the limited flight time of battery-powered multi-rotor UAVs, in this paper we address the problem of generating energy-optimal trajectories for a planar quadrotor. More specifically, by considering an accurate electrical model for the brushless DC motors and rest-to-rest maneuvers between two predefined boundary states, we explicitly compute the minimum-energy curves by adopting a free and a fixed end-time optimal control formulation. The numerical solution of these optimal control problems hinges upon a simple yet effective indirect projected gradient method. Simulation experiments illustrate the theory in a variety of realistic flight scenarios.

## I. INTRODUCTION

Over the last decade, battery-powered multi-rotor Unmanned Aerial Vehicles (UAVs) have become one of the most popular research platforms in robotics [1]. The dramatic success of quadrotors (multi-rotor UAVs with four propellers) owes much to their hovering capability, agility, affordability and mechanical simplicity. However, their limited flight time is incompatible with the requirements of several emerging applications, such as package delivery [2], long-term videography [3], wildlife monitoring [4], precision agriculture [5], and 3D modeling of large unknown environments [6].

To overcome this problem, a burst of remarkable literature has been recently published on the subject. Ground platforms for automatic battery swap/recharge (e.g. via wireless power transfer) [7]–[13] have been developed to enable long-endurance missions, while the first in-flight battery-switching system has been proposed in [14]. To minimize downtime for replacing or recharging batteries, passive or active perching systems (e.g. actuated grippers) [15]–[20], have also gained momentum. In fact, the aerial vehicle can keep a bird’s eye view of the surrounding environment while resting on the ceiling, a tree branch or a light pole. Unfortunately, these mechanisms are complex, fragile and limit UAV’s payload. Helium-filled envelopes have also been developed to sustain the weight of a rotary-wing system, and thus augment effective travel range [21], [22]. On the downside, these blimp-like flying robots suffer from poor maneuverability, especially outdoors.

As far as the problem of *energy-efficient trajectory generation* is concerned, numerical solutions have been computed for standard quadrotors [23] and hexarotors with dual-tilting

propellers [24], using off-the-shelf optimization packages (ACADO Toolkit, TOMLAB/PROPT). This problem is more challenging than time-optimal path generation [25]–[27], and the two problems have conflicting objective functions. In the same vein as [23], a geometric formulation based on Lie group variational integrators on SE(3) has been proposed in [28], while in [29] the authors used convex optimization to perform real-time 3D path planning. However, in both cases, simple cost functions were minimized, and in [29] a linear quadrotor model was considered. In [30], the authors presented a model-free, adaptive approach based on extremum seeking control, to find the velocity that maximizes the total flight time or flight distance of a quadrotor moving along a given path. This approach was extended in [31], where a new multivariable extremum seeking controller with adaptive step size is considered, and both the speed and the heading angle of the aerial robot are optimized. Finally, offering the possibility of online optimal control, deep learning methods are on the upswing (see e.g. [32], [33]). However, neural networks need massive training data, and their application to trajectory optimization is still in its infancy.

By adopting the same physics-informed cost function as in [23] and by considering the dynamics of a quadrotor on a vertical plane, in this paper we explicitly compute the minimum-energy trajectories between two rest configurations. Differently from [23], [24], we propose here both a *free* and a *fixed end-time* optimal control formulation, and examine the pros and cons. Moreover, instead of leveraging standard *direct* methods based on multiple shooting [23] or pseudospectral collocation [24] for the transcription process, we present a simple *indirect* projected gradient algorithm to numerically solve our optimal control problems. The effectiveness of the proposed approach is demonstrated via realistic simulation experiments. Note that when a trajectory optimization problem is solved with an indirect method, the adjoint equations and their gradients should be explicitly constructed. This is often a nontrivial task, but it gives an excellent accuracy metric for the solution. Direct methods are easier to set up and solve, but do not possess such a built-in accuracy metric. Direct methods are thus more widely used and geared towards non-critical applications, but indirect methods still have an important place in specialized applications, notably aeronautics, where accuracy is crucial [34].

The remainder of this paper is organized as follows. In Sect. II, we recall some basic facts about the optimal control of a nonlinear system. In Sect. III and Sect. IV, we present the dynamical model of the planar quadrotor and the electrical model of a brushless DC motor, respectively. The free and fixed end-time minimum-energy control

<sup>1</sup>MIS laboratory, Université de Picardie Jules Verne, Amiens, France. Email: fabio.morbidi@u-picardie.fr

<sup>2</sup>Institute of Fundamental Technological Research of the Polish Academy of Sciences (IPPT PAN), Warsaw, Poland. Email: dpisar@ippt.pan.pl

This work was supported by the Université de Picardie Jules Verne under the 2020 S2R 4.3 Action “Aide aux Mobilités Sortantes des Enseignants-Chercheurs: Recherche” and by the National Science Centre, Poland, under grant agreement DEC-2017/26/D/ST8/00883.

problems for the aerial robot are formulated in Sect. V. The results of our numerical experiments are discussed in Sect. VI, and conclusions are drawn in Sect. VII.

## II. PRELIMINARIES: FREE AND FIXED END-TIME OPTIMAL CONTROL

In this section, we briefly review the free and fixed end-time optimal control problems (OCPs) for a generic nonlinear time-invariant system [35].

Let  $\mathbf{x}(t) \in \mathcal{X} \subset \mathbb{R}^n$  and  $\mathbf{u}(t) \in \mathcal{U}^m = [u_{\min}, u_{\max}]^m \subset \mathbb{R}^m$  be the state vector and control input at time  $t \in [0, T]$ ,  $T \in \mathcal{T} = [T_{\min}, T_{\max}] \subset \mathbb{R}_{\geq 0}$ , respectively. Consider the general nonlinear dynamics,

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad \mathbf{x}(0) = \mathbf{x}_0, \quad (1)$$

the objective functional,

$$\mathcal{E} = \int_0^T F(\mathbf{x}, \mathbf{u}) dt, \quad (2)$$

and the terminal state  $\mathbf{x}(T) = \mathbf{x}_T$ . To accommodate the terminal state in an optimal control problem, we shall employ the penalty function,

$$h(\mathbf{x}(T)) = \alpha \|\mathbf{x}(T) - \mathbf{x}_T\|^2, \quad (3)$$

with  $\alpha > 0$ , where  $\|\cdot\|$  denotes the Euclidean norm. Let us now introduce the normalized time  $\tau \in [0, 1]$  and assume that  $t = T\tau$ , cf. [36]. Hereafter, to avoid confusion, for any time-dependent function  $p$ , we shall denote by  $p'$  the first derivative of  $p$  with respect to  $\tau$ . Note that the substitution  $t = T\tau$  leads to:

$$\mathbf{x}' = T \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad \mathbf{x}(0) = \mathbf{x}_0, \quad (4)$$

for (1), and

$$\mathcal{E} = T \int_0^1 F(\mathbf{x}, \mathbf{u}) d\tau, \quad (5)$$

for (2). Considering the end-time  $T$  as the decision variable, the free end-time problem of minimizing (5) for the dynamics (4) and the terminal-state constraint, can be written as:

### Free End-Time OCP:

Find  $(\mathbf{u}^*(T\tau), T^*)$  minimizing the functional,

$$\mathcal{J}(\mathbf{x}, \mathbf{u}, T) = h(\mathbf{x}(T)) + T \int_0^1 F(\mathbf{x}, \mathbf{u}) d\tau, \quad (6)$$

s.t.  $\mathbf{x}' = T \mathbf{f}(\mathbf{x}, \mathbf{u})$ ,  $\mathbf{x}(0) = \mathbf{x}_0$ ,  $\mathbf{u} \in \mathcal{U}^m$ ,  $T \in \mathcal{T}$ .

In what follows, we will often drop the state argument in the cost functional, and we will write it as  $\mathcal{J}(\mathbf{u}, T)$  or simply  $\mathcal{J}$ . By the same token, for the predefined end-time  $T$ , we can introduce the following fixed end-time problem:

### Fixed End-Time OCP:

Find  $\mathbf{u}^*(T\tau)$  minimizing the functional,

$$\mathcal{J}(\mathbf{x}, \mathbf{u}) = h(\mathbf{x}(T)) + T \int_0^1 F(\mathbf{x}, \mathbf{u}) d\tau, \quad (7)$$

s.t.  $\mathbf{x}' = T \mathbf{f}(\mathbf{x}, \mathbf{u})$ ,  $\mathbf{x}(0) = \mathbf{x}_0$ ,  $\mathbf{u} \in \mathcal{U}^m$ .

### A. Necessary optimality condition for the Free End-Time OCP

Let us introduce the adjoint state  $\mathbf{r} = \mathbf{r}(T\tau) \in \mathbb{R}^n$  and the Hamiltonian  $H = H(\mathbf{x}, \mathbf{r}, \mathbf{u}, T)$ , defined respectively by:

$$\mathbf{r}' = -\frac{\partial H}{\partial \mathbf{x}}, \quad \mathbf{r}(T) = -\frac{\partial h}{\partial \mathbf{x}} \Big|_{\tau=1}, \quad (8)$$

and

$$H = T(\mathbf{r}^\top \mathbf{f} - F), \quad (9)$$

where the partial derivative in the second equation of (8) is evaluated at  $\tau = 1$ . Observe from (9) that,

$$\mathbf{x}' = \frac{\partial H}{\partial \mathbf{r}}. \quad (10)$$

Moreover, from (9), the objective functional in (6) can be equivalently rewritten as:

$$\mathcal{J} = h(\mathbf{x}(T)) + \int_0^1 (\mathbf{r}^\top \mathbf{x}' - H) d\tau. \quad (11)$$

Infinitesimal changes of the control function  $\delta \mathbf{u}$  and end-time parameter  $\delta T$ , produce the variations of the functions  $\delta \mathbf{x}$ ,  $\delta \mathbf{x}'$  and  $\delta \mathbf{r}$ . Regarding the Hamiltonian dependency,  $H = H(\mathbf{x}, \mathbf{r}, \mathbf{u}, T)$ , the previous variations result in the following variation of the objective functional (11),

$$\begin{aligned} \delta \mathcal{J} = & \left[ \left( \frac{\partial h}{\partial \mathbf{x}} \right)^\top \delta \mathbf{x} \right] \Big|_{\tau=1} + \int_0^1 [\mathbf{r}^\top \delta \mathbf{x}' + (\mathbf{x}')^\top \delta \mathbf{r}] d\tau \\ & - \int_0^1 \left[ \left( \frac{\partial H}{\partial \mathbf{x}} \right)^\top \delta \mathbf{x} + \left( \frac{\partial H}{\partial \mathbf{r}} \right)^\top \delta \mathbf{r} + \left( \frac{\partial H}{\partial \mathbf{u}} \right)^\top \delta \mathbf{u} + \frac{\partial H}{\partial T} \delta T \right] d\tau. \end{aligned} \quad (12)$$

Applying integration by parts, we have,

$$\int_0^1 \mathbf{r}^\top \delta \mathbf{x}' d\tau = [\mathbf{r}^\top \delta \mathbf{x}]_0^1 - \int_0^1 (\mathbf{r}')^\top \delta \mathbf{x} d\tau. \quad (13)$$

Plugging (13) into (12), yields:

$$\begin{aligned} \delta \mathcal{J} = & \left[ \left( \frac{\partial h}{\partial \mathbf{x}} \right)^\top \delta \mathbf{x} \right] \Big|_{\tau=1} + [\mathbf{r}^\top \delta \mathbf{x}]_0^1 \\ & - \int_0^1 \left[ \left( \mathbf{r}' + \frac{\partial H}{\partial \mathbf{x}} \right)^\top \delta \mathbf{x} - \left( \mathbf{x}' - \frac{\partial H}{\partial \mathbf{r}} \right)^\top \delta \mathbf{r} \right] d\tau \\ & - \int_0^1 \left[ \left( \frac{\partial H}{\partial \mathbf{u}} \right)^\top \delta \mathbf{u} + \frac{\partial H}{\partial T} \delta T \right] d\tau. \end{aligned} \quad (14)$$

Note that the two terms in the first line of equation (14) vanish, since  $\delta \mathbf{x}(0) = 0$  (see equ. (1)) and  $\mathbf{r}(T) = -\frac{\partial h}{\partial \mathbf{x}} \Big|_{\tau=1}$  (see equ. (8)). From the first equation in (8) and (10), we conclude that the terms in the second line of (14) also vanish. Hence, the variation of the objective functional (11) can be simply written as,

$$\delta \mathcal{J} = - \int_0^1 \left( \frac{\partial H}{\partial \mathbf{u}} \right)^\top \delta \mathbf{u} d\tau - \int_0^1 \frac{\partial H}{\partial T} \delta T d\tau. \quad (15)$$

From (15), it follows that for every  $\tau \in [0, 1]$ , the derivative of the objective functional  $\mathcal{J}$  with respect to the control input  $\mathbf{u}$  is given by,

$$\frac{\delta \mathcal{J}}{\delta \mathbf{u}}(\tau) = - \frac{\partial H}{\partial \mathbf{u}} \Big|_{\tau}. \quad (16)$$

On the other hand, the derivative of  $\mathcal{J}$  with respect to the end-time parameter  $T$  is,

$$\frac{\delta \mathcal{J}}{\delta T} = - \int_0^1 \frac{\partial H}{\partial T} d\tau. \quad (17)$$

The (first-order) *necessary optimality condition* for problem (6), is thus of the following form. Let  $(\mathbf{u}^*(T\tau), T^*)$  be an optimal solution to (6), then for every  $\mathbf{u}(T\tau) \in \mathcal{U}^m$  and  $T \in \mathcal{T}$ , one has:

$$\left( \frac{\delta \mathcal{J}}{\delta \mathbf{u}}(\tau) \right)^\top (\mathbf{u}(T\tau) - \mathbf{u}^*(T\tau)) \geq 0, \quad \tau \in [0, 1], \quad (18)$$

$$\frac{\delta \mathcal{J}}{\delta T}(T - T^*) \geq 0.$$

### B. Necessary optimality condition for the Fixed End-Time OCP

The variation of the objective functional for problem (7) can be derived by analogy with steps (8)-(14), assuming a fixed  $T$ , and consequently inserting  $\delta T = 0$ . As a result,

$$\delta \mathcal{J} = - \int_0^1 \left( \frac{\partial H}{\partial \mathbf{u}} \right)^\top \delta \mathbf{u} d\tau.$$

The derivative of  $\mathcal{J}$  with respect to the control input  $\mathbf{u}$ , for every  $\tau \in [0, 1]$ , is given by (16). The *necessary optimality condition* for problem (7) is thus as follows. Let  $\mathbf{u}^*(T\tau)$  be an optimal solution to problem (7), then for every  $\mathbf{u}(T\tau) \in \mathcal{U}^m$ , condition (18) holds true.

### C. Numerical algorithm for the Free End-Time OCP

To conclude this section, we present here a simple *projected gradient method* [37, Sect. 2.3] for the numerical solution of problem (6). Given the input vector  $\mathbf{u} = [u_1, \dots, u_m]^\top$ , let  $\text{Proj}_{\mathcal{U}}(u_j)$  denote the cut-off function [38] projecting the control input  $u_j$ ,  $j \in \{1, \dots, m\}$ , onto the set  $\mathcal{U} = [u_{\min}, u_{\max}]$ , i.e.  $\text{Proj}_{\mathcal{U}}(u_j) = \max\{u_{\min}, \min\{u_{\max}, u_j\}\}$ . Analogously, the cut-off function projecting the end-time  $T$  onto the set  $\mathcal{T} = [T_{\min}, T_{\max}]$  is defined as  $\text{Proj}_{\mathcal{T}}(T) = \max\{T_{\min}, \min\{T_{\max}, T\}\}$ . The control input  $u_j$ ,  $j \in \{1, \dots, m\}$ , is thus updated via the following sequence, for  $l \in \{0, 1, \dots, l_{\max}\}$ ,

$$u_j^{l+1}(T\tau) = \text{Proj}_{\mathcal{U}} \left( u_j^l(T\tau) - \gamma_u \frac{\delta \mathcal{J}}{\delta u_j}(\tau) \right), \quad \tau \in [0, 1], \quad (19)$$

where  $l_{\max}$  denotes the maximum number of iterations and  $\gamma_u$  is a positive step size. Similarly, the end-time  $T$  can be updated by using the sequence,

$$T^{l+1} = \text{Proj}_{\mathcal{T}} \left( T^l - \gamma_T \frac{\delta \mathcal{J}}{\delta T} \right), \quad l \in \{0, 1, \dots, l_{\max}\}, \quad (20)$$

where  $\gamma_T$  is a positive step size. Overall, the numerical optimization algorithm consists of seven steps, reported in detail in the box below.

**Remark 1:** For the Fixed End-Time OCP, Algorithm 1 is executed by substituting the fixed-time value for  $T^l$  at each iteration  $l$  (**Steps 1–7**), and by omitting both the evaluation of the derivative (17) (**Step 4**) and the computation of the end-time parameter  $T^{l+1}$  (**Step 5**).  $\diamond$

Algorithm 1: Pseudo-code of the numerical algorithm for the Free End-Time OCP (6).

- Step 1:** Set  $l := 0$ . Select the initial control function  $u_j^l$ ,  $j \in \{1, \dots, m\}$  and end-time parameter  $T^l$ . Select small positive constants  $\gamma_u$ ,  $\gamma_T$  and  $\epsilon$ . Set the maximal number of iterations  $l_{\max}$ . Select  $\kappa_u, \kappa_T \in (0, 1)$ .
- Step 2:** Solve the state equation (4) by substituting  $\mathbf{u} = \mathbf{u}^l = [u_1^l, \dots, u_m^l]^\top$  and  $T = T^l$ .
- Step 3:** By backward integration, solve the adjoint state equation (8) by substituting the solution to (4) and  $\mathbf{u} = \mathbf{u}^l$ ,  $T = T^l$ .
- Step 4:** Evaluate the derivatives (16) and (17) by substituting the solution to (4), the solution to (8) and  $\mathbf{u} = \mathbf{u}^l$ ,  $T = T^l$ . Compute the value of the objective functional  $\mathcal{J}(\mathbf{u}^l, T^l)$ .
- Step 5:** Compute the control function  $u_j^{l+1}$ ,  $j \in \{1, \dots, m\}$  with (19), and the end-time parameter  $T^{l+1}$  with (20).
- Step 6:** Compute the value of the objective functional  $\mathcal{J}(\mathbf{u}^{l+1}, T^{l+1})$ . If  $\mathcal{J}(\mathbf{u}^{l+1}, T^{l+1}) < \mathcal{J}(\mathbf{u}^l, T^l)$ , then set  $l := l + 1$  and go to **Step 7**. Otherwise, modify the step size  $\gamma_u$  (and/or  $\gamma_T$ ) by  $\gamma_u := \kappa_u \gamma_u$  ( $\gamma_T := \kappa_T \gamma_T$ ), and go to **Step 5**.
- Step 7:** Check if any of the terminal conditions are fulfilled, i.e. if  $l = l_{\max}$  or if  $\frac{|\mathcal{J}(\mathbf{u}^{l+1}, T^{l+1}) - \mathcal{J}(\mathbf{u}^l, T^l)|}{\mathcal{J}(\mathbf{u}^l, T^l)} < \epsilon$ . If yes, set  $\mathbf{u}^* = \mathbf{u}^l$ ,  $T^* = T^l$  and **STOP**. Otherwise go to **Step 2**.

## III. DYNAMICAL MODEL OF A PLANAR QUADROTOR

In this section, we present the dynamical model of a quadrotor moving on the vertical  $xz$ -plane.

Let  $\mathcal{F}_W = \{\mathbf{O}_W; \mathbf{x}_W, \mathbf{z}_W\}$  be the world frame and  $\mathcal{F}_B = \{\mathbf{O}_B; \mathbf{x}_B, \mathbf{z}_B\}$  the body frame attached to the quadrotor center of mass, with  $\mathbf{z}_B$  aligned with the thrust direction (see Fig. 1). The state vector  $\mathbf{q}$  of the planar quadrotor includes its position  $\mathbf{p} = [x, z]^\top$  and linear velocity  $\dot{\mathbf{p}} = \mathbf{v} = [v_x, v_z]^\top$  in  $\mathcal{F}_W$ , and the quadrotor body orientation  $\theta$  and angular velocity  $\omega$ , thus  $\mathbf{q} = [\mathbf{p}^\top, \mathbf{v}^\top, \theta, \omega]^\top \in \mathbb{R}^6$ . The total thrust  $f$  at hover applied to the airframe and the net moment  $\tau$  arising from the aerodynamics applied to the quadrotor, are related to the angular velocity of the right and left rotor  $(w_r, w_\ell)$ , by

$$\begin{bmatrix} f \\ \tau \end{bmatrix} = \begin{bmatrix} c_f & c_f \\ dc_f & -dc_f \end{bmatrix} \begin{bmatrix} w_r^2 \\ w_\ell^2 \end{bmatrix}, \quad (21)$$

where  $c_f > 0$  is the thrust factor of the rotors (which depends on rotor geometry and profile) [39], and  $d$  is the distance between  $\mathbf{O}_B$  and rotor's center (see Fig. 1). Ignoring Coriolis effects, the dynamical model of the quadrotor can be written as follows,

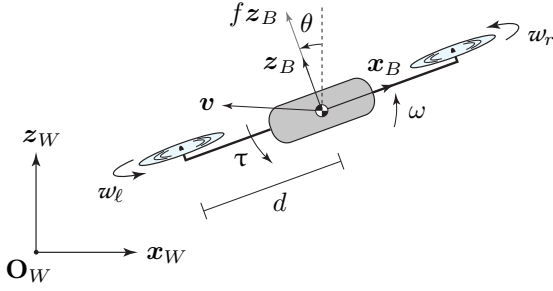


Fig. 1: Main quantities characterizing the dynamical model of a planar quadrotor.

$$\begin{cases} \dot{\mathbf{p}} = \mathbf{v}, \\ \dot{\mathbf{v}} = \begin{bmatrix} 0 \\ -g \end{bmatrix} + \frac{f}{M} \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix} - \mathbf{B}(\theta)\mathbf{v}, \\ \dot{\theta} = \omega, \\ \dot{\omega} = \frac{\tau}{I}, \end{cases} \quad (22)$$

where  $M$  and  $I$  denote the mass and inertia of the aerial vehicle, respectively, and  $g = 9.8066 \text{ m/s}^2$  is the acceleration due to gravity. The term  $\mathbf{B}(\theta)\mathbf{v}$  in (22) is used to model a body-frame air drag with possibly different magnitudes along the  $x_B$ - and  $z_B$ -axis. Matrix  $\mathbf{B}(\theta)$  is defined as,

$$\mathbf{B}(\theta) = \mathbf{R}(\theta) \begin{bmatrix} \beta_x & 0 \\ 0 & \beta_z \end{bmatrix} \mathbf{R}^\top(\theta), \quad \mathbf{R}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix},$$

where  $\mathbf{R}(\theta)$  represents the orientation of  $\mathcal{F}_B$  with respect to  $\mathcal{F}_W$ , and  $\beta_x \geq 0$  and  $\beta_z \geq 0$  are the body-frame drag coefficients along the  $x_B$ - and  $z_B$ -axis, respectively.

#### IV. ELECTRICAL MODEL OF A DC MOTOR

The model for a brushless DC motor of the quadrotor takes into account the energy dissipated in the resistive and inductive windings, and the energy required to overcome the internal and load friction [23], [40]. The instantaneous current  $i(t)$  in the motor is given by (see Fig. 2),

$$i(t) = \frac{1}{K_\tau} [\mathbb{T}_f + \mathbb{T}_L(w(t)) + D_f w(t) + (J_m + J_L) \frac{dw(t)}{dt}], \quad (23)$$

where  $w(t)$  is the angular velocity of the motor shaft [rad/s],  $K_\tau$  is the motor torque constant [Nm/A],  $\mathbb{T}_f$  is the motor friction torque,  $\mathbb{T}_L(w(t))$  is the speed-dependent load friction torque which results from propeller drag,  $D_f$  is the viscous damping coefficient of the motor [Nms/rad], and  $J_m$  and  $J_L$  are the motor and load moments of inertia, respectively. The voltage  $e(t)$  across the motor is given by:

$$e(t) = R i(t) + K_E w(t) + L \frac{di(t)}{dt}, \quad (24)$$

where  $R$  and  $L$  are the resistance and inductance of phase winding, respectively, and  $K_E$  is the voltage constant of the motor [Vs/rad]. Note that  $K_E = K_\tau$  [23]: in addition, if  $K_E$  is expressed in mV/rpm, we have that  $K_E = 1000/K_V$  where  $K_V$  is the motor velocity constant [rpm/V]. Since the resistance  $R_L$ , representing the losses in the magnetic circuit of the motor, is usually much larger than  $R$ , the effect of  $R_L$

on motor operation can be neglected. Under steady-state conditions, the current  $i(t)$  is constant and (24) reduces to:

$$e(t) = R i(t) + K_E w(t), \quad (25)$$

where  $e_g(t) = K_E w(t)$  is the counter electromotive force of the motor.

#### V. MINIMUM-ENERGY OCPS

With the previous results in place, we are now ready to introduce the Free and Fixed End-Time OCPS for system (22). Let  $e_r(t)$ ,  $e_\ell(t)$  and  $i_r(t)$ ,  $i_\ell(t)$  respectively denote the voltage and current across the right and left motor of the quadrotor at time  $t$ . The energy consumed by the quadrotor in the time interval  $[0, T]$  is then,

$$\mathcal{E} = \int_0^T [e_r(t) i_r(t) + e_\ell(t) i_\ell(t)] dt. \quad (26)$$

By leveraging (23) and (25) for the two identical motors, and by noticing that  $\mathbb{T}_L(w_r(t)) = c_\tau w_r^2(t)$ ,  $\mathbb{T}_L(w_\ell(t)) = c_\tau w_\ell^2(t)$  where  $c_\tau$  is the aerodynamic drag factor of the rotors [39], we can equivalently rewrite (26) as follows,

$$\mathcal{E} = \int_0^T \sum_{j \in \{r, \ell\}} [b_1 + b_2 w_j(t) + b_3 w_j^2(t) + b_4 w_j^3(t) + b_5 w_j^4(t) + b_6 \dot{w}_j(t) + b_7 \dot{w}_j^2(t) + b_8 w_j(t) \dot{w}_j(t) + b_9 w_j^2(t) \dot{w}_j(t)] dt,$$

where  $b_1, b_2, \dots, b_9$  are positive constants whose explicit expression is reported in [24]. To formulate the minimum-energy trajectory generation problem as a standard OCP (cf. Sect. II), let us introduce the *extended state vector*  $\mathbf{x} = [x_1, x_2, \dots, x_8]^\top = [x, v_x, z, v_z, \theta, \omega, w_r, w_\ell]^\top$  and the *auxiliary control input*  $\mathbf{u} = [u_1, u_2]^\top = [\dot{w}_r, \dot{w}_\ell]^\top$ . Then, by recalling equ. (21),

$$\mathbf{f}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} x_2 \\ -\frac{c_f(x_7^2 + x_8^2)}{M} \sin x_5 - (\beta_x \cos^2 x_5 + \beta_z \sin^2 x_5) x_2 \\ -(\beta_x - \beta_z) x_4 \cos x_5 \sin x_5 \\ x_4 \\ -g + \frac{c_f(x_7^2 + x_8^2)}{M} \cos x_5 - (\beta_z \cos^2 x_5 + \beta_x \sin^2 x_5) x_4 \\ -(\beta_x - \beta_z) x_2 \cos x_5 \sin x_5 \\ x_6 \\ \frac{dc_f(x_7^2 - x_8^2)}{I} \\ \mathbf{u} \end{bmatrix}.$$

We can finally write the following *Free End-Time OCP*:

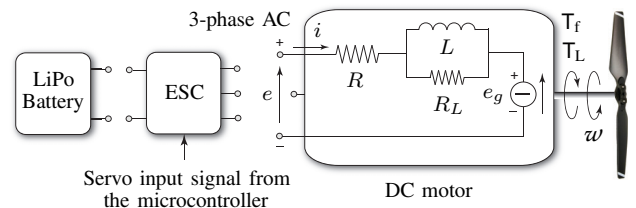


Fig. 2: Electrical model of a brushless DC motor of the quadrotor.

$$\min_{\mathbf{u}(T\tau), T} h(\mathbf{x}(T)) + T \int_0^1 F(\mathbf{x}, \mathbf{u}) d\tau, \quad (27)$$

$$\text{s.t. } \mathbf{x}' = T \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad \mathbf{x}(0) = \mathbf{x}_0, \quad \mathbf{u} \in \mathcal{U}^2, \quad T \in \mathcal{T},$$

where

$$\begin{aligned} F(\mathbf{x}, \mathbf{u}) = & 2b_1 + b_2(x_7 + x_8) + b_3(x_7^2 + x_8^2) + b_4(x_7^3 + x_8^3) \\ & + b_5(x_7^4 + x_8^4) + b_6(u_1 + u_2) + b_7(u_1^2 + u_2^2) \\ & + b_8(x_7 u_1 + x_8 u_2) + b_9(x_7^2 u_1 + x_8^2 u_2), \end{aligned}$$

$\mathbf{x}_0, \mathbf{x}_T$  are the two boundary states,  $\mathcal{U} = [u_{\min}, u_{\max}]$  and  $\mathcal{T} = [T_{\min}, T_{\max}]$ . Likewise, we can state the *Fixed End-Time OCP* as,

$$\min_{\mathbf{u}(T\tau)} h(\mathbf{x}(T)) + T \int_0^1 F(\mathbf{x}, \mathbf{u}) d\tau, \quad (28)$$

$$\text{s.t. } \mathbf{x}' = T \mathbf{f}(\mathbf{x}, \mathbf{u}), \quad \mathbf{x}(0) = \mathbf{x}_0, \quad \mathbf{u} \in \mathcal{U}^2.$$

The existence of the solutions to problems (27) and (28) follows from the continuity of the objective functional and compactness of the extended state's reachable sets (see Filippov's theorem in [41, Sect. 4.5]).

## VI. SIMULATION EXPERIMENTS

In our simulation experiments we limited our attention to rest-to-rest maneuvers of the planar quadrotor. In fact, we assumed zero linear and angular velocity of the aircraft at the boundary states, with zero body orientation at time  $t = 0$ , and free body orientation and free spinning velocities of the propellers at time  $t = T$ . To specify the start/end positions in  $\mathcal{F}_W$ , we selected the following points:  $A = (0, 0)$  m,  $B = (0, 10)$  m, and  $C = (2, 10)$  m. In **Case 1**, we computed the minimum-energy trajectory between  $A$  and  $B$ , or  $A \rightarrow B$ , in short. Analogously, **Case 2** corresponds to  $A \rightarrow C$ , **Case 3** to  $B \rightarrow C$ , **Case 4** to  $B \rightarrow A$ , and **Case 5** to  $C \rightarrow A$ .

The physical parameters of the quadrotor used in our tests are reported in Table I, cf. [23]. With these values, the angular velocity of the propellers for hovering at time  $t = 0$ , is 357.89 rad/s. To implement Algorithm 1, we considered the following set of parameters:  $\gamma_u = 10^{-6}$ ,  $\gamma_T = 10^{-10}$ ,  $\kappa_u, \kappa_T \in [0.5, 0.95]$  (different selections for individual cases based on trial and error),  $l_{\max} = 10^5$  and  $\epsilon = 10^{-7}$ . For the extreme values of the decision parameters, we set  $T_{\min} = 2.2$  s (**Case 1** and **2**),  $T_{\min} = 1.8$  s (**Case 3**),  $T_{\min} = 3.2$  s (**Case 4** and **5**),  $T_{\max} = 10$  s (**Cases 1–5**), and  $u_{\min} = -10^3$  rad/s<sup>2</sup>,  $u_{\max} = 10^3$  rad/s<sup>2</sup>. With these values of  $T_{\min}$ , the physical constraints on the angular velocity of the two propellers ( $0 \leq x_7(t) \leq w_{\max}$ ,  $0 \leq x_8(t) \leq w_{\max}$  with  $w_{\max} = 10^4$  rpm  $\simeq 1047.2$  rad/s), were fulfilled at all times.

$K_E = 9.5493/K_V$ Vs/rad	$K_V = 760$ rpm/V
$R = 0.07$ $\Omega$	$T_f = 4 \times 10^{-2}$ Nm
$D_f = 2 \times 10^{-4}$ Nms/rad	$r_{\text{rot}} = 0.0175$ m, $M_{\text{mot}} = 0.1$ kg
$n_{\text{bl}} = 2$ , $M_{\text{bl}} = 0.056$ kg	$r = 0.2134$ m, $\epsilon_{\text{cl}} = 0.0056$ m
$J_m = \frac{1}{2} M_{\text{mot}} r_{\text{rot}}^2$ kgm <sup>2</sup>	$J_L = \frac{1}{4} n_{\text{bl}} M_{\text{bl}} (r - \epsilon_{\text{cl}})^2$ kgm <sup>2</sup>
$c_f = 3.8281 \times 10^{-5}$ Ns <sup>2</sup>	$c_\tau = 4.0013 \times 10^{-7}$ Nms <sup>2</sup>
$d = 0.175$ m, $\beta_x = \beta_z = 0.3$	$M = 1$ kg, $I = 0.081$ kgm <sup>2</sup>

TABLE I: Physical parameters of the quadrotor used in the simulation experiments. For the meaning of the symbols  $r_{\text{rot}}$ ,  $M_{\text{mot}}$ ,  $r$ ,  $\epsilon_{\text{cl}}$ ,  $n_{\text{bl}}$ , and  $M_{\text{bl}}$ , see [23].

<b>Case 1: A <math>\rightarrow</math> B</b>	Time $T$ [s]	Energy consump. $\mathcal{E}$ [J]
Fixed End-Time OCP ( $T_{\min}$ )	2.2000	485.5764
Free End-Time OCP	2.5454	384.2163
Fixed End-Time OCP	3.0000	402.1428
<b>Case 2: A <math>\rightarrow</math> C</b>	Time $T$ [s]	Energy consump. $\mathcal{E}$ [J]
Fixed End-Time OCP ( $T_{\min}$ )	2.2000	502.4923
Free End-Time OCP	2.8407	395.4968
Fixed End-Time OCP	3.0000	408.2454
<b>Case 3: B <math>\rightarrow</math> C</b>	Time $T$ [s]	Energy consump. $\mathcal{E}$ [J]
Fixed End-Time OCP ( $T_{\min}$ )	1.8000	349.0976
Free End-Time OCP	2.0845	332.1295
Fixed End-Time OCP	2.5000	354.0323
<b>Case 4: B <math>\rightarrow</math> A</b>	Time $T$ [s]	Energy consump. $\mathcal{E}$ [J]
Fixed End-Time OCP ( $T_{\min}$ )	3.2000	518.2877
Free End-Time OCP	3.4547	506.5499
Fixed End-Time OCP	4.0000	523.1461
<b>Case 5: C <math>\rightarrow</math> A</b>	Time $T$ [s]	Energy consump. $\mathcal{E}$ [J]
Fixed End-Time OCP ( $T_{\min}$ )	3.2000	624.8669
Free End-Time OCP	3.6483	611.6205
Fixed End-Time OCP	4.0000	626.0317

TABLE II: Optimal values of the energy consumption  $\mathcal{E}$  for the Fixed and Free End-Time OCPs in **Cases 1–5**.

For the weighting coefficient  $\alpha$  of the penalty function (3) in problems (27) and (28), one should select a sufficiently large value, provided that the terminal state  $\mathbf{x}(T)$  is contained within a closed neighborhood of  $\mathbf{x}_T$ . In **Cases 1–5**, we chose  $\alpha = 4 \times 10^5$ , which resulted in  $\|\mathbf{x}(T) - \mathbf{x}_T\|^2 < 10^{-5}$ . For the discretization of the control function  $\mathbf{u}(t)$ , we assumed 100 time steps. The state and adjoint state dynamics were solved by using a Runge-Kutta 4th-order scheme, on a PC running Windows with 2.9 GHz Intel Core i9 processor and 16 GB RAM without parallelization. The average time for finding an optimal solution was around 50 s, under Matlab R2017. In each scenario, we started the optimization for the Fixed End-Time OCP, by assuming  $T = T_{\min}$  and  $\mathbf{u}^0 = \mathbf{0}$ . The terminal time  $T_{\min}$  and optimal control  $\mathbf{u}^*$  resulting from this first optimization, were then used as initial guesses of  $\mathbf{u}^0$  and  $T^l$  for the Free End-Time OCP. Finally, for comparison, the Fixed End-Time OCP was solved by setting  $T = 3.0$  s (**Case 1** and **2**),  $T = 2.5$  s (**Case 3**),  $T = 4.0$  s (**Case 4** and **5**), and  $\mathbf{u}^0 = \mathbf{0}$  (**Cases 1–5**). Table II reports the three optimal values of  $\mathcal{E}$  in **Cases 1–5**.

**Case 2** is representative of our numerical results. Fig. 4(a) reports the value of  $\mathcal{J}(\mathbf{u}^l, T^l)$  in problem (27) during the optimization process, showing the convergence of the projected gradient descent in Algorithm 1. For the initial control, we selected the solution to the Fixed End-Time OCP with  $T = 2.2$  s, and the procedure stopped after  $l = 8212$  iterations, when the condition on the value of the objective functional in **Step 7** of Algorithm 1, was fulfilled. The time evolution of the  $x$ - and  $z$ - coordinates of the quadrotor is reported in Figs. 3(a) and 3(b), and that of the control inputs  $u_1$  and  $u_2$  in Figs. 3(c) and 3(d), respectively (solid, Free-End Time; dashed, Fixed-End Time with  $T = 2.2$  s; dotted, Fixed-End Time with  $T = 3.0$  s). The minimum-energy

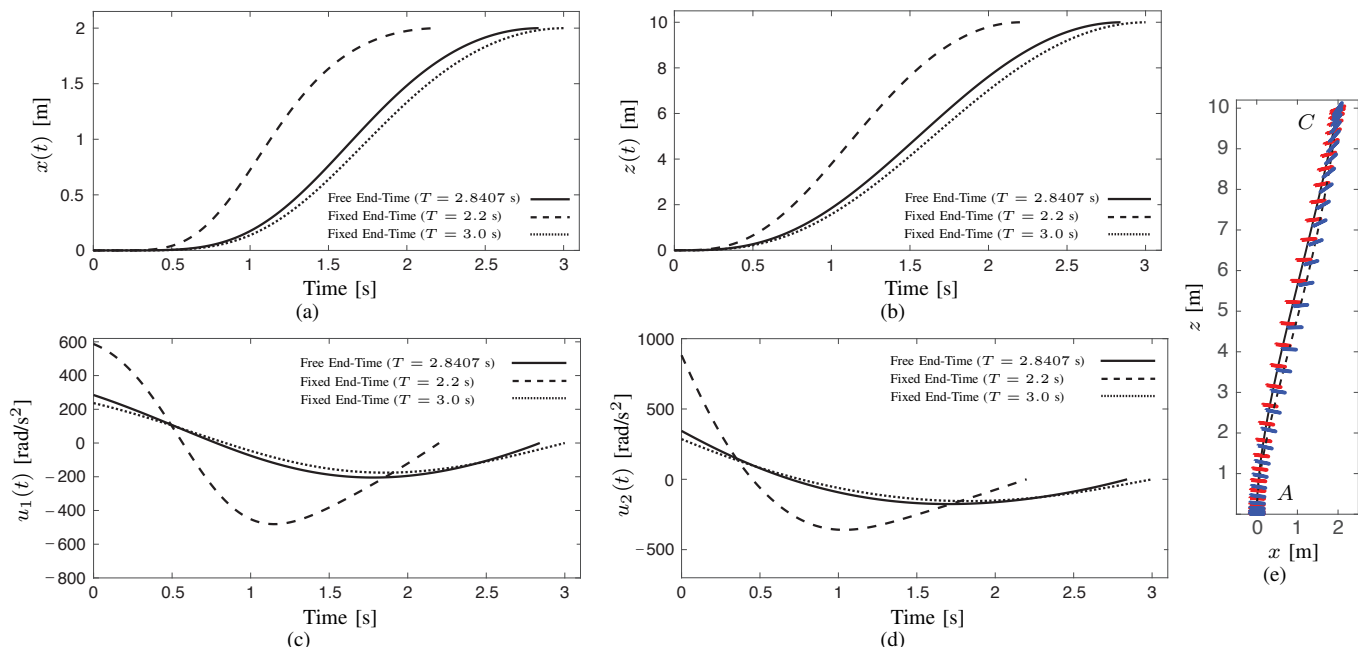


Fig. 3: **Case 2**: (a),(b) Time-evolution of the  $x$ - and  $z$ - coordinates, and (c),(d) time-evolution of the control inputs  $u_1(t)$  and  $u_2(t)$  of the quadrotor; (e) Minimum-energy trajectory of the quadrotor with Free End-Time (red) and Fixed End-Time with  $T = 2.2$  s (blue).

trajectory of the quadrotor for Free End-Time (red) and Fixed End-Time with  $T = 2.2$  s (blue) is shown in Fig. 3(e).

For use as a baseline, we also numerically solved problem (28) for **Case 3** with the OpenOCL software library for Matlab [42]. We considered 30 control intervals with an interpolating polynomial of degree 3 in each interval, and we set  $\alpha = 4 \times 10^5$  and  $T = 2.5$  s. After 59 iterations (and 129 objective function evaluations), this off-the-shelf solver found an optimal value of 362.5460 for  $\mathcal{J}$ , compared to 354.0674 obtained by Algorithm 1. Our simple indirect method is thus competitive with OpenOCL in terms of accuracy, and it is transparent to the user (in fact, all optimization

parameters are accessible). However, our code not being optimized, OpenOCL currently outperforms Algorithm 1 in terms of total CPU time (about 2 s). The minimum-energy trajectory of the quadrotor computed with OpenOCL v7 (red) and with Algorithm 1 (blue), is shown in Fig. 4(b). The final body orientation  $\theta$  is  $18.3^\circ$  and  $18.9^\circ$ , respectively.

We conclude this section with two remarks. From Table II, we see that the selection of a fixed end-time has a significant impact on energy consumption. The Free End-Time formulation is always preferable, with an energy saving ranging from 2.12% (**Case 5**) to 21.29% (**Case 2**), with respect to the Fixed End-Time case with  $T = T_{\min}$ . This is not surprising, since a free end-time offers more flexibility to the numerical solver. Second, from Figs. 3(a) and 3(b), we notice that the time-evolution of the  $x$ - and  $z$ -coordinates of the quadrotor, is *sigmoidal*. In future works, we will take advantage of this property by assuming a sigmoid profile beforehand, and optimizing its parameters. Although this solution comes with no certificate of optimality, it will enable real-time performance on the microcontroller of the quadrotor.

## VII. CONCLUSION AND FUTURE WORK

In this paper, we have presented a practical and accurate numerical algorithm to solve the free and fixed end-time minimum-energy optimal control problems for a planar quadrotor flying between two rest configurations.

Since our planar model only captures the vertical dynamics, in future research we plan to extend our results to a full 6-DOF multi-rotor model. We also aim at transforming problem (27) into an equivalent polynomial OCP [43], [44], and at generalizing our point-to-point formulation by considering multiple intermediate waypoints (cf. [45]). Finally, further work is needed to formally characterize the optimality and computational complexity of Algorithm 1.

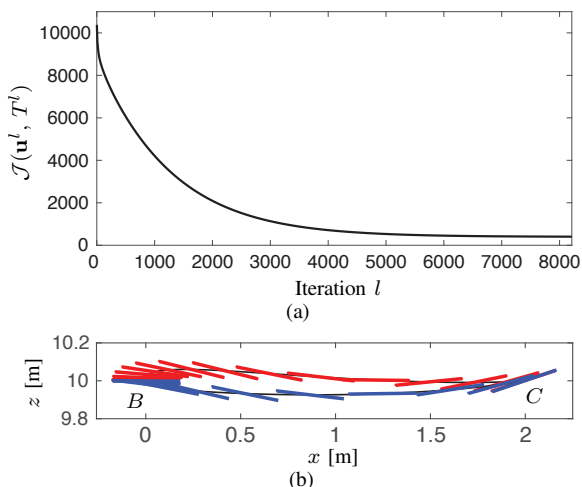


Fig. 4: (a) **Case 2**, Free End-Time OCP: Evolution of  $\mathcal{J}(\mathbf{u}^l, T^l)$  during the optimization process; (b) **Case 3**, Fixed End-Time OCP with  $T = 2.5$  s: Minimum-energy trajectory of the quadrotor computed with OpenOCL (red) and with Algorithm 1 (blue).

## REFERENCES

- [1] D. Floreano and R.J. Wood. Science, technology and the future of small autonomous drones. *Nature*, 521(7553):460–466, 2015.
- [2] S. Choudhury, K. Solovey, M.J. Kochenderfer, and M. Pavone. Efficient Large-Scale Multi-Drone Delivery using Transit Networks. *J. Artif. Intell. Res.*, 70:757–788, 2021.
- [3] T. Nägeli, J. Alonso-Mora, A. Domahidi, D. Rus, and O. Hilliges. Real-Time Motion Planning for Aerial Videography With Dynamic Obstacle Avoidance and Viewpoint Optimization. *IEEE Robot. Autom. Lett.*, 2(3):1696–1703, 2017.
- [4] K. Shah, G. Ballard, A. Schmidt, and M. Schwager. Multidrone aerial surveys of penguin colonies in Antarctica. *Sci. Robot.*, 5(47):eabc3000, 2020.
- [5] P. Radoglou-Grammatikis, P. Sarigiannidis, T. Lagkas, and I. Moscholios. A compilation of UAV applications for precision agriculture. *Comput. Netw.*, 172:107148, 2020.
- [6] S. Song, D. Kim, and S. Jo. Online coverage and inspection planning for 3D modeling. *Auton. Robot.*, 44(8):1431–1450, 2020.
- [7] J. Leonard, A. Savvaris, and A. Tsourdos. Energy Management in Swarm of Unmanned Aerial Vehicles. *J. Intell. Robot. Syst.*, 74(1-2):233–250, 2014.
- [8] N. Kemal Ure, G. Chowdhary, T. Toksoz, J.P. How, M. Vavrina, and J. Vian. An Automated Battery Management System to Enable Persistent Missions with Multiple Aerial Vehicles. *IEEE-ASME Trans. Mech.*, 20(1):275–286, 2015.
- [9] A.B. Junaid, Y. Lee, and Y. Kim. Design and implementation of autonomous wireless charging station for rotary-wing UAVs. *Aerosp. Sci. Technol.*, 54:253–266, 2016.
- [10] G.M. Plaizier, E. Andersen, B. Truong, X. He, S. Roundy, and K.K. Leang. Design, Modeling, and Analysis of Inductive Resonant Coupling Wireless Power Transfer for Micro Aerial Vehicles (MAVs). In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 6104–6109, 2018.
- [11] C. Brommer, D. Malyuta, D. Hentzen, and R. Brockers. Long-Duration Autonomy for Small Rotorcraft UAS including Recharging. In *Proc. IEEE/RSJ Int. Conf. Intel. Robots Syst.*, pages 7252–7258, 2018.
- [12] A. Trotta, M. Di Felice, F. Montori, K.R. Chowdhury, and L. Bononi. Joint Coverage, Connectivity, and Charging Strategies for Distributed UAV Networks. *IEEE Trans. Robot.*, 34(4):883–900, 2018.
- [13] M. Won. UBAT: On Jointly Optimizing UAV Trajectories and Placement of Battery Swap Stations. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 427–433, 2020.
- [14] K.P. Jain and M.W. Mueller. Flying batteries: In-flight battery switching to increase multirotor flight time. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 3510–3516, 2020.
- [15] K. Zhang, P. Chermprayong, T.M. Alhinai, R. Siddall, and M. Kovac. SpiderMAV: Perching and Stabilizing Micro Aerial Vehicles with Bio-inspired Tensile Anchoring Systems. In *Proc. IEEE/RSJ Int. Conf. Intel. Robots Syst.*, pages 6849–6854, 2017.
- [16] K.M. Popek, M.S. Johannes, K.C. Wolfe, R.A. Hegeman, J.M. Hatch, J.L. Moore, K.D. Katyal, B.Y. Yeh, and R.J. Bamberger Jr. Autonomous Grasping Robotic Aerial System for Perching (AGRASP). In *Proc. IEEE/RSJ Int. Conf. Intel. Robots Syst.*, pages 6220–6225, 2018.
- [17] C. Huang, Y. Liu, and X. Ye. Design, simulation and experimental study of a force observer for a flying–perching quadrotor. *Robot. Autom. Syst.*, 120:103237, 2019.
- [18] K. Hang, X. Lyu, H. Song, J.A. Stork, A.M. Dollar, D. Kragic, and F. Zhang. Perching and resting – A paradigm for UAV maneuvering with modularized landing gears. *Sci. Robot.*, 4(28):6637, 2019.
- [19] R. Kitchen, N. Bierwolf, S. Harbertson, B. Platt, D. Owen, K. Griessmann, and M.A. Minor. Design and Evaluation of a Perching Hexacopter Drone for Energy Harvesting from Power Lines. In *Proc. IEEE/RSJ Int. Conf. Intel. Robots Syst.*, pages 1192–1198, 2020.
- [20] S. Park, D.S. Drew, S. Follmer, and J. Rivas-Davila. Lightweight High Voltage Generator for Untethered Electroadhesive Perching of Micro Air Vehicles. *IEEE Robot. Autom. Lett.*, 5(3):4485–4492, 2020.
- [21] M. Burri, L. Gasser, M. Käch, M. Krebs, S. Laube, A. Ledergerber, D. Meier, R. Michaud, L. Mosimann, L. Müri, C. Ruch, A. Schaffner, N. Vuillomenet, J. Weichart, K. Rudin, S. Leutenegger, J. Alonso-Mora, R. Siegwart, and P. Beardsley. Design and Control of a Spherical Omnidirectional Blimp. In *Proc. IEEE/RSJ Int. Conf. Intel. Robots Syst.*, pages 1873–1879, 2013.
- [22] S.H. Song, G.Y. Yeon, H.W. Shon, and H.R. Choi. Design and Control of Soft Unmanned Aerial Vehicle ‘S-CLOUD’. *IEEE/ASME Trans. Mechatron.*, 26(1):267–275, 2021.
- [23] F. Morbidi, R. Cano, and D. Lara. Minimum-Energy Path Generation for a Quadrotor UAV. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 1492–1498, 2016.
- [24] F. Morbidi, D. Bicego, M. Ryll, and A. Franchi. Energy-Efficient Trajectory Generation for a Hexarotor with Dual-Tilting Propellers. In *Proc. IEEE/RSJ Int. Conf. Intel. Robots Syst.*, pages 6226–6232, 2018.
- [25] S. Spedicato and G. Notarstefano. Minimum-Time Trajectory Generation for Quadrotors in Constrained Environments. *IEEE Trans. Contr. Syst. Tech.*, 26(4):1335–1344, 2018.
- [26] I. Spasojevic, V. Murali, and S. Karaman. Perception-aware time optimal path parameterization for quadrotors. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 3213–3219, 2020.
- [27] G. Tang, W. Sun, and K. Hauser. Enhancing Bilevel Optimization for UAV Time-Optimal Trajectory using a Duality Gap Approach. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 2515–2521, 2020.
- [28] A. Nikhilraj, H. Simha, and H. Priyadarshan. Optimal Energy Trajectory Generation for a Quadrotor UAV Using Geometrically Exact Computations on SE(3). *IEEE Contr. Syst. Lett.*, 3(1):216–221, 2019.
- [29] M. Szmuk, C.A. Pascucci, and B. Açıkmeşe. Real-Time Quadrotor Path Planning for Mobile Obstacle Avoidance Using Convex Optimization. In *Proc. IEEE/RSJ Int. Conf. Intel. Robots Syst.*, pages 5906–5911, 2018.
- [30] A. Tagliabue, X. Wu, and M.W. Mueller. Model-free Online Motion Adaptation for Optimal Range and Endurance of Multicopters. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 5650–5656, 2019.
- [31] X. Wu and M.W. Mueller. In-flight range optimization of multicopters using multivariable extremum seeking with adaptive step size. In *Proc. IEEE/RSJ Int. Conf. Intel. Robots Syst.*, pages 1545–1550, 2020.
- [32] G. Tang, W. Sun, and K. Hauser. Learning trajectories for real-time optimal control of quadrotors. In *Proc. IEEE/RSJ Int. Conf. Intel. Robots Syst.*, pages 3620–3625, 2018.
- [33] S. Li, E. Öztürk, C. De Wagter, G.C. de Croon, and D. Izzo. Aggressive Online Control of a Quadrotor via Deep Network Representations of Optimality Principles. In *Proc. IEEE Int. Conf. Robot. Automat.*, pages 6282–6287, 2020.
- [34] J.T. Betts. Survey of Numerical Methods for Trajectory Optimization. *J. Guid. Control Dynam.*, 21(2):193–207, 1998.
- [35] D.E. Kirk. *Optimal Control Theory: An Introduction*. Dover Publications, 2004.
- [36] A. Hussu. The conjugate-gradient method for optimal control problems with undetermined final time. *Int. J. Control*, 15(1):79–82, 1972.
- [37] D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2nd edition, 1999.
- [38] R. Herzog and K. Kunisch. Algorithms for PDE-constrained optimization. *GAMM-Mitteilungen*, 33(2):163–176, 2010.
- [39] R. Mahony, V. Kumar, and P. Corke. Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor. *IEEE Rob. Autom. Mag.*, 19(3):20–32, 2012.
- [40] A. Bosso, C. Conficoni, D. Raggini, and A. Tilli. A Computational-Effective Field-Oriented Control Strategy for Accurate and Efficient Electric Propulsion of Unmanned Aerial Vehicles. *IEEE/ASME Trans. Mechatron.*, 2020, in press. DOI: 10.1109/TMECH.2020.3022379.
- [41] D. Liberzon. *Calculus of Variations and Optimal Control Theory: A Concise Introduction*. Princeton University Press, 2012.
- [42] J. Koenemann, G. Licitra, M. Alp, and M. Diehl. OpenOCL: Open Optimal Control Library. In *Proc. Robotics Science and Systems, Workshops*, 2019. [web]: <https://openocl.github.io/>.
- [43] J.B. Lasserre, D. Henrion, C. Prieur, and E. Trélat. Nonlinear Optimal Control via Occupation Measures and LMI-Relaxations. *SIAM J. Control Optim.*, 47(4):1643–1666, 2008.
- [44] D. Henrion, J.B. Lasserre, and C. Savorgnan. POCP: A Package for Polynomial Optimal Control Problems. In *Proc. European Contr. Conf.*, pages 1173–1178, 2009.
- [45] C. Aoun, E. Shammass, and N. Daher. Energy-Optimal Tours for Quadrotors to Scan Moth-Infested Trees in Densely-Packed Forests. In *Proc. American Contr. Conf.*, pages 58–63, 2020.