Energy-Efficient Trajectory Planning with B-Splines and the Schoenberg Quasi-Interpolant

Vincent Marguet, Florin Stoican, Ionela Prodan

Energy-Efficient Trajectory Planning with B-Splines and the Schoenberg Quasi-Interpolant

Vincent Marguet[1], Florin Stoican[2] and Ionela Prodan[1]

***Abstract*— Few studies address the challenge of minimizing energy consumption during trajectory generation, particularly in the context of multicopter dynamics, characterized by strong non-linearity. This paper introduces a novel approach by formulating energy consumption as a function of B-splines control points, enabling comprehensive optimization of total energy expenditure throughout the trajectory. To mitigate the complexity of representation, we leverage the Schoenberg quasi-interpolant, which not only facilitates the solution of the optimization problem but also effectively reduces computational overhead. The approach is validated through two distinct scenarios: a conventional trajectory used in precision agriculture, and a scenario involving more aggressive maneuvers.**

***Index Terms*– Energy-efficient trajectories, B-spline parameterization, Schoenberg quasi-interpolant, Multicopter.**

I. Introduction

Generating energy-efficient trajectories for drones is crucial for several reasons [1], [2]: i) it extends flight time on a single battery charge, enhancing mission capabilities; ii) it reduces operational costs, making drone applications more economically viable; iii) it lowers carbon emissions, aligning with sustainability goals; and iv) it allows drones to carry heavier payloads or additional sensors without sacrificing flight time or maneuverability.

However, there are several challenges when generating energy-efficient trajectories [3]: i) dynamic environments demand sophisticated trajectory planning algorithms to ensure safe flight while minimizing energy usage; ii) trajectories must meet various mission constraints, such as obstacle avoidance and altitude limits, complicating the generation process; iii) optimizing energy-efficient trajectories involves solving complex optimization problems, often requiring significant computational resources, especially for real-time applications; iv) achieving a balance between energy efficiency and other performance metrics, like flight and mission completion times, requires careful optimization and trade-off consideration.

Formulating energy consumption precisely presents a challenge due to its computational complexity, leading to various approximations in the literature to simplify the optimization problem. Typically, in works like [4], [5], and more recently, the rulebook of the ICUAS2024 conference competition [6], it is often assumed, for simplicity, that minimizing time or travelled distance will also minimize energy consumption. While this approach offers computationally straightforward formulas, it may not always reflect real-world scenarios accurately. Alternatively, [7] and [8] aim to minimize the integral of the square of the snap's norm, leveraging the fact that inputs (net body force and body moments) can be expressed using the first four derivatives of position. In contrast, [9] provides an insightful overview of drone energy consumption factors and models, detailing dynamic factors impacting energy consumption along with environmental influences (e.g., wind, temperature) and drone design considerations. However, it does not offer a specific energy formula. Meanwhile, [10] studies energy consumption into two distinct cases: straight-and-level flight and banked turn maneuvers, each with its unique energy requirements. Furthermore, [2] states that for non-aggressive flights, energy consumption is proportional to thrust. Finally, [11] begins with voltage and current data to derive an expression dependent on motor angular acceleration and related parameters, tailored to motor and propeller geometry. Subsequently, under certain assumptions, it simplifies the energy formulation.

This paper revisits the energy formulations discussed in the literature and introduces a novel approach by formulating energy consumption based on the B-spline control points defining the trajectory. The optimization problem is structured with the control points serving as decision variables [12], [13]. To manage the complexity of representation, we employ the Schoenberg quasi-interpolant [14] to reduce computational overhead. This extends our prior work in [15], where we applied the Schoenberg operator to approximate a simpler cost function and constraints for a fixed-wing aircraft. B-spline functions are widely used in trajectory generation [16], [17], and environmental mapping [18]. However, the synergistic application of their properties with the Schoenberg quasi-interpolant to address energy consumption issues in a comprehensive manner has not, to our knowledge, been explored before. We analyze in-depth and compare the discussed formulas to demonstrate the effectiveness of our approach in minimizing energy consumption. Our approach is validated by simulation results from two scenarios: one representing a conventional trajectory typical in precision agriculture, and another involving more aggressive maneuvers.

Section II reviews B-spline concepts, including the

* This work is supported by the French National Research Agency in the framework of the "France 2030" program (ANR-15-IDEX-0002) and by the LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01) and by the GAR2023 Research Grant funded and managed by the Patrimony Foundation (Fundatia "Patrimoniu") of the Romanian Academy, from the Recurrent Fund of Donors, Contract No. 260/28.11.2023.

[1] Univ. Grenoble Alpes, Grenoble INP†, LCIS, F-26000, Valence, France {vincent.marguet, ionela.prodan}@lcis.grenoble-inp.fr,

† Institute of Engineering and Management Univ. Grenoble Alpes.

[2] Faculty of Automatic Control and Computers, Politehnica University of Bucharest, ACSE, Bucharest, Romania, florin.stoican@upb.ro

Schoenberg quasi-interpolant. Section III presents the flatness-based model of a multicopter. Section IV states the main goal. Section V demonstrates the approach through simulation scenarios and solves the proposed optimization problem. Section VI concludes the paper.

We consistently employ the following notations: $t \in [0, T_f]$ denotes the time instant of the flight. The B-spline basis functions $B_{k,p,\xi}(t)$ (ξ can be omitted for brevity) are collected in the matrix $\mathbf{B}_{p,\xi}(t)$ and have an order p, associated with the knot vector ξ. The knot vector consists of $m+1$ time instants denoted τ_k. $\mathbf{P}$ is the matrix composed of n control points $\mathbf{P}_k$. x, $\dot{x}$, $\ddot{x}$, $\dddot{x}$ and $x^{(4)}$ refer to a B-spline trajectory on the x-axis and its derivatives. The transpose of a matrix $\mathbf{X}$ is denoted $\mathbf{X}^\top$, $\mathbf{M}_{p,p-r}$ is the matrix used to calculate the derivatives of the B-splines up to order r. g is the gravitational acceleration and $\mathbf{p}_\ell^{wp}$ refers to the position of the ℓ-th waypoint through which the multicopter must pass at time t_ℓ.

II. Prerequisites on B-splines functions

Let us consider a trajectory denoted by $\mathbf{z}(t)$. Its parametrization is given by a linear combination of control points, $\mathbf{P} = \begin{bmatrix}\mathbf{P}_0 & \dots & \mathbf{P}_{n-1}\end{bmatrix}$, and B-spline basis functions [19], $\mathbf{B}_{p,\xi}(t) = \begin{bmatrix}B_{0,p,\xi}(t) & \dots & B_{n-1,p,\xi}(t)\end{bmatrix}^\top$:

$$\mathbf{z}(t) = \sum_{k=0}^{n-1} \mathbf{P}_k B_{k,p,\xi}(t) = \mathbf{P}\mathbf{B}_{p,\xi}(t), \forall t \in [0, T_f], \quad (1)$$

with $\xi = \{\tau_0 \leq \tau_1 \leq ... \leq \tau_m\}$ a knot sequence[1] starting at 0 and ending at T_f and the k-th B-spline basis function of order p defined recursively by:

$$B_{k,1,\xi}(t) = \begin{cases} 1, & t \in [\tau_k; \tau_{k+1}], \\ 0, & \text{otherwise}, \end{cases} \quad (2a)$$

$$B_{k,p,\xi}(t) = \frac{t - \tau_k}{\tau_{k+p} - \tau_k} B_{k,p-1,\xi}(t) + \\ + \frac{\tau_{k+p+1} - t}{\tau_{k+p+1} - \tau_{k+1}} B_{k+1,p-1,\xi}(t). \quad (2b)$$

This family of functions enjoys many properties of practical use in motion planning [5]:
P1) Each B-spline basis function has a local support:

$$B_{k,p,\xi}(t) = 0, \forall t \notin [\tau_k; \tau_{k+p+1}). \quad (3)$$

P2) The B-spline functions partition the unity:

$$\sum_{k=0}^{n-1} B_{k,p,\xi}(t) = 1, \forall t \in [\tau_0; \tau_m] \quad (4a)$$

P3) The 'r' order derivatives of B-spline basis functions are linear combinations of B-splines of lower order, i.e. there exists a matrix $\mathbf{M}_{p,p-r}$ such that:

$$\mathbf{B}_{\mathbf{p},\xi}{}^{(r)}(t) = \mathbf{M}_{p,p-r}\mathbf{B}_{\mathbf{p}-\mathbf{r},\xi}(t). \quad (5)$$

[1]If $m \geq p+2$, B-splines of order up to p over the knot sequence can be defined.

P4) The B-spline curve (1) lies within the union of all convex hulls defined by subsets of p consecutive control points. In particular, repeating the first and last p knot values forces the curve to pass through its first and last control point, hence, making it a *clamped B-spline curve*. For further use, note that we take $n = m - p + 1$.

The *Schoenberg quasi-interpolant* [14], [19], [20] approximates a function (call it $f(\cdot)$) by interpolating it as weighted sum of B-splines basis functions, $\tilde{B}_{k,\tilde{p},\tilde{\xi}}(t)$, and the function's values in the Greville points, $\tilde{t}_k$:

$$\tilde{f}(t) = \sum_{k=0}^{\tilde{n}-1} f(\tilde{t}_k)\tilde{B}_{k,\tilde{p},\tilde{\xi}}(t),\ \tilde{t}_k = \frac{\tilde{\tau}_{k+1} + ... + \tilde{\tau}_{k+\tilde{p}-1}}{\tilde{p}-1}. \quad (6)$$

Even if $f(\cdot)$ is itself a combination of B-splines, it is not necessary that the same family of B-splines is used in (6), hence, $\tilde{n}, \tilde{p}, \tilde{\xi}$ may differ from n, p, ξ.

The rationale for using the Schoenberg operator is dual: on one hand it has excellent approximation properties and on the other it significantly alleviates conservatism and computational complexity in the constrained optimization problem in which it is integrated.

III. Full multicopter flat representation

In the literature, multicopter dynamics are often modeled using flatness-based model inversion [21], as it provides a comprehensive representation of all states and inputs in terms of a flat output[2]. However, existing literature typically omits the full flat representation of the multicopter model. Here, instead of recapitulating the multicopter model from the literature [22], we present its complete flat representation, which is crucial for accurately formulating energy consumption. Additionally, while the chain of relations is often implied in other papers, to our knowledge, the torques are never explicitly considered in terms of the flat output.

Consider the multicopter model[3] from [22, page 25]:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}), \quad (7)$$

where the inputs are the thrust and the torques, $\mathbf{u} = [T\ \ \tau_\phi\ \ \tau_\theta\ \ \tau_\psi]^\top$, and the states are the position and velocity on each axis, the roll, pitch and yaw angles, and the angular velocities of the multicopter: $\mathbf{x} = [x\ y\ z\ \dot{x}\ \dot{y}\ \dot{z}\ \phi\ \theta\ \psi\ \omega_x\ \omega_y\ \omega_z]^\top$. We take the flat output usually considered in the literature, the position and the yaw angle $\mathbf{z} = [x\ y\ z\ \psi]^\top$. Consequently, all the inputs and states will be ultimately expressed in function of the flat output and its derivatives (up to the fourth order):

$$\mathbf{x} = f_{\mathbf{x}}(\mathbf{z}, \dot{\mathbf{z}}, \ddot{\mathbf{z}}, \dddot{\mathbf{z}}), \qquad \mathbf{u} = f_{\mathbf{u}}\left(\mathbf{z}, \dot{\mathbf{z}}, \ddot{\mathbf{z}}, \dddot{\mathbf{z}}, \mathbf{z}^{(4)}\right). \quad (8)$$

[2]A flat output is a variable that can be expressed explicitly in terms of the systems' states, inputs and a finite number of their derivatives, without needing to solve complex equations or deal with implicit relationships [21].

[3]For compactness and due to convoluted relations we omit the time dependence in the dynamical model and its flat representation.

Let us explicitly express each state and input variable in terms of the aforementioned flat outputs, x, y, z, ψ:

$$\phi = \arcsin\left(\Phi_x s\psi - \Phi_y c\psi\right), \tag{9a}$$

$$\theta = \arctan\left(\Theta_x c\psi + \Theta_y s\psi\right), \tag{9b}$$

$$\omega_x = \frac{\dot{\Phi}_x s\psi - \dot{\Phi}_y c\psi}{c\phi}, \tag{9c}$$

$$\omega_y = c\phi c^2\theta\left(\dot{\Theta}_x c\psi + \dot{\Theta}_y s\psi\right), \tag{9d}$$

$$\omega_z = -s\phi c^2\theta\left(\dot{\Theta}_x c\psi + \dot{\Theta}_y s\psi\right) + \frac{c\theta}{c\phi}\dot{\psi}, \tag{9e}$$

$$T = m\sqrt{\ddot{x}^2 + \ddot{y}^2 + (\ddot{z} + g)^2}, \tag{9f}$$

$$\tau_\phi = J_x\dot{\omega}_x + (J_z - J_y)\omega_y\omega_z, \tag{9g}$$

$$\tau_\theta = J_y\dot{\omega}_y + (J_x - J_z)\omega_z\omega_x, \tag{9h}$$

$$\tau_\psi = J_z\dot{\omega}_z + (J_y - J_x)\omega_x\omega_y, \tag{9i}$$

where c and s are shorthands for cos and sin, respectively. Due to the length of the intermediary relations, the details of the calculations in (9) are in Appendix I. Lastly, we express the angular velocities of each propeller as functions of the thrust and torques:

$$\begin{pmatrix}\Omega_1^2\\ \Omega_2^2\\ \Omega_3^2\\ \Omega_4^2\end{pmatrix} = \begin{pmatrix}\frac{1}{4K_b} & 0 & \frac{-1}{2LK_b} & \frac{-1}{4\kappa_\tau}\\ \frac{1}{4K_b} & \frac{-1}{2LK_b} & 0 & \frac{1}{4\kappa_\tau}\\ \frac{1}{4K_b} & 0 & \frac{1}{2LK_b} & \frac{-1}{4\kappa_\tau}\\ \frac{1}{4K_b} & \frac{1}{2LK_b} & 0 & \frac{1}{4\kappa_\tau}\end{pmatrix}\begin{pmatrix}T\\ \tau_\phi\\ \tau_\theta\\ \tau_\psi\end{pmatrix}, \tag{10}$$

with L the distance from the center of the quadcopter to any of the propellers, κ_τ the drag coefficient and K_b a coefficient depending on the area swept out by the rotor, the density of the surrounding air, and other proportionality constants assumed to be known.

We may thus conclude that (9) and (10) express in a convoluted but explicit manner, the angular velocity of the propellers in function of the system's flat outputs.

IV. Energy-efficient trajectories

Trajectory planning frequently involves solving a constrained optimization problem such as:

$$\min_{\mathbf{z}(t)} \quad \int_0^{T_f} \mathcal{E}(\mathbf{z}(t))\mathrm{d}t \tag{11a}$$

$$\text{s.t.} \quad g_i(\mathbf{z}(t)) \leq 0,\ 0 \leq t \leq T_f, \tag{11b}$$

$$h_j(\mathbf{z}(t)) = 0,\ 0 \leq t \leq T_f. \tag{11c}$$

The objective is to find a trajectory $\mathbf{z}(t)$ which minimizes the cost $\mathcal{E}(\cdot)$ from (11a) while simultaneously respecting (11b) and (11c) over the time interval $[0, T_f]$.

To better illustrate the subsequent ideas let us particularize (11) by taking $\mathbf{z}(t)$ as in (1). Then, we have:

- the cost $\mathcal{E}(\cdot)$ is taken as the energy consumed along the time horizon $[0, T_f]$ and has to be minimized;
- the bounds on the thrust magnitude are denoted by $\underline{T}$ and $\overline{T}$, respectively; similarly, the bounds on the torques are denoted by $\underline{\tau_\phi}$, $\underline{\tau_\theta}$, $\underline{\tau_\psi}$ and $\overline{\tau_\phi}$, $\overline{\tau_\theta}$, $\overline{\tau_\psi}$ respectively; all grouped into inequalities (11b);
- passing through waypoints $\mathbf{p}_\ell^{wp}$ at times t_ℓ serves as constraint equalities (11c).

Thus, (11) is instantiated to the particular form:

$$\min_{\mathbf{z}(t)} \int_0^{T_f} \mathcal{E}(\mathbf{z}(t))\mathrm{d}t \tag{12a}$$

$$\text{s.t.} \quad \underline{T} \leq T(t) \leq \overline{T},\ \forall t \in [0, T_f] \tag{12b}$$

$$\underline{\tau_\phi} \leq \tau_\phi(t) \leq \overline{\tau_\phi},\ \forall t \in [0, T_f] \tag{12c}$$

$$\underline{\tau_\theta} \leq \tau_\theta(t) \leq \overline{\tau_\theta},\ \forall t \in [0, T_f] \tag{12d}$$

$$\underline{\tau_\psi} \leq \tau_\psi(t) \leq \overline{\tau_\psi},\ \forall t \in [0, T_f] \tag{12e}$$

$$\mathbf{z}(t_\ell) = \mathbf{p}_\ell^{wp}, \forall \ell. \tag{12f}$$

Except very specific cases (e.g., optimal control approaches for simple dynamics which allow analytic solutions) it is often not possible to numerically solve (11) or (12), for an arbitrary $\mathbf{z}(t)$. The usual approach is to project $\mathbf{z}(t)$ onto a basis of functions and reformulate (12), into a more manageable discrete optimization problem.

While the terms appearing in the constraints of (12) have already been described in (9) and (10), we are still left with the cost expression. The literature considers several approaches to approximate the energy consumption:

i) proportional to the flying time [6]:

$$E_0 = \gamma_0 T_f. \tag{13}$$

ii) proportional to the travelled distance [5]:

$$E_1 = \gamma_1 \int_0^{T_f} \|\dot{\mathbf{z}}(t)\|_2 \mathrm{d}t. \tag{14}$$

iii) proportional to the integral of the square of the norm of the snap [7]:

$$E_2 = \gamma_2 \int_0^{T_f} \|\mathbf{z}^{(4)}(t)\|_2^2 \mathrm{d}t. \tag{15}$$

iv) proportional to the integral of the thrust [2]:

$$E_3 = \gamma_3 \int_0^{T_f} T(t)\mathrm{d}t. \tag{16}$$

v) directly, as the power drawn by the motors [11]:

$$E_4 = \int_0^{T_f} \sum_{i=1}^{4}\Big[c_1 + c_2\Omega_i(t) + c_3\Omega_i^2(t) + c_4\Omega_i^3(t) + c_5\Omega_i^4(t) + c_6\dot{\Omega}_i(t) + c_7\dot{\Omega}_i^2(t) + c_8\Omega_i(t)\dot{\Omega}_i(t) + c_9\Omega_i^2(t)\dot{\Omega}_i(t)\Big]\mathrm{d}t, \tag{17}$$

with constants c_i specific to the motors' parameters and the propellers' geometry:

$$c_1 = \frac{RT_g^2}{K_T^2},\ c_2 = \frac{T_g}{K_T}\left(\frac{2RD_f}{K_T} + K_E\right),$$

$$c_3 = \frac{D_f}{K_T}\left(\frac{RD_f}{K_T} + K_E\right) + \frac{2RT_g\kappa_\tau}{K_T^2},$$

$$c_4 = \frac{\kappa_\tau}{K_T}\left(\frac{2RD_f}{K_T} + K_E\right),\ c_5 = \frac{R\kappa_\tau^2}{K_T^2},\ c_6 = \frac{2RJT_g}{K_T^2}$$

$$c_7 = \frac{RJ^2}{K_T^2},\ c_8 = \frac{J}{K_T}\left(\frac{2RD_f}{K_T} + K_E\right),\ c_9 = \frac{2RJ\kappa_\tau}{K_T^2}.$$

Arguably, (13)–(17) offer progressively better approximations of energy consumption (with (17) being the exact formula) but at the price of increasingly complex expressions in terms of the initial flat outputs $\mathbf{z}(t)$. In what follows we integrate a modified form of (17), simplified through the Schoenberg operator. Let us consider

$$f_c(t) = \sum_{i=1}^{4} c_1 + c_2\Omega_i(t) + c_3\Omega_i^2(t) + c_4\Omega_i^3(t) + c_5\Omega_i^4(t), \quad (18)$$

which denotes the instantaneous energy cost and where, under the assumptions that the initial and final angular velocities of each motor are identical and that the terms associated to the angular accelerations are negligible, constants $c_6, \ldots, c_9$ are discarded. To (18) corresponds

$$E_{4c} = \int_0^{T_f} f_c(t)\mathrm{d}t. \quad (19)$$

Finally, applying the Schoenberg operator (6) to (19), we obtain an approximation of the energy consumption

$$E_5 = \gamma_5 \int_0^{T_f} \sum_{k=0}^{\tilde{n}-1} f_c(\tilde{t}_k)\tilde{B}_{k,\tilde{p},\tilde{\xi}}(t)\mathrm{d}t, \quad (20)$$

$$= \gamma_5 \sum_{k=0}^{\tilde{n}-1} f_c(\tilde{t}_k) \int_0^{T_f} \tilde{B}_{k,\tilde{p},\tilde{\xi}}(t)\mathrm{d}t. \quad (21)$$

Note that (21) allows to tune the complexity of the optimization problem in a controllable manner: by the B-spline basis size ($\tilde{n}$) we control the complexity of the cost; $\tilde{n}$ and the remaining parameters $\tilde{p}, \tilde{\xi}$ influence the approximation error. Not least, the 'good' approximations (16) and (17), need to be integrated numerically (e.g., by computing a Gauss quadrature approximation) whereas (21) is derived without any loss of precision from (20). The resolution of (11) may be described through a construction similar to the one from [15, Algorithm 1].

V. Validation through simulation scenarios

A. Energy profiles approximations

We consider two scenarios with the parameters from Table I. In both cases, we analyze the actual energy expenditure as given by (17) versus the various ways to approximate it for an a priori computed trajectory.

TABLE I: Parameters of the Phantom 2 drone as in [11].

m	1.3 kg	R	0.2 Ω
J_x	0.081 kg.m^2	T_g	4.10^{-2} N.m
J_y	0.081 kg.m^2	D_f	2.10^{-4} N.m.s.rad^{-1}
J_z	0.142 kg.m^2	K_E	$1.04.10^{-2}$V.s.rad^{-1}
L	0.175 m	K_T	$1.04.10^{-2}$V.s.rad^{-1}
K_b	$3.83.10^{-6}$ kg.m	J	$4.19.10^{-5}$ kg.m^2
κ_τ	$2.25.10^{-8}$ kg.m^2		

Scenario 1: Non aggressive flight

This is a trajectory typical in precision agriculture applications. It is inspired from [6] where a multicopter needs to visit beds and count the number of fruits from different types (tomato, eggplant and pepper). As some fruits may be seen from only one side of the beds, the multicopter needs to pass to both side and compare the position of the fruits that it is detecting. For simplification, only one layer composed of 3×3 beds is considered as shown in Figure 1 and we assume that the objective is to count the number of fruits of each type in all the beds. Constraints on the multicopter's position, velocity, acceleration and waypoint passing are considered. The allowed flying volume is restricted to a box representing the indoor environment from which a smaller box including the layer composed of the beds is removed. For this application, the trajectory can be precomputed off-line and followed on-line. We focus on trajectory generation aimed at minimizing the energy expended by the multicopter. We address this optimization problem by considering a B-spline curve of degree 6 with 50 control points. The 20 waypoints that the multicopter should visit at time t_ℓ are listed in Table II.

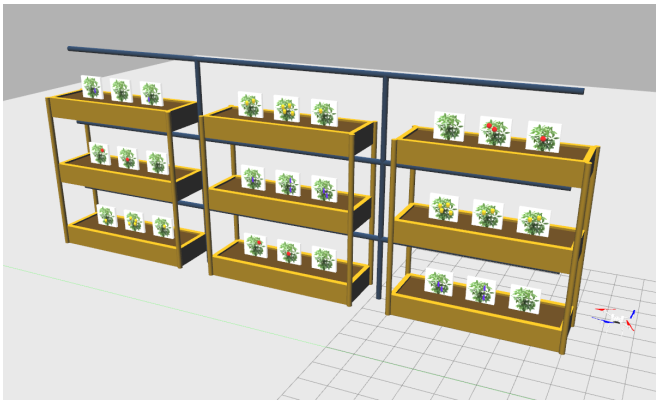


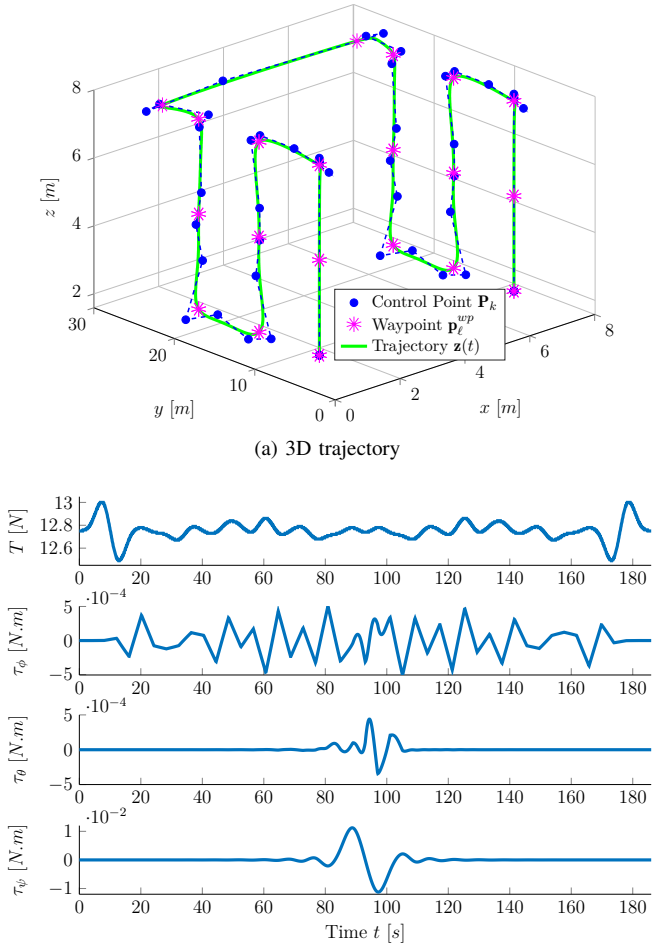
Fig. 1: Arena with the beds containing the fruits (Scenario 1).

TABLE II: Waypoint list (Scenario 1).

ℓ	1	2	3	4	5	6	7	8	9	10
$\mathbf{p}_\ell^{wp}$	1	1	1	1	1	1	1	1	1	1
	6	6	6	13.5	13.5	13.5	21	21	21	25.5
	2.05	4.85	7.65	7.65	4.85	2.05	2.05	4.85	7.65	7.65
	0	0	0	0	0	0	0	0	0	0
t_ℓ[s]	0	10	20	30	40	50	60	70	80	88
ℓ	11	12	13	14	15	16	17	18	19	20
$\mathbf{p}_\ell^{wp}$	7	7	7	7	7	7	7	7	7	7
	25.5	21	21	21	13.5	13.5	13.5	6	6	6
	7.65	7.65	4.85	2.05	2.05	4.85	7.65	7.65	4.85	2.05
	π	π	π	π	π	π	π	π	π	π
t_ℓ[s]	98	106	116	126	136	146	156	166	176	186

The trajectory obtained off-line satisfying the optimization problem is depicted in Figure 2a. Figure 2b shows the profiles of the thrust and torques during the simulation. It is notable that the thrust significantly exceeds the torques for this non aggressive flight, which explains why the energy evolution computed using the thrust-based formula, (16), closely approximates the actual energy consumption (17) in Figures 3a and 3b. Note that since E_0, E_3, E_4 and E_5 are all similar, they overlap in Figure 3a. The differences are better observed in Figure 3b. As the flight is non aggressive, the energy is spent most of the time to counteract the weight of

the UAV, which explains why the formula to compute the energy using only the time, (13) is quite good. However,



(a) 3D trajectory

(b) thrust and torques profiles

Fig. 2: Non-aggressive trajectory generation (Scenario 1).

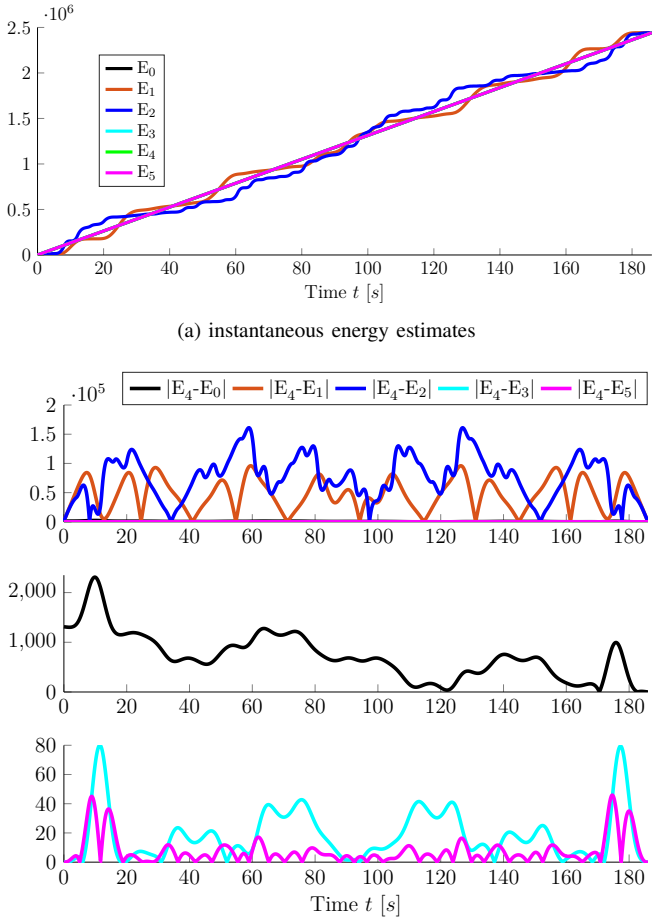
the energy calculated using the snap (15) and the travelled distance (14) exhibit a different profile than the actual energy consumed. The approximation using the Schoenberg operator (21) outputs the smallest error with the real energy consumed. The coefficients $\gamma_0 = 1313.1$, $\gamma_1 = 5061.2$, $\gamma_2 = 1922700$, $\gamma_3 = 102.8996$ and $\gamma_5 = 1$ were computed to obtain the same final value for the consumed energy and used to compare the energy profiles.

Overall, we may conclude that (16) and (21) are reasonable approximations when most of the effort is due to the thrust. In such a scenario, it may be convincingly argued that the thrust integral or, even better, our Schoenberg's approximation give the best result for the required effort.

Scenario 2: Aggressive flight

Having to move quickly between consecutive waypoints and/or having sharp turns will test the capabilities of the multicopter and make it function 'near the limits'. Such an illustrative example is provided in this scenario via the waypoints listed in Table III.

A B-spline curve of degree 6, comprising 31 control



(a) instantaneous energy estimates

(b) relative error estimates with adapted y-axis scale

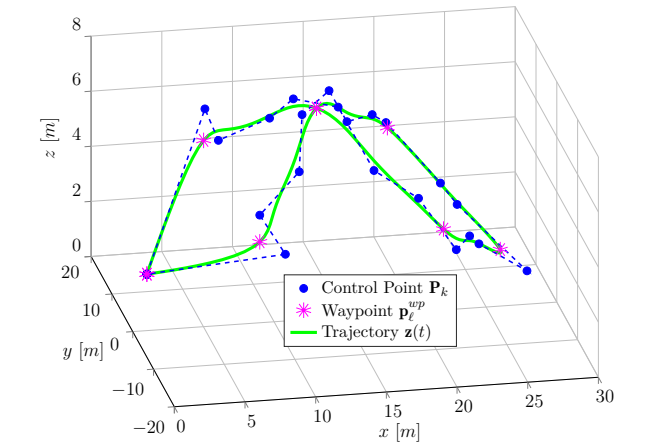
Fig. 3: Energy estimates (Scenario 1).

TABLE III: Waypoint list (Scenario 2).

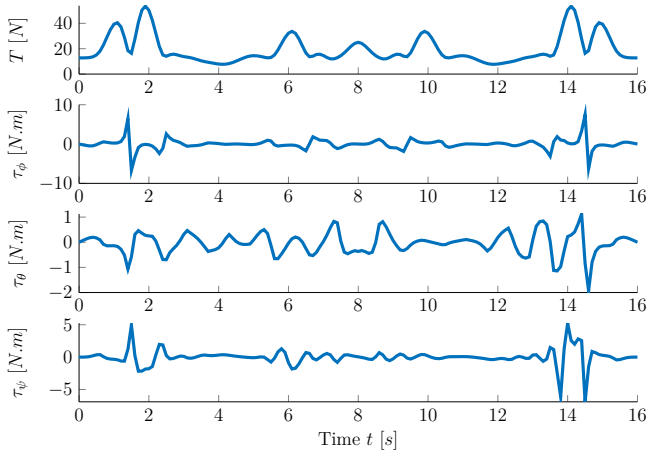
ℓ	1	2	3	4	5	6	7	8	9
$\mathbf{p}_\ell^{wp}$	1	7	13	20	26	20	13	7	1
	0	13.5	0	-13.5	0	13.5	0	-13.5	0
	2.05	4.85	7.65	4.85	2.05	4.85	7.65	4.85	2.05
	0	0	0	π	π	π	π	0	0
t_ℓ[s]	0	2	4	6	8	10	12	14	16

points, is generated to traverse all waypoints (for x, y, z, and ψ) at specified times. The optimization problem is solved offline, and the resulting trajectory of the multicopter is depicted in Figure 4a. Figure 4b shows higher values for thrust and torques, indicating a more aggressive trajectory. The torques, not always negligible compared to thrust, lead to inaccuracies in the energy profile computed using the thrust-based formula (16), as demonstrated in Figure 5a.

This confirms the findings of [2], suggesting that the thrust integral's approximation of energy consumption is valid only for non-aggressive flights. Moreover, for this scenario, our approximation of the energy consumption using the Schoenberg operator (21) is much better than the other approximations as can be seen in Figures 5a and 5b. Coefficients $\gamma_0 = 14158$, $\gamma_1 = 199.6485$, $\gamma_2 = 0.4819$, $\gamma_3 = 728.8035$, and $\gamma_5 = 1.0527$ were post-computed to compare energy profiles.

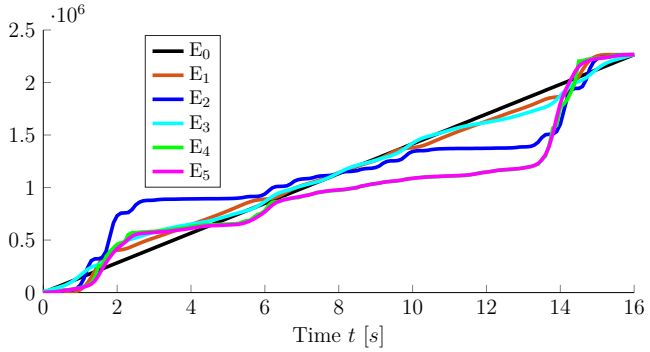


(a) 3D trajectory

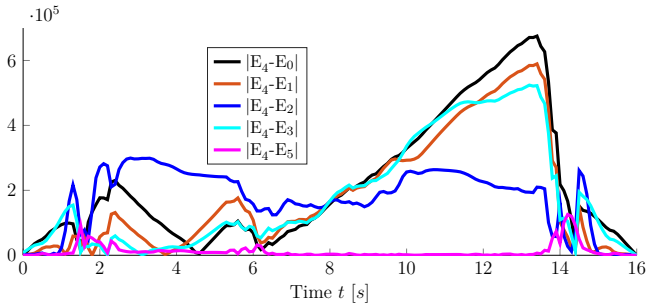


(b) thrust and torques profiles

Fig. 4: Aggressive trajectory generation (Scenario 2).



(a) instantaneous energy estimates



(b) relative error estimates

Fig. 5: Energy estimates (Scenario 2).

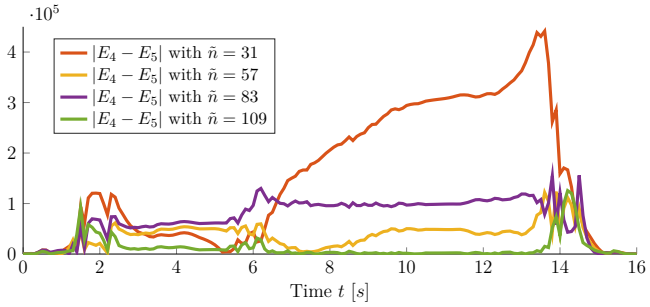


Fig. 6: Energy consumption error computed as in (21) with different $\tilde{n}$ (Scenario 2).

To assess the performance of the approximation. we expanded the basis functions from 31 to 109 for the Schoenberg operator.

In Figure 6, we show the influence of the expansion of the basis functions in the approximation error for the energy consumption. It is shown that the error depends on the number and position of the Greville points in the expected manner: more points lead to a better approximation, at the price of a higher computational time.

B. Energy minimization via optimization (12)

For the optimization problem proposed in (12) we consider the waypoints given in Table III with the associated times multiplied by 2.5 to ensure a trajectory of moderate aggressiveness, lasting $T_f = 40$ s. Initial and final conditions dictate that the multicopter is hovering, meaning that derivatives along the x, y and z axes have zero values. The bounds in (12) are: $\underline{T} = 12.11$ N, $\overline{T} = 13.39$ N, $-\underline{\tau_\phi} = -\underline{\tau_\theta} = -\underline{\tau_\psi} = \overline{\tau_\phi} = \overline{\tau_\theta} = \overline{\tau_\psi} = 0.1$ N.m. The yaw angle profile is prescribed as a given B-spline. The objective is to obtain the control point matrix corresponding to the trajectory along the x, y and z axes that minimizes our energy approximation given in (21), using the Schoenberg operator with 109 Greville points. To fully leverage the Schoenberg approximations, we also approximate the thrust constraint (12b) and the torques constraints (12c), (12d), (12e). Without these approximations, solving the optimization problem proves to be infeasible due to various numerical issues and the complex formulas for thrust and torques. To compensate the approximation error due to the use of the Schoenberg operator, we tightened the bounds on the thrust by 0.1 N, meaning that we imposed the values of the thrust at the Greville points to stay inside the interval $[12.21, 13.29]$ N.

We used CasADi [23], a highly efficient tool for solving nonlinear optimization problems. The IPOPT solver [24] returned a feasible solution after 700 iterations in 405 s using Matlab 2023a. Trajectory and constraints verification are depicted in Figures 7 and 8. It is evident that all constraints, including initial and final conditions, are satisfied as torques have zero values and thrust is equal to mg at the start and end of the figures.

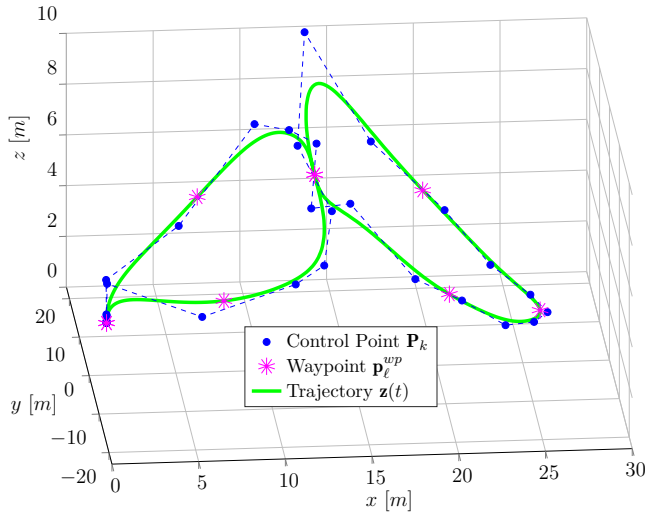


Fig. 7: Waypoint passing trajectory generated with the optimization problem (12f).

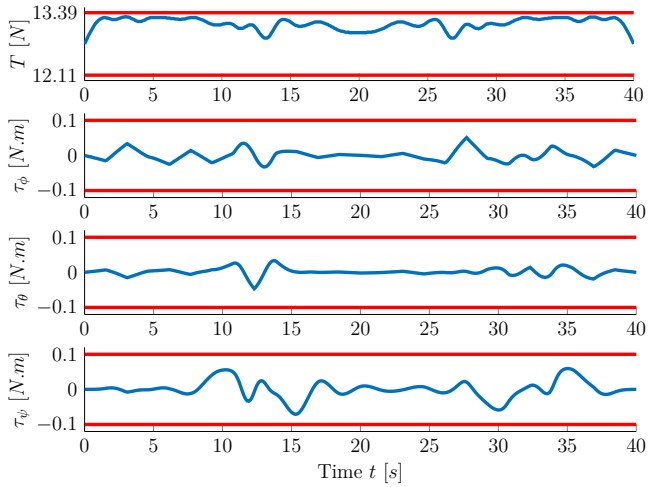


Fig. 8: Constraints satisfaction for thrust and torques, (12b), (12c), (12d), (12e).

VI. Conclusion

We formulated energy consumption as a function of B-spline control points, enabling to optimize the energy expenditure across the entire trajectory. Using the Schoenberg quasi-interpolant, we effectively managed the complexity of representation, reducing computational overhead. Analysis for aggressive and non-aggressive trajectories planning of a multicopter systems validated our approach. These results underscored the potential impact and versatility of our methodology which can be applied to every system admiting a flat representation. Future research will focus on analyzing the approximation error introduced by the Schoenberg operator and extending the algorithm to accommodate multiple multicopters, considering inter-multicopter communication range, obstacle and collision avoidance.

References

[1] Y. Wang, Y. Wang, and B. Ren, "Energy saving quadrotor control for field inspections," *IEEE Tran. on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 3, pp. 1768–1777, 2020.

[2] H. Alkomy and J. Shan, "Investigating the effects of polynomial trajectories on energy consumption of quadrotors," *IEEE/ASME Transactions on Mechatronics*, 2022.

[3] B. Rubí, R. Pérez, and B. Morcego, "A survey of path following control strategies for uavs focused on quadrotors," *Journal of Intelligent & Robotic Systems*, vol. 98, no. 2, pp. 241–265, 2020.

[4] Q. Xia, S. Liu, M. Guo, H. Wang, Q. Zhou, and X. Zhang, "Multi-uav trajectory planning using gradient-based sequence minimal optimization," *Robotics and Autonomous Systems*, vol. 137, p. 103728, 2021.

[5] F. Stoican, I. Prodan, D. Popescu, and L. Ichim, "Constrained trajectory generation for uav systems using a b-spline parametrization," in *25th Mediterranean Conference on Control and Automation*. IEEE, 2017, pp. 613–618.

[6] L. Markovic, F. Petric, A. Ivanovic, J. Goricanec, M. Car, M. Orsag, and S. Bogdan, "Towards a standardized aerial platform: ICUAS'22 firefighting competition," *Journal of Intelligent & Robotic Systems*, vol. 108, no. 3, July 2023.

[7] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Int. Conf. on Robotics and Automation*. IEEE, 2011, pp. 2520–2525.

[8] V. Freire and X. Xu, "Flatness-based quadcopter trajectory planning and tracking with continuous-time safety guarantees," *IEEE Tran. on Control Systems Technology*, 2023.

[9] P. Beigi, M. S. Rajabi, and S. Aghakhani, "An overview of drone energy consumption factors and models," *Handbook of Smart Energy Systems*, pp. 1–20, 2022.

[10] Y. A. Sambo, P. V. Klaine, J. P. B. Nadas, and M. A. Imran, "Energy minimization uav trajectory design for delay-tolerant emergency communication," in *Int. Conference on Communications Workshops*. IEEE, 2019, pp. 1–6.

[11] F. Morbidi, R. Cano, and D. Lara, "Minimum-energy path generation for a quadrotor uav," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1492–1498.

[12] F. Stoican, A. Postolache, and I. Prodan, "Nurbs-based trajectory design for motion planning in a multi-obstacle environment," in *European Control Conference*. IEEE, 2021, pp. 2014–2019.

[13] H.-T. Do, I. Prodan, and F. Stoican, "Analysis of alternative flat representations of a uav for trajectory generation and tracking," in *25th Int. Conf. on System Theory, Control and Computing*. IEEE, 2021, pp. 58–63.

[14] L. Beutel, H. Gonska, D. Kacsó, and G. Tachev, "On variation-diminishing schoenberg operators: new quantitative statements," *Multivariate Approximation and Interpoltaion with Applications*, vol. 20, pp. 9–58, 2002.

[15] V. Marguet, F. Stoican, and I. Prodan, "On the application of the schoenberg quasi-interpolant for complexity reduction in trajectory generation," in *2023 European Control Conference (ECC)*. IEEE, 2023, pp. 1–6.

[16] D. Delahaye, C. Peyronne, M. Mongeau, and S. Puechmorel, "Aircraft conflict resolution by genetic algorithm and b-spline approximation," in *Int. Workshop on ATM/CNS*, 2010, pp. 65–71.

[17] L. Tang, H. Wang, Z. Liu, and Y. Wang, "A real-time quadrotor trajectory planning framework based on b-spline and nonuniform kinodynamic search," *Journal of Field Robotics*, vol. 38, no. 3, pp. 452–475, 2021.

[18] R. T. Rodrigues, N. Tsiogkas, A. Pascoal, and A. P. Aguiar, "Online range-based slam using b-spline surfaces," *Robotics and Automation Letters*, vol. 6, no. 2, pp. 1958–1965, 2021.

[19] T. Lyche, C. Manni, and H. Speleers, "Foundations of spline theory: B-splines, spline approximation, and hierarchical refinement," in *Splines and PDEs: From Approximation Theory to Numerical Linear Algebra*. Springer, 2018, pp. 1–76.

[20] T. Zapryanova and G. Tachev, "Generalized inverse theorem for schoenberg operator," *Journal of Modern Mathematics Frontier*, vol. 1, no. 2, pp. 11–16, 2012.

[21] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of non-linear systems: introductory theory and examples," *Int. Journal of Control*, vol. 61, no. 6, pp. 1327–1361, 1995.

[22] N. T. Nguyen, "Reliable hierarchical control for multicopter systems," Ph.D. dissertation, Université Grenoble Alpes, 2019.

[23] 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.

[24] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, pp. 25–57, 2006.

APPENDIX I
DETAILED MATHEMATICAL MULTICOPTER MODEL

For further use we denote

$$\lambda = \ddot{x}^2 + \ddot{y}^2 + (\ddot{z}+g)^2 \tag{22}$$

We derive first the auxiliary terms Φ_x, Φ_y, Θ_x and Θ_y, used for the Euler angle computations

$$\Phi_x = \ddot{x}/\sqrt{\lambda}, \qquad \Phi_y = \ddot{y}/\sqrt{\lambda}, \tag{23a}$$
$$\Theta_x = \ddot{x}/(\ddot{z}+g), \qquad \Theta_y = \ddot{y}/(\ddot{z}+g). \tag{23b}$$

Next, we obtain first and second order derivatives, for use in the quadcopter's angular velocities (inertial frame) and angular accelerations (body frame)

$$\dot{\Phi}_x = \big[\dddot{x}(\ddot{y}^2 + (\ddot{z}+g)^2) - \ddot{x}(\ddot{y}\dddot{y} + (\ddot{z}+g)\dddot{z})\big]/\lambda^{\frac{3}{2}}, \tag{23c}$$

$$\dot{\Phi}_y = \big[\dddot{y}(\ddot{x}^2 + (\ddot{z}+g)^2) - \ddot{y}(\ddot{x}\dddot{x} + (\ddot{z}+g)\dddot{z})\big]/\lambda^{\frac{3}{2}}, \tag{23d}$$

$$\dot{\Theta}_x = \frac{\dddot{x}(\ddot{z}+g) - \ddot{x}\dddot{z}}{(\ddot{z}+g)^2}, \; \dot{\Theta}_y = \frac{\dddot{y}(\ddot{z}+g) - \ddot{y}\dddot{z}}{(\ddot{z}+g)^2}, \tag{23e}$$

$$\begin{aligned}\ddot{\Phi}_x =& \big[x^{(4)}(\ddot{y}^2 + (\ddot{z}+g)^2) + \ddot{x}(\dddot{y}\ddot{y} + \dddot{z}(\ddot{z}+g)) - \dddot{x}(\dddot{y}^2 \\ &+ \ddot{y}y^{(4)} + \dddot{z}^2 + z^{(4)}(\ddot{z}+g))\big]/\lambda^{\frac{3}{2}} \\ &- \big[3\big(\dddot{x}(\ddot{y}^2 + (\ddot{z}+g)^2) - \ddot{x}(\ddot{y}\dddot{y} + (\ddot{z}+g)\dddot{z})\big) \\ &\cdot(\dddot{x}\ddot{x} + \dddot{y}\ddot{y} + \dddot{z}(\ddot{z}+g))\big]/\lambda^{\frac{5}{2}},\end{aligned} \tag{23f}$$

$$\begin{aligned}\ddot{\Phi}_y =& \big[y^{(4)}(\ddot{x}^2 + (\ddot{z}+g)^2) + \dddot{y}(\dddot{x}\ddot{x} + \dddot{z}(\ddot{z}+g)) - \ddot{y}(\dddot{x}^2 \\ &+ \ddot{x}x^{(4)} + \dddot{z}^2 + z^{(4)}(\ddot{z}+g))\big]/\lambda^{\frac{3}{2}} \\ &- \big[3\big(\dddot{y}(\ddot{x}^2 + (\ddot{z}+g)^2) - \ddot{y}(\ddot{x}\dddot{x} + (\ddot{z}+g)\dddot{z})\big) \\ &\cdot(\dddot{x}\ddot{x} + \dddot{y}\ddot{y} + \dddot{z}(\ddot{z}+g))\big]/\lambda^{\frac{5}{2}},\end{aligned} \tag{23g}$$

$$\begin{aligned}\ddot{\Theta}_x =& \big[(x^{(4)}(\ddot{z}+g) - \ddot{x}z^{(4)})(\ddot{z}+g) \\ &- 2\dddot{z}(\dddot{x}(\ddot{z}+g) - \ddot{x}\dddot{z})\big]/(\ddot{z}+g)^3,\end{aligned} \tag{23h}$$

$$\begin{aligned}\ddot{\Theta}_y =& \big[(y^{(4)}(\ddot{z}+g) - \ddot{y}z^{(4)})(\ddot{z}+g) \\ &- 2\dddot{z}(\dddot{y}(\ddot{z}+g) - \ddot{y}\dddot{z})\big]/(\ddot{z}+g)^3.\end{aligned} \tag{23i}$$

These elements allow to compute the angular velocities (inertial frame)

$$\dot{\phi} = \frac{(\dot{\psi}\Phi_x - \dot{\Phi}_y)c\psi + (\dot{\psi}\Phi_y + \dot{\Phi}_x)s\psi}{\sqrt{1-(\Phi_x s\psi - \Phi_y c\psi)^2}}, \tag{23j}$$

$$\dot{\theta} = \frac{(\dot{\Theta}_x + \dot{\psi}\Theta_y)c\psi + (\dot{\Theta}_y - \dot{\psi}\Theta_x)s\psi}{1+(\Theta_x c\psi + \Theta_y s\psi)^2}, \tag{23k}$$

and the angular accelerations (body frame)

$$\begin{aligned}\dot{\omega}_x =& \big[((\ddot{\Phi}_x + \dot{\psi}\dot{\Phi}_y)s\psi + (\dot{\Phi}_x\dot{\psi} - \ddot{\Phi}_y)c\psi)c\phi \\ &+ \dot{\phi}s\phi(\dot{\Phi}_x s\psi - \dot{\Phi}_y c\psi)\big]/c^2\phi,\end{aligned} \tag{23l}$$

$$\begin{aligned}\dot{\omega}_y =& -(\dot{\phi}s\phi c^2\theta + \dot{\theta}s2\theta c\phi)(\dot{\Theta}_x c\psi + \dot{\Theta}_y s\psi) \\ &+ c\phi c^2\theta(c\psi(\ddot{\Theta}_x + \dot{\psi}\dot{\Theta}_y) + s\psi(\ddot{\Theta}_y - \dot{\psi}\dot{\Theta}_x)),\end{aligned} \tag{23m}$$

$$\begin{aligned}\dot{\omega}_z =& (-\dot{\phi}c\phi c^2\theta + \dot{\theta}s\phi s2\theta)(\dot{\Theta}_x c\psi + \dot{\Theta}_y s\psi) \\ &- s\phi c^2\theta((\ddot{\Theta}_x + \dot{\psi}\dot{\Theta}_y)c\psi + (\ddot{\Theta}_y - \dot{\psi}\dot{\Theta}_x)s\psi) \\ &+ \frac{(\ddot{\psi}c\theta - \dot{\theta}\dot{\psi}s\theta)c\phi + \dot{\psi}\dot{\phi}s\phi c\theta}{c^2\phi}.\end{aligned} \tag{23n}$$