On the Energy Consumption of a Quadcopter Navigating in an Orchard Environment

Florin Stoican, Vincent Marguet, Dan Popescu, Ionela Prodan, Loretta Ichim

2024 32nd Mediterranean Conference on Control and Automation (MED)
June 11-14, 2024 | Chania, Crete, Greece

On the energy consumption of a quadcopter navigating in an orchard environment*

Florin Stoican[1], Vincent Marguet[2], Dan Popescu[1], Ionela Prodan[2], Loretta Ichim[1]

***Abstract*— The problem of efficient motion planning is of significant interest in precision agriculture. We propose a mechanism to estimate it via Bezier function parametrization of the position, velocity and acceleration profiles for a quadcopter system. This is done by providing closed-form descriptions of the cost components, which are further incorporated into the overall motion planning problem to analyze the effect of various tuning parameters (such as total flight time, maximum velocity and acceleration). The ideas are tested over a proof-of-concept orchard navigation setup (challenging due to the close grouping of the way-points) with the final goal being thee comparison between horizontal-first and vertical-first path construction.**

I. Introduction

Precision agriculture is nowadays a hotbed of innovation due to the twin pressures of ever-increasing productivity demand and cost of qualified labor [1], [2]. Hence, the use of robotic platform to augment or even fully replace manual labor is of great interest to the industry. There has been significant work done with ground-based platforms ("mini-trucks", [3]) but more recently, unmanned aerial vehicles (UAVs) have come to the fore. While more limited in payload and autonomy, they counterbalance by greater flexibility and even reduced costs [4], [5].

Within the broad class of UAVs we may separate between fixed-wings and rotary (multicopter) variants. They are complementary in their capacities: while the former has more autonomy (up to tens of kilometers and hours of flight) the later permits a broader range of trajectories (no stall velocity constraint, capacity to hover in place, etc.). For the later, most popular platforms are quadcopters (with four motors, the minimum number to ensure simultaneous position and yaw angle tracking) as they provide the best compromise in availability, price and redundancy.

While intensely studied in recent decades, the issue of motion planning for drones is still an open one [6], due to the many constraints that may appear (of internal dynamics, external requirements, cost and reliability). Most commonly, the user provides a list of way-points from which a path is then derived by ancillary software installed by the vendor on the drone (the "auto-pilot"). The resulting trajectory is then tracked in a given amount of time and with a certain energy consumption [7].

Hereinafter we will consider as specific application the problem of orchard navigation [8] which is more challenging than the typical mission in that it requires to fly inbetween rows and take measurements at specific way-points. These requirements raise challenges in two directions:

i) due to the closely-grouped way-points, the quadcopter dynamics can no longer be ignored (imposing thus limitations on accelerations, velocities and rate of turns);
ii) the cost along the path strongly depends on the order in which the way-points are passed, a combination of either the total time spent and/or the battery consumption has to be considered.

The idea followed in this paper is to propose a smooth parametrization of the profile (via Bezier functions [9]) thus alleviating the infeasibility concerns from item i) and, using as a proxy the thrust input of the translational dynamics of the quadcopter, to provide a reasonable approximation of the energy consumption required by item ii).

The key advantage of Bezier parametrization is that it allows to provide an explicit formulation of the cost in terms of the weights which define the trajectory (the control points). This allows to analyze the cost value in different configurations. In particular, we compare horizontal and vertical sweep strategies taking into account the velocity and acceleration profiles along the path. Specifically, we provide a velocity profile which allows an arbitrary hovering time at each way-point.

Having explicit relations between average velocity, path length and way-points position allows to tweak the overall mission to reduce duration (uses the maximum allowed energy), energy consumption (at the price of a longer run time) or a mix between the two of them. Section II gives the prerequisites further used in the energy consumption analysis carried in Section III and for the profile generation procedure discussed in Section IV. These are further combined for the proof-of-concept implementation from Section V. Conclusions are drawn in Section VI.

II. Prerequisites

This section gives the relevant (for this paper) elements of Bezier function parametrizations and quadcopter dynamics.

[1]"Politehnica" University of Bucharest (UNSTPB), Romania, 313 Splaiul Independenţei, 060042, Sector 6, Bucharest, Romania (e-mails: florin.stoican,dan.popescu, loretta.ichim@upb.ro).
[2] Univ. Grenoble Alpes, Grenoble INP†, LCIS, F-26000, Valence, France, vicent.marguet, ionela.prodan@lcis.grenoble-inp.fr. † Institute of Engineering and Management, Univ. Grenoble Alpes.

* This work was partially supported through the HALY.ID project which is part of ERA-NET Cofund ICT-AGRI-FOOD, with funding provided by national sources [Funding agency UEFISCDI, project number 202/2020, within PNCDI III] and co-funding by the European Union's Horizon 2020 research and innovation program, Grant Agreement number 862665 ERA-NET ICT-AGRI-FOOD (HALY-ID 862671).

2024 32nd Mediterranean Conference on Control and Automation (MED) | 979-8-3503-9544-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/MED61351.2024.10566251

A. Bezier functions

For further use we consider a trajectory $\xi(t) \in \mathbb{R}^d$, expressed as a combination of basis functions $B_{i,n}(t) \in \mathbb{R}$, weighted by control points $P_i \in \mathbb{R}^d$:

$$\xi(t) = \sum_{i=0}^{n} P_i B_{i,n}(t), \quad \forall t \in [0,1]. \tag{1}$$

Although theoretically any family of basis functions can be chosen, we prefer to work with Bezier functions[1] as they have many interesting properties [10] and are simple to handle:

$$B_{i,n}(t) = \binom{n}{i}(1-t)^{n-i}t^i, \quad \forall t \in [0,1]. \tag{2}$$

Among the many properties of interest we note that the Bezier functions are positive ($B_{i,n}(t) \geq 0$) and partition the unity ($\sum_{i=0}^{n} B_{i,n}(t) = 1$) over their domain, $[0,1]$. This allows to state the first property of interest [10].

Property 1. *The continuous curve $\xi(t)$ lies in the convex hull defined by control points $\{P_0, \ldots, P_n\}$.*

Further, any p-times derived Bezier function of order n may be expressed as a combination of order $n-p$ functions. This leads to the second property.

Property 2. *First and second order derivations of* (1) *are themselves combinations of Bezier functions (of order $n-1$ and $n-2$ this time) and control points:*

$$\dot{\xi}(t) = n\sum_{i=0}^{n-1}(P_{i+1} - P_i)B_{i,n-1}(t), \tag{3a}$$

$$\ddot{\xi}(t) = n(n-1)\sum_{i=0}^{n-2}(P_{i+2} - 2P_{i+1} + P_i)B_{i,n-2}(t). \tag{3b}$$

Incidentally, Property 1, applied to (3) shows that also the velocity and acceleration are contained within convex hulls (this time given by sequences $P'_i := n(P_{i+1} - P_i)$ and $P''_i := n(n-1)(P_{i+2} - 2P_{i+1} + P_i)$, respectively).

Recall that, by construction, Bezier functions are defined over the domain $[0,1]$. This may be relaxed as follows.

Property 3. *Considering that a profile $\xi(t)$ is defined over the interval $t \in [\alpha, \beta]$ we may reformulate* (1) *by an affine transformation of the Bezier functions' argument:*

$$\xi(t) = \sum_{i=0}^{n} P_i B_{i,n}\left(\frac{t-\alpha}{\beta-\alpha}\right), \quad \forall t \in [\alpha, \beta]. \tag{4}$$

Remark 1. *Note that using* (4) *instead of* (1) *will change accordingly the relations from* (3)*, e.g.,* (3b) *becomes*

$$\ddot{\xi}(t) = \frac{n(n-1)}{(\beta-\alpha)^2}\sum_{i=0}^{n-2}(P_{i+2} - 2P_{i+1} + P_i)B_{i,n-2}\left(\frac{t-\alpha}{\beta-\alpha}\right).$$

♦

[1]The binomial term is given by the formula: $\binom{n}{i} = \frac{n!}{i!(n-i)!}$.

B. Quadcopter dynamics

In what follows we recapitulate some of the standard results in quadcopter modelling and control [11]. It is well-known that a multi-copter's dynamics are best put in an inner (rotational dynamics) and outer (translational dynamics) loop. The former is, without exception (at least for commercial drones) implemented on the drone itself and cannot be accessed by the end-user. What may be tweaked (again, depending on the vendor and the technical prowess of the user) are the control parameters of the outer loop (i.e., the translational dynamics).

To this end, we give[2] the standard (in the inertial coordinate frame) translational nonlinear dynamics of a generic quadcopter system:

$$\ddot{x} = (c\phi\, s\theta\, c\psi + s\phi\, s\psi)\cdot T, \tag{5a}$$

$$\ddot{y} = (c\phi\, s\theta\, s\psi - s\phi\, c\psi)\cdot T, \tag{5b}$$

$$\ddot{z} = g + c\phi\, c\theta \cdot T, \tag{5c}$$

where the system's state (position plus velocity) is given by $\begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} \end{bmatrix}$ and $\begin{bmatrix} T & \phi & \theta \end{bmatrix}$ denotes the input, where T is the normalized thrust, ϕ is the roll angle, and θ is the pitch angle. The yaw angle (ψ) of the quadcopter is considered a known (measured) parameter. Notations 'c' and 's' are shorthands for the trigonometric cos and sin operators.

It is well-known [11], and we do not further detail the intermediary calculations here, that all input components (thrust, roll and pitch) may be expressed in terms of the desired accelerations (further denoted with the 'overline' symbol):

$$T = \sqrt{\ddot{\overline{x}}^2 + \ddot{\overline{y}}^2 + (\ddot{\overline{z}} - g)^2}, \tag{6a}$$

$$\overline{\phi} = \arcsin\left(\frac{\ddot{\overline{x}}s\psi - \ddot{\overline{y}}c\psi}{\sqrt{\ddot{\overline{x}}^2 + \ddot{\overline{y}}^2 + (\ddot{\overline{z}} - g)^2}}\right), \tag{6b}$$

$$\overline{\theta} = \arctan\left(\frac{\ddot{\overline{z}} + g}{\ddot{\overline{x}}c\psi + \ddot{\overline{y}}s\psi}\right). \tag{6c}$$

The thrust is given directly to the drone but the roll and pitch are passed as references to the inner loop which controls the rotational dynamics. Assuming that the inner loop ensures $\theta \mapsto \overline{\theta}, \phi \mapsto \overline{\phi}$ and directly applying the thrust T to the drone means that, under nominal conditions, we have $x \mapsto \overline{x}, y \mapsto \overline{y}, z \mapsto \overline{z}$.

A couple of remarks are in order.

Remark 2. *It must be noted that any realistic control scheme should consider a feedback component to counteract tracking errors due to internal model uncertainties and external disturbances (such as wind gusts).* ♦

Remark 3. *Relations* (6a)–(6c) *implement a so-called model inversion procedure where the desired profile (terms $\overline{x}, \overline{y}, \overline{z}$) are used to determine the desired behavior of the input. While powerful, this approach may also be dangerous due to the*

[2]All signals appearing here are time-dependent but, to avoid the clutter, whenever clear from context we discard the argument from the equations (e.g., x instead of $x(t)$).

nonlinearities appearing in the equations. We use it here since it is quite popular in the literature and, simultaneously, allows us to give explicit bounds for the cost. ♦

III. Explicit cost descriptions

Ignoring energy consumption for ancillary mechanisms (such as payload pan/tilt movements, the current drawn for the camera and for telemetry) the main energy expenditure is due to the quadcopter's motor torques (proportional with ω_i^2 where ω_i is the i-th motor angular velocity). Expressing these in terms of the desired profile (given by $\overline{x}, \overline{y}, \overline{z}$) is quite impractical due to the convoluted mathematical relations that link them [12]. Therefore, in the rest of the paper, as a proxy for energy consumption, we consider the total thrust effort along the simulation:

$$J_{\text{thrust}} = \int_{\alpha}^{\beta} [T(t)]^2 \, \mathrm{d}t. \tag{7}$$

Remark 4. *Using the thrust cost* (7) *as a proxy for energy consumption is adequate when the angle torques (those that govern the roll and pitch dynamics) are small. This is a reasonable assumption in the precision agriculture setting where trajectories are relatively 'mild', without aggressive changes in pitch and roll values. Thus, the only angle torques come from the feedback component of the control law and those are expected to be small under nominal functioning.*♦

Combining the tools given in the previous section we provide an explicit description of (7). The first step is to express the thrust in terms of control points and Bezier functions.

Lemma 1. *Taking* $\xi = \begin{bmatrix} x & y & z \end{bmatrix}^\top$*, parametrized as in* (1)*, we may now express* (6a) *over the interval* $t \in [\alpha, \beta]$ *with the help of* (3b) *as*

$$T(t) = \sqrt{\sum_{i,j=0}^{n-2} (P_i'' - e_3 g)^\top (P_j'' - e_3 g) \, B_{i,n-2}(\tau) B_{j,n-2}(\tau)}, \tag{8}$$

with notation $\tau := (t-\alpha)/(\beta-\alpha)$.

Proof: Using the Bezier properties allows to write

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} - g \end{bmatrix} = \ddot{\xi} - e_3 g = \sqrt{\sum_{i=0}^{n-2} (P_i'' - e_3 g) \, B_{i,n-2}(t)},$$

with the shorthands $P_i'' := n(n-1)(P_{i+2} - 2P_{i+1} + P_i)$ and $e_3^\top := \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\top$. With this notation, T, defined as in (6a), is rewritten as $T = \|\ddot{\xi} - e_3 g\|_2$, which, together with Property 3, directly leads to (8), thus concluding the proof. □

With the afore-defined lemma we may now provide an explicit expression for the cost (7).

Proposition 1. *With the notation of Lemma 1 and the observation from Remark 1, the cost* (7) *may be expressed in term of the control points* $\{P_i''\}$ *as*

$$J_{\textit{thrust}} = (\beta - \alpha) \cdot \sum_{i,j=0}^{n-2} \left[I_{ij,n-2} \left(\frac{P_i''}{(\beta-\alpha)^2} - e_3 g \right)^\top \left(\frac{P_j''}{(\beta-\alpha)^2} - e_3 g \right) \right], \tag{9}$$

with the shorthand $I_{ij,n-2} := \int_0^1 B_{i,n-2}(\tau) B_{j,n-2}(\tau) \, d\tau$.

Proof: Using (8) from Lemma 1 in (7) and switching the sum/integral operator ordering, we obtain

$$J_{\text{thrust}} = \sum_{i,j=0}^{n-2} \left[\left(\frac{P_i''}{(\beta-\alpha)^2} - e_3 g \right)^\top \left(\frac{P_j''}{(\beta-\alpha)^2} - e_3 g \right) \int_{\alpha}^{\beta} B_{i,n-2}(\tau) B_{j,n-2}(\tau) \, \mathrm{d}t \right].$$

Recalling that $\tau = (t-\alpha)/(\beta-\alpha)$ leads to $\mathrm{d}\tau = \mathrm{d}t/(\beta-\alpha)$ allows to rewrite the integral term as

$$(\beta - \alpha) \int_0^1 B_{i,n-2}(\tau) B_{j,n-2}(\tau) \, \mathrm{d}\tau,$$

thus concluding the proof.

The overall cost should not only consider the thrust's energy but also to account for the total time,

$$J_{\text{time}} = \beta - \alpha, \tag{10}$$

and total length[3] of the trajectory,

$$J_{\text{path}} = \int_{\alpha}^{\beta} \|\xi'(\tau)\|^2 \mathrm{d}t. \tag{11}$$

Applying the same reasoning as in Proposition 1, the cost (11) is rewritten as

$$J_{\text{path}} = \frac{1}{\beta - \alpha} \cdot \sum_{i,j=0}^{n-1} \left[I_{ij,n-1} \cdot (P_i')^\top P_j' \right], \tag{12}$$

with the shorthand $I_{ij,n-1} := \int_0^1 B_{i,n-1}(\tau) B_{j,n-1}(\tau) \, \mathrm{d}\tau$.

Put together, the overall cost may then be written as

$$J_{\text{total}} = \gamma_1 J_{\text{thrust}} + \gamma_2 J_{\text{time}} + \gamma_3 J_{\text{path}}, \tag{13}$$

with $\gamma_{1,2,3}$, the weights which decide the relative importance of a cost component.

Remark 5. *Note that all components of the total cost depend on the length of the simulation/experiment time (the term '*$\beta - \alpha$*' appearing in* (9)*,* (10) *and* (12)*). In particular,* (9) *shows that the more time is spent flying, the more the relative importance of canceling the drone's weight (due to the gravitational acceleration term –* g*) increases.* ♦

[3] In fact, (11) gives the energy of the trajectory but it is used nonetheless due to its quadratic form and because it provides a good approximation of the total length.

IV. Profile generation

The next step is to provide a procedure for choosing the control points such as to ensure reasonable position, velocity and acceleration profiles. While additional requirements may always be considered, we restrict here to:

i) velocity and acceleration bounds: $\|\dot{\xi}(t)\|_2 \leq \bar{v}$ and $\|\ddot{\xi}(t) - e_3 g\|_2 \leq \bar{a}$ for $t \in [\alpha, \beta]$, respectively;
ii) for a list of a priori given way-points, W_ℓ, the drone has to hover at that position for a pre-specified time τ_ℓ.

The first condition, item i), is readily implemented by enforcing the convexity condition of Property 1 for the relations from Property 2, with the modifications shown in Remark 1:

$$\frac{1}{(\beta-\alpha)^2}\sum_{i,j=0}^{n-1}(P_i')^\top P_j' \leq \bar{v}^2, \quad (14a)$$

$$\sum_{i,j=0}^{n-2}\left(\frac{P_i''}{(\beta-\alpha)^2} - e_3 g\right)^\top \left(\frac{P_j''}{(\beta-\alpha)^2} - e_3 g\right) \leq \bar{a}^2. \quad (14b)$$

Remark 6. *Note that conditions* (14) *are only sufficient. There may be curves* (1) *which respect the requirements of item i) without checking* (14)*. The advantage of* (14) *is that the constraints are expressed only in the control points sequence* $\{P_i\}$*, thus greatly simplifying the subsequent calculations.* ♦

Of greater interest is item ii) which ensures that the actual mission objective (taking pictures at the desired way-point) is fulfilled. To do so, let us consider the behavior at and around the i-th way-point W_i:

i) the path goes from the mid-point between the current way-point an its predecessor (point $(W_{i-1}+W_i)/2$) and until the mid-point between the current way-point and its successor (point $(W_i + W_{i+1})/2$);
ii) the associated time interval is $[\alpha_i, \beta_i]$ with $\alpha_i := \beta_{i-1}$;
iii) at way-point W_i, the drone hovers (velocity and acceleration are zero) for a duration τ_i;
iv) velocity and acceleration are assumed to be maximal ($\bar{v}$ and $\bar{a}$) at the start and end points of the trajectory.

While there are multiple ways of tackling these specifications, here we consider a closed-form construction to avoid expensive (in computation time and hardware resources) numerical optimizations. Specifically, we divide the interval $[\alpha_i, \beta_i]$ into

$$\left[\alpha_i, \frac{\beta_i+\alpha_i-\tau_i}{2}\right) \cup \underbrace{\left[\frac{\beta_i+\alpha_i-\tau_i}{2}, \frac{\beta_i+\alpha_i+\tau_i}{2}\right)}_{\tau_i} \cup \left[\frac{\beta_i+\alpha_i+\tau_i}{2}, \beta_i\right), \quad (15)$$

where the middle sub-interval corresponds to the hovering stage and the first and third ones are described by Bezier curve profiles. As a first step we give the following lemma.

Lemma 2. *Consider a profile defined as in* (4) *over the interval* $[\alpha, \beta]$ *with* $n \geq 6$*. Then, the position, velocity and acceleration specifications* $\xi(\alpha) = \bar{p}_1, \dot{\xi}(\alpha) = \bar{v}_1, \ddot{\xi}(\alpha) = \bar{a}_1$ *and* $\xi(\beta) = \bar{p}_2, \dot{\xi}(\beta) = \bar{v}_2, \ddot{\xi}(\beta) = \bar{a}_2$*, respectively are respected iff the following relations hold:*

$$\begin{aligned}
P_0 &= \bar{p}_1, & P_n &= \bar{p}_2, \\
P_1 &= \bar{p}_1 + \frac{\Delta}{n}\bar{v}_1, & P_{n-1} &= \bar{p}_2 - \frac{\Delta}{n}\bar{v}_2, \\
P_2 &= \bar{p}_1 + \frac{2\Delta}{n}\bar{v}_1 + \frac{\Delta^2}{n(n-1)}\bar{a}_1, & P_{n-2} &= \bar{p}_2 - \frac{2\Delta}{n}\bar{v}_2 + \frac{\Delta^2}{n(n-1)}\bar{a}_2.
\end{aligned} \quad (16)$$

with shorthand $\Delta := \beta - \alpha$*.*

Proof: By the definition (1), at the end-points of the domain ($t = \alpha$ and $t = \beta$, respectively) only the first and, respectively, the last Bezier function are non-zero. Thus, we may write the position, velocity and acceleration conditions as

$$\begin{aligned}
\bar{p}_1 &= P_0, & \bar{p}_2 &= P_n, \\
\bar{v}_1 &= \frac{nP_0'}{\beta-\alpha}, & \bar{v}_2 &= \frac{nP_{n-1}'}{\beta-\alpha}, \\
\bar{a}_1 &= \frac{n(n-1)P_0''}{(\beta-\alpha)^2}, & \bar{a}_2 &= \frac{n(n-1)P_{n-2}''}{(\beta-\alpha)^2},
\end{aligned}$$

recalling the shorthand notations $P_i' := P_{i+1} - P_i$ and $P_i'' := P_{i+2} - 2P_{i+1} + P_i$. Simple manipulations of these relations lead to (16), thus concluding the proof. □

Proposition 2. *For the first and third sub-intervals from* (15)*, the desired specifications are respected by profiles of form* (4)*, defined by control point sequences* $\{P_0^{i,-}, \dots P_n^{i,-}\}$ *where*

$$\begin{aligned}
P_0^{i,-} &= \frac{W_{i-1}+W_i}{2}, & P_n^{i,-} &= W_i, \\
P_1^{i,-} &= \frac{W_{i-1}+W_i}{2} + \frac{\beta_i-\alpha_i-\tau_i}{2n}\bar{v}, & P_{n-1}^{i,-} &= W_i, \\
P_2^{i,-} &= \frac{W_{i-1}+W_i}{2} + \frac{\beta_i-\alpha_i-\tau_i}{n}\bar{v} + \frac{(\beta_i-\alpha_i-\tau_i)^2}{4n(n-1)}\bar{a}, & P_{n-2}^{i,-} &= W_i.
\end{aligned} \quad (17)$$

and, respectively, $\{P_0^{i,+}, \dots P_n^{i,+}\}$ *where*

$$\begin{aligned}
P_0^{i,+} &= W_i, \quad P_n^{i,+} = \frac{W_i+W_{i+1}}{2}, \\
P_1^{i,+} &= W_i, \quad P_{n-1}^{i,+} = \frac{W_i+W_{i+1}}{2} - \frac{\beta_i-\alpha_i-\tau_i}{2n}\bar{v}, \\
P_2^{i,+} &= W_i, \quad P_{n-2}^{i,+} = \frac{W_i+W_{i+1}}{2} - \frac{\beta_i-\alpha_i-\tau_i}{n}\bar{v} + \frac{(\beta_i-\alpha_i-\tau_i)^2}{4n(n-1)}\bar{a}.
\end{aligned} \quad (18)$$

Proof: Lemma 2 is applied twice. First we make the substitutions $\bar{p}_1 \leftarrow\!\rightarrow (W_{i-1}+W_i)/2, \bar{v}_1 \leftarrow\!\rightarrow \bar{v}, \bar{a}_1 \leftarrow\!\rightarrow \bar{a}, \bar{p}_2 \leftarrow\!\rightarrow W_i, \bar{v}_2 \leftarrow\!\rightarrow 0, \bar{a}_2 \leftarrow\!\rightarrow 0$ and $\alpha \leftarrow\!\rightarrow \alpha_i, \beta \leftarrow\!\rightarrow (\beta_i+\alpha_i-\tau_i)/2$. These lead to equations (17). The procedure is repeated for (18) for the substitutions $\bar{p}_1 \leftarrow\!\rightarrow W_i, \bar{v}_1 \leftarrow\!\rightarrow 0, \bar{a}_1 \leftarrow\!\rightarrow 0, \bar{p}_2 \leftarrow\!\rightarrow$

$(W_i + W_{i+1})/2, \bar{v}_2 \leftarrow\!\shortmid\, \bar{v}, \bar{a}_2 \leftarrow\!\shortmid\, \bar{a}$ and $\alpha \leftarrow\!\shortmid\, (\beta_i + \alpha_i + \tau_i)/2, \beta \leftarrow\!\shortmid\, \beta_i$, thus concluding the proof.

Remark 7. *Note that Proposition 2 restricts only the first and last three control points of the profiles. The remaining ones ($P_3^{i,\pm}, \dots P_{n-3}^{i,\pm}$) are free and may be the subject of further optimization. Hereinafter, to simplify, we take $n = 6$ and avoid the issue altogether.* ♦

V. Application to orchard navigation

We have now all prerequisites to solve the larger navigation problem. To illustrate and validate the procedure we consider a real apple orchard strip (35m x 68m) where the following steps have been carried out:

i) an initial photogrammetry flight has taken overlapping pictures (around 80, at height of 25m, with front and lateral overlap of over 60%) to construct the orthomosaic depicted in Fig. 1a;
ii) from the raw data obtained at the previous step, elevation models are obtained, Fig. 1b depicts the difference between surface and terrain elevations (the heights from the DSM – *digital surface map* are compared with those stored in the DTM – *digital terrain map* to isolate the tree shapes);
iii) which is then manipulated further in Fig. 1c via standard image processing tools (repeated operations of inflation and erosion, followed subsequently by a segmentation step) to obtain the blobs defining the orchard rows;
iv) lastly, in Fig. 1d, oriented bounding boxes are fitted over each row and way-points which respect pre-specified horizontal and vertical spacings are selected.

These steps have been carried in the software suite ODM – *Open Drone Map* [13], with subsequent done with the Image Processing toolbox of Matlab.

Remark 8. *Note that in this particular case we choose the (x, y) positions of the way-point such that they are equally distanced along the oriented boxes' length. This is a reasonable assumption for modern orchards where trees are positioned at pre-defined distances and where anyhow their coronas overlap into a "wall of folliage". Nonetheless, additional steps may be carried out, e.g., by identifying regions of higher density within a given row (the tree centers).* ♦

In the specific case of the orchard, the way-points define vertical slices (planes parallel to each orchard row, offset from the row's center to a safe distance). Without any loss of generality let us consider a single slice which is defined by way-points $W_{h,v}$ where indices h, v denote respectively the offset along the row length (horizontal movement) and the row height (vertical movement), as obtained in step iv) of the algorithm. We may assume two types of movement:

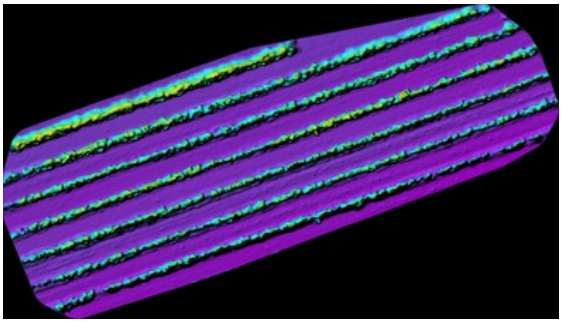
i) width-first, we move horizontally, at fixed height

$$W_i \leftarrow\!\shortmid\, W_{h,v}, \quad W_{i\pm1} \leftarrow\!\shortmid\, W_{h\pm1,v}, \tag{19}$$

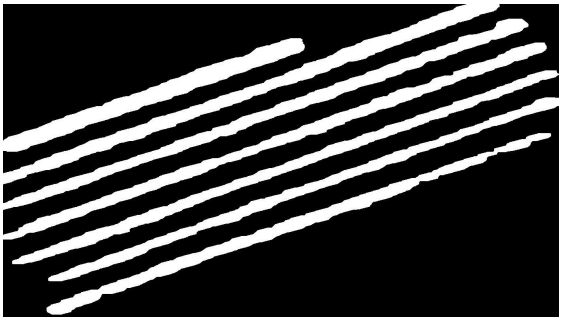
ii) height first, we move vertically, at fixed coordinates

$$W_i \leftarrow\!\shortmid\, W_{h,v}, \quad W_{i\pm1} \leftarrow\!\shortmid\, W_{h,v\pm1}. \tag{20}$$

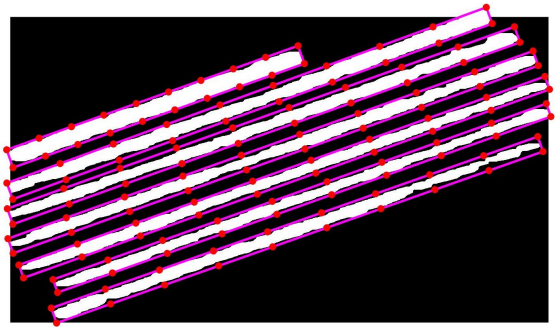
(a) mosaic map from photogrammetry flight



(b) depth map from cloud point



(c) segmentation



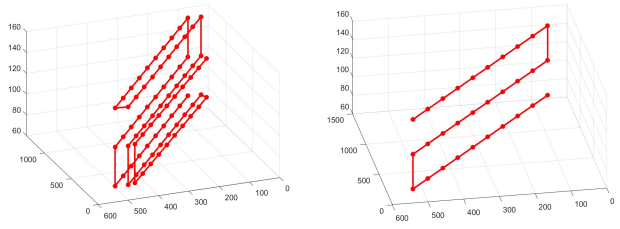
(d) way-point selection

Fig. 1: Illustration of way-point selection in orchard coverage

Note that the sign may change if the movement is one of advance/return or of ascent/descent (e.g., instead of mapping $W_{i\pm1} \leftarrow\!\shortmid\, W_{h,v\pm1}$ we may have $W_{i\pm1} \leftarrow\!\shortmid\, W_{h,v\mp1}$).

Fig. 2 illustrates the horizontal-first sweep mode. Fig. 2a shows three vertical planes (the left and right planes for the first orchard row and the left for the second row). The first plane (with markers denoting the way-point locations) is

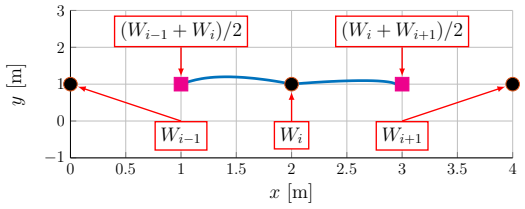
detailed in Fig. 2b. Subsequently, an analysis which estimates the total cost (13) may be carried. It should be noted that the actual numerical values (and hence, the decision of which sweep mode is more efficient) are significantly influenced by the various parameter employed (maximum velocity and acceleration, way-point horizontal and vertical distance, etc.).



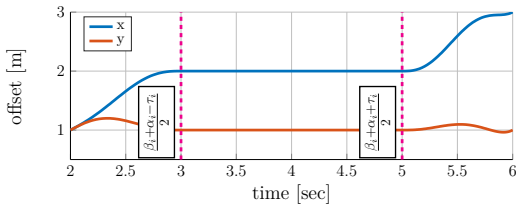
(a) horizontal sweep (b) vertical plane detail

Fig. 2: Illustration for the horizontal sweep mode

To each way-point W_i we have a profile composed of three segments: the Bezier curve characterized by (17), the hovering of duration τ_i and the Bezier curve characterized by (18). To each of these corresponds a cost as defined in (13). Note that for the mid-segment (the one with the hovering) we may directly give $J_{\text{thrust}} = g^2\tau_i$, $J_{\text{time}} = \tau_i$ and $J_{\text{path}} = 0$.



(a) profile in $x-y$ coordinates



(b) profile as function of time

Fig. 3: Illustration of profile generation for a given way-point

As illustration, we consider a proof-of-concept implementation where the way-points are in a vertical grid, at heights $\{0.5, 0.75, 1\}$m and are separated horizontally every 2.5m for an orchard row of length $50m$. Thus, we have $3 \times 21 = 63$ way-points that we may cover either as in (19) or as in (20). Further, we assume $\tau_i = 2$sec, $\beta_i - \alpha_i = 4$sec, $\bar{v} = 1$m/sec and $\bar{a} = 0.1$m/sec^2. Figure 3 depicts the profiles (position in $x-y$ coordinates and as functions of time) obtained for a particular way-point in mode (19). Computing the J_{total} as in (9) gives us $388.19 + 39.2 + 380.73 = 808.12$.

VI. Conclusions

We proposed a mechanism to estimate the total energy consumption for a quadcopter during flight using as proxy the total thrust effort. We provided closed-form descriptions of all elements of interest (cost components, profile shape) in terms of the control points which weight the Bezier parametrization of the profile. This allowed us to analyze the behavior of the navigation scheme in two different sweep patterns. Further work will concentrate on the influence of the tuning parameters, the selection of the mid-points, the validation of additional constraints and on more accurate energy consumption estimations.

References

[1] P. Pradeep, S. G. Park, and P. Wei, "Trajectory optimization of multirotor agricultural UAVs," in *2018 IEEE Aerospace Conference*. IEEE, 2018, pp. 1–7.

[2] A. Mokrane, A. C. Braham, and B. Cherki, "UAV coverage path planning for supporting autonomous fruit counting systems," in *2019 International Conference on Applied Automation and Industrial Diagnostics (ICAAID)*, vol. 1. IEEE, 2019, pp. 1–5.

[3] I. A. Sulistijono, M. R. Ramadhani, and A. Risnumawan, "Aerial drone mapping and trajectories generator for agricultural ground robots," in *2020 International Symposium on Community-centric Systems (CcS)*. IEEE, 2020, pp. 1–6.

[4] J. G. A. Barbedo, "A review on the use of unmanned aerial vehicles and imaging sensors for monitoring and assessing plant stresses," *Drones*, vol. 3, no. 2, p. 40, 2019.

[5] A. I. Castro, de, Y. Shi, J. M. Maja, and J. M. Peña, "UAVs for vegetation monitoring: Overview and recent scientific contributions," *Remote Sensing*, vol. 13, no. 11, p. 2139, 2021.

[6] A. Furchì, M. Lippi, R. F. Carpio, and A. Gasparri, "Route optimization in precision agriculture settings: A multi-steiner tsp formulation," *IEEE Transactions on Automation Science and Engineering*, 2022.

[7] Y.-H. Tu, S. Phinn, K. Johansen, A. Robson, and D. Wu, "Optimising drone flight planning for measuring horticultural tree crop structure," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 160, pp. 83–96, 2020.

[8] C. Zhang, J. Valente, L. Kooistra, L. Guo, and W. Wang, "Orchard management with small unmanned aerial vehicles: A survey of sensing and analysis approaches," *Precision Agriculture*, vol. 22, no. 6, pp. 2007–2052, 2021.

[9] L. Greco, H. Mounier, and M. Bekcheva, "An approximate characterisation of the set of feasible trajectories for constrained flat systems," *Automatica*, vol. 144, p. 110484, 2022.

[10] L. Piegl and W. Tiller, *The NURBS book*. Springer Science & Business Media, 1996.

[11] N. T. Nguyen, I. Prodan, and L. Lefèvre, "Flat trajectory design and tracking with saturation guarantees: a nano-drone application," *International Journal of Control*, vol. 93, no. 6, pp. 1266–1279, 2020.

[12] M. Faessler, A. Franchi, and D. Scaramuzza, "Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 620–626, 2017.

[13] "Opendronemap – a command line toolkit to generate maps, point clouds, 3d models and dems from drone, balloon or kite images," 2020, https://github.com/OpenDroneMap/ODM.