Energy-Aware Dynamic Mission Planning Algorithm for UAVs

Adam Seewald[1], Hector Garcia de Marina[2], and Ulrik Pagh Schultz[1]

***Abstract*— abstract abstract abstract abstract abstract abstract abstract abstract abstract abstract abstract abstract abstract abstract**

I. Introduction

Planning a dynamic mission for unmanned aerial vehicles (UAVs) operating outdoors is a challenging task. Scenarios such as precision agriculture, search and rescue, and surveillance require advanced levels of autonomy but have limited energy budgets. A typical example of a mission with these limitations is a UAV flying a trajectory and performing some on-board computations. For instance, the UAV might detect some patterns and inform other, ground-based, actors. Planning a dynamic mission for such UAV involves finding optimal tradeoffs between the trajectory and computations. However, UAVs flying outdoors are often semi-autonomous, in the sense that the mission is static and usually defined using mission planning software [1]. Such a state of practice has prompted us to propose an *energy-aware dynamic mission planning algorithm* for UAVs. The algorithm combines and generalizes some of the past body of knowledge on mobile robot planning problems, and addresses the increasing *computational demands* and their relation to energy consumption, path, and autonomy.

Planning algorithms for mobile robots are related to topics such as trajectory generation and path planning. Generally, these algorithms select an energy-optimized trajectory [2], e.g., by maximizing the operational time [3], but in practice apply to a small number of robots [4], and focus on optimizing motion control for these robots [5], despite compelling evidence for the energy consumption also being influenced by computations and not only bare motion [6]. For UAVs specifically, rotorcrafts have gained interest in terms of algorithms for energy-optimized trajectory generation [7], [8]. Mission planning algorithms, which include a broader concept of a mission being a set of tasks along with a motion plan, all focus on the trajectory [6], [9] and apply to a small number of robots [10], [11]. Given the availability of powerful GPU-equipped mobile hardware, the use of computations is expected to increase in the near future.

The user specifies an initial mission plan–the trajectory and the tasks–along with an energy budget and the following mission specification parameters. The Quality of Service (QoS) computations parameters are relative to the tasks, and trajectory-explicit equations (TEEs) adjustments parameters to the path. The proposed algorithm replans the initial mission plan dynamically as the energy budget changes due to atmospheric interferences. In the remainder of the paper, we adopt the following notation. We refer to the values of mission-specific QoS and TEEs parameters as computations and adjustments, we refer to the constraints sets that delimit such computations and adjustments as QoS and TEEs sets, and we refer to the current trajectory as TEE. Our goal is to complete the mission with the highest possible computations and adjustments as the UAV flies and its batteries drain. The algorithm both optimizes computations in the form of robot operating system (ROS) nodes, and it optimizes adjustments–a way to alter the trajectory–guiding the UAV using a vector field [12] that converges smoothly to such trajectory. It relies on the assumptions of the mission being *periodic* and *uncertain*. Periodicity is often observed, e.g., when the UAV is flying in repetitive patterns for monitoring or search and rescue scenarios. Uncertainty accounts for the atmospheric interference, e.g., a fixed-wing UAV drifting due to windy weather. The algorithm addresses the periodicity by modeling the energy with Fourier analysis: the mission is periodic, so we expect the energy to also approximately evolve periodically. The uncertainty is addressed with a state estimator. The algorithm outputs the control input (adjustments and computations) using output model predictive control (MPC).

In the spirit of reducing costs and resources, we showcase the algorithm using the problem of dynamic mission planning for a precision agriculture fixed-wing UAV. Such a scenario is often put into practice [13] with ground mobile robots used for harvesting [14]–[19], and UAVs for preventing damage and ensuring better crop quality [1], [20]. The mission is structured as follows. Trajectory-wise, the UAV flies in circles and lines covering a polygon. Computationally-wise, it detects obstacles using a convolutional neural network (CNN), and informs grounded mobile robots employed for

This work is supported and partly funded by the European Union's Horizon2020 research and innovation program under grant agreement No. 779882 (TeamPlay).

[1]Adam Seewald, Ulrik Pagh Schultz are with the SDU UAS Center, Mærsk Mc-Kinney Møller Institute, University of Southern Denmark, Odense, Denmark. Email: ads@mmmi.sdu.dk.

[2]Hector Garcia de Marina is with the Faculty of Physics, Department of Computer Architecture and Automatic Control, Universidad Computense de Madrid, Spain.

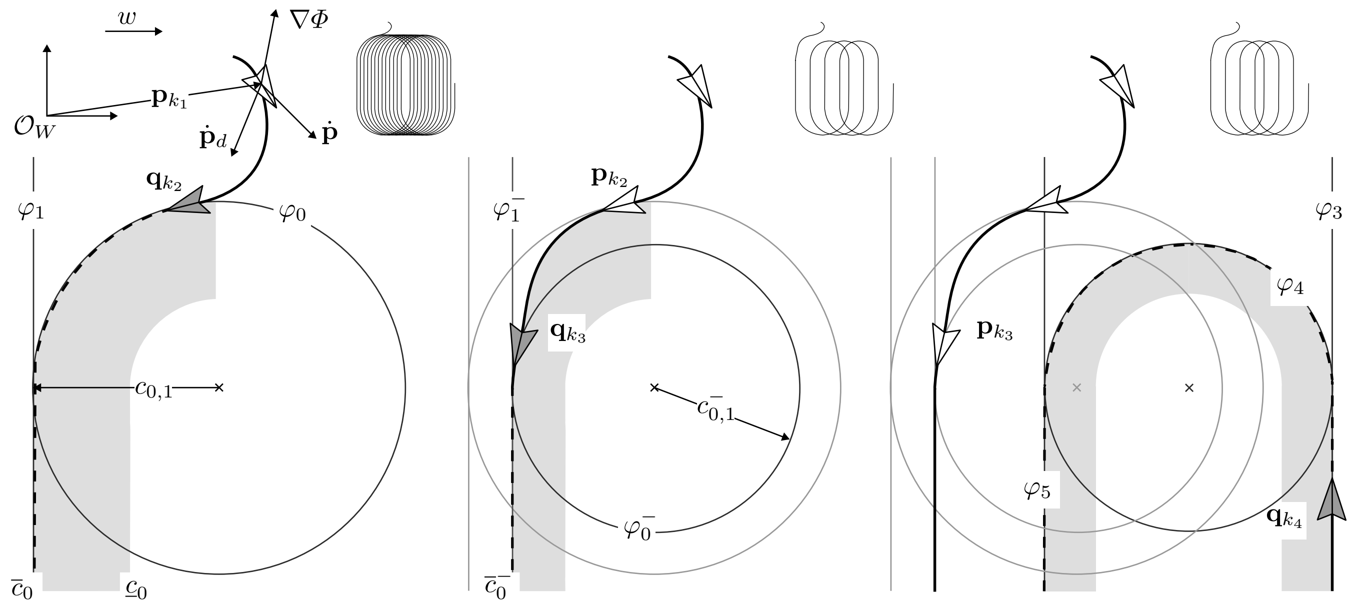


Fig. 1. The mission consists of circles and lines covering a polygon. The UAV heads to φ_0 initially, using the desired velocity vector $\dot{\mathbf{p}}_d$. It later heads to φ_0^- reducing the radius $c_{0,1}$ to satisfy the battery constraints. The UAV then converges to φ_1 in stage $\mathcal{M}_1$, φ_2 in stage $\mathcal{M}_2$, and so on (the circle φ_2 is not visible in the figure; it connects φ_1 and φ_3).

harvesting. The algorithm plans the mission controlling the processing rate and the radius of the circles (affecting the distance between the lines). Figure 1 shows an initial slice of such a mission. The UAV first heads to a circular TEE with a given radius, which is later reduced as, e.g., windy weather requires the adjustment of the control input to avoid battery depletion. Computations significantly impact the battery, with a potential extension of up to 13 minutes over an hour by merely switching to the lowest computations (see Section IV).

The remainder of the paper is organized as follows. The overview of dynamic mission planning is set in Section II, along with a suitable model for the position and energy. The algorithm that uses the model and solves the UAV dynamic mission planning problem is proposed in Section III. Section IV presents the result and showcase the performances. The paper finishes with some conclusions in Section V.

II. Mission Planning Overview

We assume that the initial mission plan consists of different stages. At each stage the robot must follow a path and do some tasks.

Let the path at stage i be characterized by a generic continuous twice differentiable TEE $\varphi_i : \mathbb{R}^2 \times \mathbb{R}^\rho \to \mathbb{R}$ and the tasks by functions $\psi_1, \ldots, \psi_\sigma : \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$. Moreover, let $[a]$ be the set $\{0, 1, \ldots, a\}$, $[a]^+$ the set $[a]/\{0\}$, $\langle a_1, a_2, ..., a_n \rangle$ an ordered list of n elements, and $\underline{c}, \bar{c}$ the lower and upper bound of the parameter c retrieved from a lookup table.

Definition II.1 (Stage and mission)**.** A *stage* $\mathcal{M}_i$ at time instant k of a mission $\mathcal{M}$ is defined as the ordered list

$$\begin{aligned}\mathcal{M}_i := \{&\langle \varphi_i(\mathbf{p}_k, c_{i,1}, \ldots, c_{i,\rho}), \psi_1(s_{i,1}), \ldots, \psi_\sigma(s_{i,\sigma})\rangle \\ &\mid \exists\ \mathbf{p}_k, \varphi_i(\mathbf{p}_k, c_{i,1}, \ldots, c_{i,\rho}) \in \mathbb{C}_i, \\ &\psi_j(s_{i,j}) \in \mathbb{S}_{i,j}\ \forall j \in [\sigma]^+\},\end{aligned} \tag{1}$$

where $\mathbb{C}_i := [\underline{c}_i, \bar{c}_i] \subseteq \mathbb{R}$ is the TEEs set, and $\mathbb{S}_{i,j} := [\underline{s}_{i,j}, \bar{s}_{i,j}] \subseteq \mathbb{Z}_{\geq 0}$ the j-th task QoS set. $\mathbf{p}_k := (x, y)$ is a point of a UAV flying at an assigned altitude $h \in \mathbb{R}_{>0}$ w.r.t. some inertial navigation frame $\mathcal{O}_W$. The parameters of the TEE φ_i are the point and the adjustments (Subsection II-A). The parameters of the tasks $\psi_1, \ldots, \psi_\sigma$ are the computations (Subsection II-C).

The *mission* is then $\mathcal{M} : \mathbb{R}^2 \to \mathcal{M}_i$ a function which maps the point $\mathbf{p}_k$ to a specific stage $\mathcal{M}_i$.

For simplicity the system is sampled in discrete time. The algorithm takes as input $\mathcal{M}$, initial position, and energy coefficients guess, and outputs the new position, the instantaneous energy consumption, and the control input–an action performed evolving the mission state.

A. State: position and energy

The state is the UAV's position in space and the energy coefficients in time. We show a linear relation between the instantaneous energy consumption and the energy coefficients in Theorem III.1, but the two are different. We show after the main results how this approach allows us variability in terms of the systems behaving periodically, piece-wise periodically, or merely linearly with sporadic periodicity.

The set

$$\mathcal{P}_i := \{\mathbf{p}_k \mid \varphi_i(\mathbf{p}_k, c_{i,1}, \ldots, c_{i,\rho}) \in \mathbb{C}_i\}, \tag{2}$$

delimits the area where the i-th TEE φ_i is free to evolve using ρ adjustments $\mathbf{c}_i := c_{i,1}, ..., c_{i,\rho}$ (the gray area in Figure 1).

The algorithm uses the set from Equation (2) to select the optimal adjustments $\mathbf{c}_i^0$ s.t. $\varphi_i(\mathbf{p}_k, \mathbf{c}_i^0)$ has the highest instantaneous energy consumption (while still respecting the energy budget). It guides the UAV to the new position $\mathbf{p}_{k+1}$ using the function $\Phi := \varphi_i(\mathbf{p}_k, \mathbf{c}_i^0)$, computing its vector field $\nabla\Phi := (\partial\Phi/\partial x, \partial\Phi/\partial y)$, and deriving the direction to follow in the form of velocity vector

$$\dot{\mathbf{p}}_d(\mathbf{p}_k) := E\nabla\Phi - k_e\Phi\nabla\Phi, \;\; E = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \qquad (3)$$

where E specifies the tracking direction, and $k_e \in \mathbb{R}_{\geq 0}$ the gain to adjusts the speed of convergence. The direction the velocity vector $\dot{\mathbf{p}}_d$ is pointing at is generally different from the course heading due to the atmospheric interference, such as wind ($w \in \mathbb{R}$ in Figure 1).

B. Energy evolution due to trajectory

The algorithm models the energy using as state energy coefficients $\mathbf{q} \in \mathbb{R}^m$ derived from Fourier analysis (the size of the energy coefficients vector m is related to the order of a Fourier series) and decomposes the evolution in energy due to the trajectory and computations. This approach is adapted from our earlier work on computational energy analysis [21], [22], and energy estimation of a fixed-wing UAV [23].

Let us consider a Fourier series of an arbitrary order $r \in \mathbb{Z}_{\geq 0}$ for the purpose of energy consumption modeling of the mission

$$h(k) = \sum_{i=0}^{r} a_i \cos(ik/\xi) + b_i \sin(ik/\xi), \qquad (4)$$

where $h : \mathbb{R}_{\geq 0} \to \mathbb{R}$ maps time to the modeled instantaneous energy consumption, $\xi \in \mathbb{R}$ is the fundamental frequency, and $a, b \in \mathbb{R}$ the Fourier series coefficients.

Lemma II.1 (Fourier series, harmonic oscillator equivalence)**.** Suppose there is no state and output uncertainty (the contributions $\mathbf{w}_k \in \mathbb{R}^m, v_k \in \mathbb{R}$ are zero). The non-linear model in Equation (4) can be expressed using an equivalent linear discrete time-invariant state-space model

$$\begin{aligned} \mathbf{q}_{k+1} &= A\mathbf{q}_k + B\mathbf{u}_k + \mathbf{w}_k, \\ y_k &= C\mathbf{q}_k + v_k, \end{aligned} \qquad (5)$$

where $y_k \in \mathbb{R}_{\geq 0}$ is the instantaneous energy consumption. The state $\mathbf{q}$ are the energy coefficients

$$\begin{aligned} \mathbf{q}_k &= \begin{bmatrix} \alpha_0 & \alpha_1 & \beta_1 & \cdots & \alpha_r & \beta_r \end{bmatrix}^T, \\ A &= \begin{bmatrix} 1 & & & \\ & A_1 & & \\ & & \ddots & \\ & & & A_r \end{bmatrix}, \; A_i = \begin{bmatrix} 0 & i/\xi \\ -i^2/\xi^2 & 0 \end{bmatrix}, \\ C &= \begin{bmatrix} 1 & 1 & 0 & \cdots & 1 & 0 \end{bmatrix}, \end{aligned} \qquad (6)$$

where $\mathbf{q}_k \in \mathbb{R}^m$ given $m = 2r+1$, $A \in \mathbb{R}^{m \times m}$ is the state transmission matrix, and $C \in \mathbb{R}^m$ is the output matrix. In matrix A, the top left entry is one, the diagonal entries are $A_1, \ldots, A_r$, the remaining entries are zero.

Proof. * ■

Suppose at time instant k the mission reached stage i and the control input is $\mathbf{u}_k^a := \langle \mathbf{c}_k, \mathbf{s}_k \rangle$. The control along with the input matrix

$$\begin{aligned} \mathbf{u}_k &= \begin{bmatrix} g(\mathbf{s}_k) - g(\mathbf{s}_{k-1}) & \mathbf{c}_k - \mathbf{c}_{k-1} \end{bmatrix}^T, \\ B &= \begin{bmatrix} 1 & \omega_{i,1} & \cdots & \omega_{i,\rho} \\ & 0 & & \\ & & \ddots & \\ & & & 0 \end{bmatrix}, \end{aligned} \qquad (7)$$

where $\mathbf{s}_k, g(\mathbf{s}_k)$ are the computations and the instantaneous computational energy consumption (Subsection II-C), $\mathbf{u}_k \in \mathbb{R}^n$ is the control given $n = 1+\rho$, and $B \in \mathbb{R}^{m\times n}$. Moreover, the top-left entry of B is one, while the others on the first row are gain factors $\Omega_i := \{\omega_{i,1}, \ldots, \omega_{i,\rho}\} \in \mathbb{R}^\rho$, quantifying the contribution of a given adjustment to the instantaneous energy consumption.

Equation (7) accounts for the energy due to the adjustments and computations. The difference $g(\mathbf{s}_k) - g(\mathbf{s}_{k-1})$ quantifies the change in the instantaneous computational energy consumption. Similarly, $\Omega_i(\mathbf{c}_k - \mathbf{c}_{k-1})$ quantifies the change in the instantaneous trajectory energy consumption. For instance, when the TEE φ_0 is a circle (see Figure 1), a decrement in the adjustment radius of the circle $c_{0,1}$ adds a negative contribution. It thus simulates the lowering of instantaneous energy consumption $\omega_{0,1}(c_{0,1} - c_{0,1}^-) < 0$.

Note from Equation (7) that the control input $\mathbf{u}_k^a$ differs from the nominal control $\mathbf{u}_k$ in Equation (5). The first is an output of the algorithm. The second is a function that maps two consecutive control inputs to the difference in instantaneous energy consumption $\mathbf{u}_k : \mathbb{R}^{2\rho} \times \mathbb{Z}_{\geq 0}^{2\sigma} \to \mathbb{R}^{1+\rho}$.

The energy consumption modeling of the mission necessitates the following assumption.

Assumption II.2 (Energy evelution periodicity)**.** Given two time instants k_1, k_2 s.t. $k_1 > k_2$ and a constant value $n \in \mathbb{Z}_{>0}$

$$|y_k - y_{k+n}| \in \mathbb{E} \subset \mathbb{R}_{\geq 0} \;\; \forall k \in [k_1, k_2]. \qquad (8)$$

Physically this means that the time evolution of the instantaneous energy consumption is assumed to be approximately periodic.

C. Energy evolution due to computations

The energy cost of the computations (the set of tasks being executed on an embedded board on the UAV) is assessed using `powprofiler`, an open-source modeling tool presented in our previous work [21]. The tool measures the instantaneous computational energy consumption of software components within the QoS sets. It builds an energy model: a linear interpolation, one per each task. In a ROS based system, it requires the user to implement one or more ROS nodes changing the computational load by node-specific ROS parameters. The approach has been tested in simulation in our previous work [24].

Specifically, the mission $\mathcal{M}$ contains a set of ordered lists with tasks (recall Definition II.1). These tasks are implemented by software components (such as ROS nodes in a ROS based system) $\Psi(\mathbf{s}_i) := \langle \psi_1(s_{i,1}), \ldots, \psi_\sigma(s_{i,\sigma}) \rangle$. They input the desired and output the actual computations

$$\mathbf{s}_i := \left\{ \langle s_{i,1}, \ldots, s_{i,\sigma} \rangle \mid \psi_j(s_{i,j}) \in \mathbb{S}_i \, \forall j \in [\sigma]^+ \right\}, \tag{9}$$

where $s_{i,j} : \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$ returns the j-th desired computation at stage i, $\mathbf{s}_i \in \mathbb{S}_i \subseteq \mathbb{Z}^\sigma_{\geq 0}$ the union of all QoS sets given $\mathbb{S}_i := \bigcup_{j \in [\sigma]^+} \mathbb{S}_{i,j}$.

For instance, if the task ψ_1 is a CNN object detector, $s_{1,1}$ corresponds to the computation frames-per-second (fps) rate changing the detection frequency. The algorithm outputs the control input which contains the desired fps rate $s_{1,1} \in \mathbb{S}_{1,1}$. The task $\psi_1(s_{1,1})$ outputs the actual fps rate $s_{k,1}$ which might differ from the desired one (e.g., the CNN object detector might not be able to reach a high fps rate with the current computational resources).

Let us further define $g : \mathbb{Z}^\sigma_{\geq 0} \to \mathbb{R}_{\geq 0}$ as the instantaneous computational energy consumption value obtained using `powprofiler`

$$y_k^s := g\left(\Psi(\mathbf{s}_i)\right) = g\left(\mathbf{s}_k\right). \tag{10}$$

Moreover, let $g(\{\emptyset\})$ be zero.

III. Algorithm

Given l stages $\mathcal{M}_i$ (TEE φ_i, tasks Ψ, computations $\mathbf{s}_i \in \mathbb{S}_i$, and adaptations $\mathbf{c}_i \in \mathbb{C}_i$ for all $i \in [l]$), the main purpose of the algorithm is to output a control input sequence $\mathbf{u}^a := \{\mathbf{u}_0^a, \mathbf{u}_1^a, \ldots\}$ in a valid mission.

Definition III.1 (Valid mission)**.** A mission is valid if for every stage $\mathcal{M}_{i-1}$, $i \in [l]^+$ there exist a control input $\mathbf{u}_k^a$ that produce the next stage $\mathcal{M}_i$

$$\begin{aligned}\mathbf{u}_k^a = \{\langle \mathbf{c}_k, \mathbf{s}_k \rangle \mid \exists n \in \mathbb{Z}_{>0}, \\ \langle \varphi_{i-1}(\mathbf{p}_{k-n}, \mathbf{c}_{k-n}), \Psi(\mathbf{s}_{k-n}) \rangle \in \mathcal{M}_{i-1} \\ \implies \langle \varphi_i(\mathbf{p}_k, \mathbf{c}_k), \Psi(\mathbf{s}_k) \rangle \in \mathcal{M}_i\}.\end{aligned} \tag{11}$$

Let us proof that if the mission is valid, the instantaneous energy consumption can be modeled as linear combination of the state from the Equation (5).

Theorem III.1 (Periodic energy model)**.** Consider the mission from Definition II.1, the valid mission from III.1. Assume Assumption II.2 holds, the model of Equation (5) behaves ideally ($\mathbf{w} = \mathbf{0}, v = 0$), the initial energy coefficients state $\mathbf{q}_0$ is y_0^a/m for the first coefficient where $y_0^a \in \mathbb{R}_{>0}$ is an initial measurement[1], $(1/2)y_0^a/m$ for all the others, and the mission is valid. Then, the instantaneous energy consumption y_k is a linear combination of the state $\mathbf{q}_k$.

Proof. The proof is based on mathematical induction. Base case: we proof that $y_0 = y_0^a$. Recall the definition of the state in Equation (6). The output is $y_0 = \alpha_{0,0} + \alpha_{0,1} + \cdots + \alpha_{0,r} = y_0^a/m + (1/2)y_0^a/m + \cdots + (1/2)y_0^a/m = y_0^a$.

Induction step: by inspection of Equation (5), the output at instant k can be expressed $y_k = (\alpha_{0,0} + B\mathbf{u}_0 + \cdots + B\mathbf{u}_{k-1}) + p_1(k)\alpha_{0,1} + \cdots + p_r(k)\alpha_{0,r}$, where $\forall t \in \mathbb{Z}_{\geq 2}$

$$p_r(t) := \begin{cases} \prod_{i=1}^{t/2} r^3/\xi^3 & \text{for even } t \\ (r/\xi) \prod_{i=1}^{(t-1)/2} r^3/\xi^3 & \text{for odd } t \end{cases}. \tag{12}$$

Suppose k is even and the theorem holds up to k. Initial energy coefficients state $\mathbf{q}_0$ leads to $y_k = (y_0^a/m + B\mathbf{u}_0 + \cdots + B\mathbf{u}_{k-1}) + p_1(k)(1/2)y_0^a/m + \cdots + p_r(k)(1/2)y_0^a/m = y_k^a$.

We prove now that the instantaneous energy consumption at $k+1$ is still a linear combination of the state. We express the output in function of the previous state $y_{k+1} = (\alpha_{0,0} + B\mathbf{u}_0 + \cdots + B\mathbf{u}_k) + (1/\xi)\beta_{k,1} + \cdots + (r/\xi)\beta_{k,r}$. Notice that the coefficients α, β have an equivalent evolution (indeed this allows to simulate the periodicity) and $\beta_{k,r} = p_r(k)\beta_{0,r}$. Thus, the output can be expressed $y_{k+1} = (\alpha_{0,0} + B\mathbf{u}_0 + \cdots + B\mathbf{u}_k) + (1/\xi)p_1(k)\beta_{0,1} + \cdots + (r/\xi)p_r(k)\beta_{0,r}$. The expression is equivalent to $y_{k+1} = (\alpha_{0,0} + B\mathbf{u}_0 + \cdots + B\mathbf{u}_k) + p_r(k+1)\beta_{0,r} + \cdots + p_r(k+1)\beta_{0,r}$ using the definition of p_r in Equation (12). Again, the state $\mathbf{q}_0$ leads to $y_{k+1} = (y_0^a/m + B\mathbf{u}_0 + \cdots + B\mathbf{u}_k) + p_r(k+1)(1/2)y_0^a/m + \cdots + p_r(k+1)(1/2)y_0^a/m = y_{k+1}^a$, alike the previous statement, but at instant $k+1$. The proof for odd k is equivalent. ■

A. Output constraints set

We stated earlier the output y_k–the instantaneous energy consumption–evolves in $\mathbb{R}_{\geq 0}$. This is generally untrue. Physical UAVs are bounded by strict energy budgets due to battery limitations.

Let us hence consider the state of charge (SoC) of such battery with a simplistic difference equation [23]

$$\mathrm{SoC}_k = -\left(V - \sqrt{V^2 - 4R_r\tilde{V}y_kV^{-1}}\right)/2R_rQ_c, \tag{13}$$

where $V \in \mathbb{R}$ is the internal battery and $\tilde{V} \in \mathbb{R}$ the stabilized voltage, $R_r \in \mathbb{R}$ the resistance, and $Q_c \in \mathbb{R}$ the constant nominal capacity. We define the output constraints set

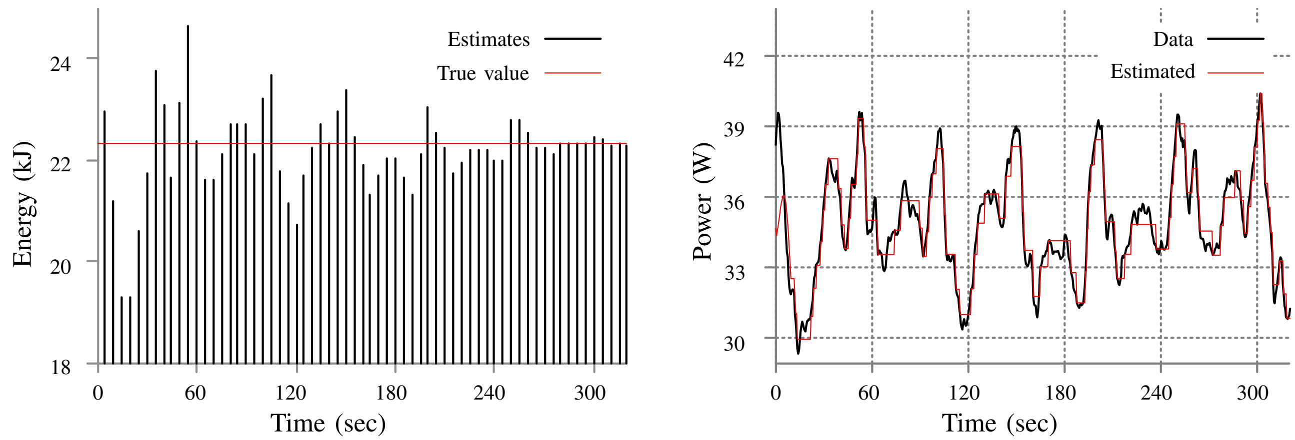
$$\mathbb{Y}_k := \{y_k \mid y_k \in [0, \mathrm{SoC}_kQ_cV] \subseteq \mathbb{R}_{\geq 0}\}, \tag{14}$$

and $\max \mathbb{Y}_k$ is the maximum discharge capacity by the internal battery voltage–the maximum instantaneous energy consumption.

B. Deployment algorithm

procedure $\textsc{Step}(\mathbf{q}_{k-1}, \mathbf{u}_{k-1}^a, \mathbf{u}_{k-2}^a, P_{k-1})$
  $\mathbf{u}_{k-1} \leftarrow \mathbf{u}_{k-1}(\max \mathbf{u}_{k-1}^a, \mathbf{u}_{k-2}^a)$
  $\mathbf{u}_{k-1}^0 \leftarrow \arg\max_{\mathbf{u}} \sum_{i=k-1}^{k+N-2} l(\mathbf{q}_i, \mathbf{u}_i) + V_f(\mathbf{q}_{k+N-1})$
  $\hat{\mathbf{q}}_k^- \leftarrow A\hat{\mathbf{q}}_{k-1} + B\mathbf{u}_{k-1}^0$
  if $C\hat{\mathbf{q}}_k^- \notin \mathbb{Y}_k$ **then**
    $\mathbf{u}_{k-1}^a \leftarrow \mathbf{u}_{k-1}^a / \{\max \mathbf{u}_{k-1}^a\}$
    return $\textsc{Step}(\mathbf{q}_{k-1}, \mathbf{u}_{k-1}^a, \mathbf{u}_{k-2}^a, P_{k-1})$
  else
    if $|y_k^a - C\hat{\mathbf{q}}_k^-| \leq \varepsilon$ **then**
      $\hat{\mathbf{q}}_k \leftarrow \hat{\mathbf{q}}_k^-$
      $P_k \leftarrow P_k^-$
    else
      $P_k^- \leftarrow AP_{k-1}A^T + Q$

[1] y_0^a can be the initial measurement or the measurement from a previous instance of the algorithm



(a) .

(b) .

Fig. 2. .

14: $K \leftarrow P_k^- C^T/(CP_k^- C^T + R)$

15: $\hat{\mathbf{q}}_k \leftarrow \hat{\mathbf{q}}_k^- + K(y_k^a - C\hat{\mathbf{q}}_k^-)$

16: $P_k \leftarrow (I + KC)P_k^-$

17: **end if**

18: $\mathbf{u}_k^a \leftarrow \mathbf{u}_{k-1}^a$

19: **return** $(\hat{\mathbf{q}}_k, P_k, \mathbf{u}_k^a)$

20: **end if**

21: **end procedure**

22: **procedure** EADMPA($\mathcal{M}$, $\mathbf{p}_0$, $\mathbf{q}_0$)

23: $k \leftarrow 0$

24: $\mathbf{u}_{k-1}^a \leftarrow \{\emptyset\}$

25: $\mathbf{p}_k \leftarrow \mathbf{p}_0$

26: $\mathbf{q}_k \leftarrow \mathbf{q}_0$

27: **while** $k \leq t_f$ **do**

28: $\mathbf{u}_k^a \leftarrow \{\mathbf{u}_k^a \mid (\varphi_i(\mathbf{p}_k, \mathbf{c}_i), \Psi(\mathbf{s_i})) \in \lambda(\mathbf{p}_k)\}$

29: $(\mathbf{q}_k, P_k, \mathbf{u}_k^a) \leftarrow$STEP($\mathbf{q}_k$, $\mathbf{u}_k^a$, $\mathbf{u}_{k-1}^a$, P_k)

30: $\mathbf{p}_k \leftarrow \mathbf{p}_k \dot{\mathbf{p}}_d(\mathbf{p}_k)/v$

31: $\mathbf{u}_{k-1}^a \leftarrow \mathbf{u}_k^a$

32: $k \leftarrow k + 1$

33: **end while**

34: **end procedure**

Per each time step k (the final time t_f is unknown), the algorithm updates the state–the position at line 30 and the energy coefficients at line 29–and the control input. Note that the position can be computed directly from Equation (3). If the velocity is $v \in \mathbb{R}_{\geq 0}$, and the starting point $\mathbf{p}_0$, $\mathbf{p}_{k+1} = \mathbf{p}_k \dot{\mathbf{p}}_d(\mathbf{p}_k)/v$.

In detail, initial guess for $P_0 \in \mathbb{R}^{j \times j}$ is positive definite and derived empirically, for $\mathbf{q}_0$ the initial measurement is distributed to the coefficients (see Theorem III.1). Line 2 selects the maximum possible control from the current control input. Line 3 uses robust output feedback model predictive control (MPC) [25] to select the optimal control $\mathbf{u}^0$ for a given horizon $N \in \mathbb{Z}_{>0}$ from the cost function

$$\begin{aligned} l(\mathbf{q}_k, \mathbf{u}_k) &:= (1/2)(\mathbf{q}_k^T Q \mathbf{q}_k + \mathbf{u}_k^T R \mathbf{u}_k), \\ V_f(\mathbf{q}_k) &:= (1/2)(\mathbf{q}_k^T P_f \mathbf{q}_k), \end{aligned} \tag{15}$$

where matrices $Q \in \mathbb{R}^{j \times j}, R \in \mathbb{R}^{l \times l}$ are positive definite.

Follows a check if the mission can finish without the eventuality of battery discharge (output constraints satisfaction) at line 5, with the control input being eventually updated and the process reiterated at line 7.

Before the next step, state estimator–the discrete-time Kalman filter [26] at lines 13–16–predicts the state $\mathbf{q}$ if the modeled instantaneous energy consumption diverges from the sensor's value y_k^a more than a given $\varepsilon \in \mathbb{R}_{\geq 0}$, or sensor measurements are unavailable ($y_k^a = 0$).

IV. Results

In this section

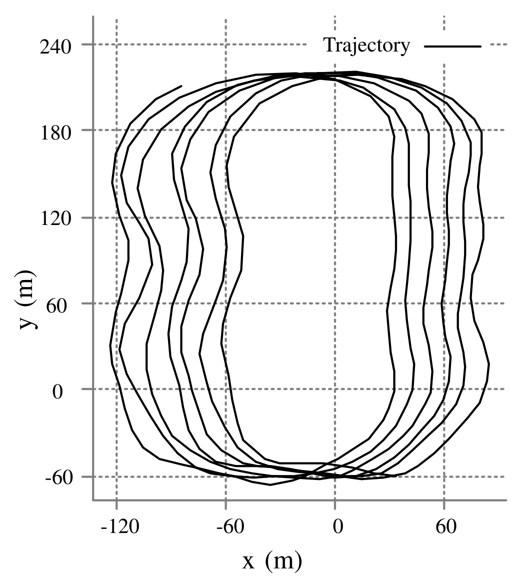


Fig. 3. .

*

V. Conclusion and Future Work

*

REFERENCES

[1] P. Daponte, L. De Vito, L. Glielmo, L. Iannelli, D. Liuzza, F. Picariello, and G. Silano, "A review on the use of drones for precision agriculture," in *IOP Conference Series: Earth and Environmental Science*, vol. 275, no. 1. IOP Publishing, 2019, p. 012022.

[2] Y. Mei, Y.-H. Lu, Y. C. Hu, and C. G. Lee, "Energy-efficient motion planning for mobile robots," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 5. IEEE, 2004, pp. 4344–4349.

[3] M. Wahab, F. Rios-Gutierrez, and A. El Shahat, *Energy modeling of differential drive robots*. IEEE, 2015.

[4] C. H. Kim and B. K. Kim, "Energy-saving 3-step velocity control algorithm for battery-powered wheeled mobile robots," in *Proceedings of the 2005 IEEE international conference on robotics and automation*. IEEE, 2005, pp. 2375–2380.

[5] H. Kim and B.-K. Kim, "Minimum-energy translational trajectory planning for battery-powered three-wheeled omni-directional mobile robots," in *2008 10th International Conference on Control, Automation, Robotics and Vision*. IEEE, 2008, pp. 1730–1735.

[6] Y. Mei, Y.-H. Lu, Y. C. Hu, and C. G. Lee, "A case study of mobile robot's energy consumption and conservation techniques," in *ICAR'05. Proceedings., 12th International Conference on Advanced Robotics, 2005*. IEEE, 2005, pp. 492–497.

[7] 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.

[8] N. Kreciglowa, K. Karydis, and V. Kumar, "Energy efficiency of trajectory generation methods for stop-and-go aerial robot navigation," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2017, pp. 656–662.

[9] Y. Mei, Y.-H. Lu, Y. C. Hu, and C. G. Lee, "Deployment of mobile robots with energy and timing constraints," *IEEE Transactions on robotics*, vol. 22, no. 3, pp. 507–522, 2006.

[10] A. Sadrpour, J. Jin, and A. G. Ulsoy, "Mission energy prediction for unmanned ground vehicles using real-time measurements and prior knowledge," *Journal of Field Robotics*, vol. 30, no. 3, pp. 399–414, 2013.

[11] ——, "Experimental validation of mission energy prediction model for unmanned ground vehicles," in *2013 American Control Conference*. IEEE, 2013, pp. 5960–5965.

[12] H. G. De Marina, Y. A. Kapitanyuk, M. Bronz, G. Hattenberger, and M. Cao, "Guidance algorithm for smooth trajectory tracking of a fixed wing uav flying in wind flows," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 5740–5745.

[13] S. S. H. Hajjaj and K. S. M. Sahari, "Review of research in the area of agriculture mobile robots," in *The 8th International Conference on Robotic, Vision, Signal Processing & Power Applications*. Springer, 2014, pp. 107–117.

[14] F. Qingchun, Z. Wengang, Q. Quan, J. Kai, and G. Rui, "Study on strawberry robotic harvesting system," in *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, vol. 1. IEEE, 2012, pp. 320–324.

[15] F. Dong, W. Heinemann, and R. Kasper, "Development of a row guidance system for an autonomous robot for white asparagus harvesting," *Computers and Electronics in Agriculture*, vol. 79, no. 2, pp. 216–225, 2011.

[16] Z. De-An, L. Jidong, J. Wei, Z. Ying, and C. Yu, "Design and control of an apple harvesting robot," *Biosystems engineering*, vol. 110, no. 2, pp. 112–122, 2011.

[17] A. Aljanobi, S. Al-Hamed, and S. Al-Suhaibani, "A setup of mobile robotic unit for fruit harvesting," in *19th International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD 2010)*. IEEE, 2010, pp. 105–108.

[18] Z. Li, J. Liu, P. Li, and W. Li, "Analysis of workspace and kinematics for a tomato harvesting robot," in *2008 International Conference on Intelligent Computation Technology and Automation (ICICTA)*, vol. 1. IEEE, 2008, pp. 823–827.

[19] Y. Edan, D. Rogozin, T. Flash, and G. E. Miles, "Robotic melon harvesting," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 6, pp. 831–835, 2000.

[20] V. Puri, A. Nayyar, and L. Raja, "Agriculture drones: A modern breakthrough in precision agriculture," *Journal of Statistics and Management Systems*, vol. 20, no. 4, pp. 507–518, 2017.

[21] A. Seewald, U. P. Schultz, E. Ebeid, and H. S. Midtiby, "Coarse-grained computation-oriented energy modeling for heterogeneous parallel embedded systems," *International Journal of Parallel Programming*, pp. 1–22, 2019. [Online]. Available: https://adamseew.bitbucket.io/short/coarse2019

[22] A. Seewald, U. P. Schultz, J. Roeder, B. Rouxel, and C. Grelck, "Component-based computation-energy modeling for embedded systems," in *Proceedings Companion of the 2019 ACM SIGPLAN International Conference on Systems, Programming, Languages, and Applications: Software for Humanity*. ACM, 2019, pp. 5–6. [Online]. Available: https://adamseew.bitbucket.io/short/component2019

[23] A. Seewald, H. Garcia de Marina, H. S. Midtiby, and U. P. Schultz, "Mechanical and computational energy estimation of a fixed-wing drone," in *2020 Fourth IEEE International Conference on Robotic Computing (IRC)*. IEEE, 2020, pp. 135–142. [Online]. Available: https://adamseew.bitbucket.io/short/mechanical2020

[24] G. Zamanakos, A. Seewald, H. S. Midtiby, and U. P. Schultz, "Energy-aware design of vision-based autonomous tracking and landing of a uav," in *2020 Fourth IEEE International Conference on Robotic Computing (IRC)*. IEEE, 2020, pp. 294–297. [Online]. Available: https://adamseew.bitbucket.io/short/energy2020

[25] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model predictive control: theory, computation, and design*. Nob Hill Publishing Madison, WI, 2017, vol. 2.

[26] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.