Energy-Aware Planning-Scheduling for Autonomous Aerial Robots

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

***Abstract*—The paper presents a planning algorithm for autonomous UAVs with energy constraints and solves the problem of dynamic planning of the path and computations simultaneously. The planning strategy is based on a periodic energy model that is empirically motivated and formally proved. The model is enhanced with the time and energy contribution of the path and computations. The algorithm replans both in function of the battery state accounting for uncertainty. The dynamic planning allows to exploit all the available resources, yet avoid possible in-flight failure in case of unexpected battery drops due to, e.g., adverse atmospheric conditions.**

***Index Terms*—Aerial Systems: Perception and Autonomy, Optimization and Optimal Control, Planning, Scheduling and Coordination, Planning under Uncertainty**

I. Introduction

USE CASES involving aerial robots span broadly. They compromise diverse planning and scheduling strategies and often require high autonomy under strict energy budgets. A typical instance is an aerial robot visiting each point in a given space [1], running some on-board computational tasks, a problem in the literature under coverage path planning (CPP) [2], [3]. Here, the aerial robot might detect ground patterns and notify other ground-based actors with little human interaction (see Figure 1). Such use cases arise in, e.g., precision agriculture [4], where harvesting involves ground [5]–[10], damage prevention aerial robots [11], [12]. In these and many others, the robot frequently mounts microcontroller and heterogeneous computing hardware [13] (i.e., with CPUs and GPUs) running power-demanding computational tasks [14]–[17]. We refer to computational tasks that can be scheduled with an energy impact as computations. We are interested in the simultaneous energy optimization of motion plans and computations schedules in-flight and refer to it as energy-aware planning-scheduling. Generic mobile robotics studies that deal with planning-scheduling energy awareness are scarce [18]–[21] and focused on ground-based robots [13], [20]–[23]. Yet, aerial robots are particularly affected by various energy considerations. Indeed it would be generally required to land to recharge the battery [24]. Such a state of practice has prompted us to propose a planning-scheduling approach for autonomous aerial robots. It combines the past body of knowledge, addressing aerial robots' peculiarities such as the atmospheric, battery, and turning radius constraints. Numerical simulations and experimental data of both static and dynamic plans-schedules show improved power savings and fault tolerance with the aerial robots remedying in-flight failures. Figure 1 illustrates the intuition: an aerial robot flies full plan-schedule (i), that is optimized w.r.t. the battery (ii), and altered due to, e.g., unexpected battery defect (iii).

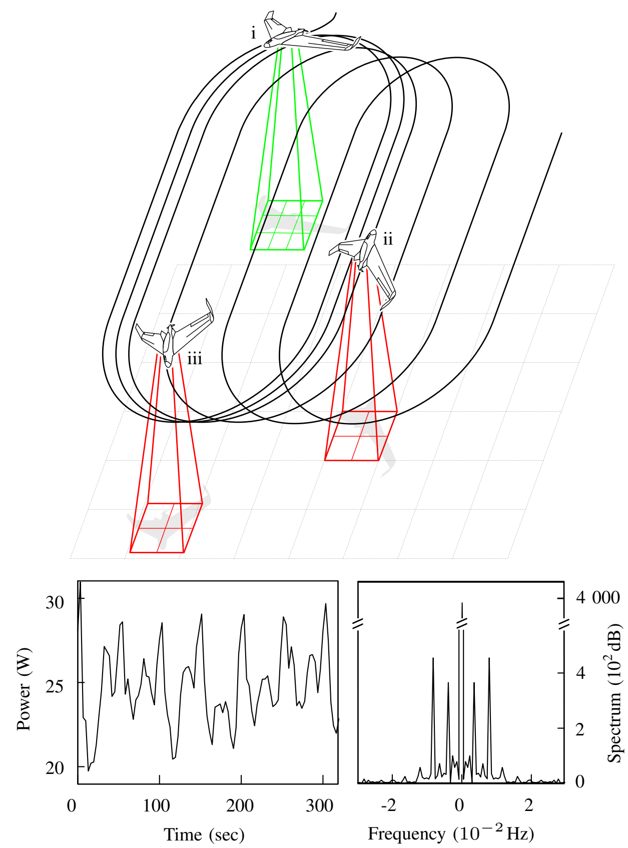


Fig. 1: An illustrative example of planning-scheduling. An initial plan (in i) is re-planned in-flight computations-wise (in ii) and motion-wise (in iii). Follow the collected energy data of a physical aerial robot flying the use case.

There are numerous planning algorithms applied to a variety of robots. An instance is an algorithm selecting an energy-optimized trajectory [25] by, e.g., maximizing the operational time [26]. Many apply to a small number of robots [27] and focus exclusively on planning the trajectory [28], despite evidence of the energy influence of consumptions [13], [18], [19], [21]. In view of the availability of powerful heterogeneous computing hardware [29], the use of computations is further expected to increase in the foreseeable future [30]–[32]. In this context, planning-scheduling energy awareness is a recent research direction. Early studies (2000–2010) varied

Manuscript received: Month, Day, Year; Revised Month, Day, Year; Accepted Month, Day, Year.

This paper was recommended for publication by Editor Editor A. Name upon evaluation of the Associate Editor and Reviewers' comments.

[1]A. S., H. S. M., and U. P. S. are with the Unmanned Aerial Systems Center, Mærsk Mc-Kinney Møller Institute, University of Southern Denmark, Odense. Email: ads@mmmi.sdu.dk;

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

Digital Object Identifier (DOI): see top of this page.

hardware-dependent aspects, e.g., frequency, voltage, along with motion aspects, e.g., motor and travel velocities [13], [18], [22], [33] whereas the literature from the past decade derives energy-aware plans-schedules in broader terms. These include simultaneous considerations for planning-scheduling in perception [21], localization [20], navigation [23], [34], [35], and anytime planning [19]. We extend the relevant literature to the case of CPP with aerial robots.

Our focus is on fixed wings, i.e., airborne robots where wings provide lift, propellers forward thrust, and control surfaces maneuvering. Here, motion and computations energies are within an order of magnitude from each other [36]. Indeed there are other classes where planning-scheduling energy awareness leads to irrelevant savings, i.e., when the motion energy contribution far outreaches the computations and vice-versa. The occurrence happens frequently with rotary-wing aerial robots (e.g., quadrotors or quadcopters, hexacopters, etc.) and lighter-than-air aerial robots (e.g., blimps). It is a common theme with wider planning-scheduling literature, focusing on energy-efficient ground-based robots such as Pioneer 3DX [13], [23], [34], [35], ARC Q14 [20], [21], and Pack-Bot UGV [22].

To guarantee energy awareness, our approach uses optimal control where both the paths and schedules variations are trajectories, varying between given bounds (e.g., physical constraints of the robot and computing hardware, the desired quality of the coverage, etc). Numerous past planning-scheduling studies also employ optimization techniques [18], [20], [21], [33], whereas some others greedy [13], [19], [22] and reinforcement learning-based approaches [23], [34], [35]. Both the variations trajectories are derived for future time instants employing computations and motion energies and battery models. The energy model for the computations uses regressional analysis from our earlier study on heterogeneous computing hardware energy modeling [37], [38], whereas the battery an equivalent circuit model (ECM) from the literature [39]–[41]. The motion wraps the past two in a cohesive model that uses differential and periodic modeling to predict future energy behavior of various plans-schedules. In Figure 1, collected energy data (bottom left) and spectrum analysis (right) of a fixed-wing aerial robot flying CPP motivates the motion energy model: the evolution is periodic, an observation we exploit in the energy models for the computations, motion, and battery models in Section III.

The remaining sections of the letter are then organized as follows. Section II provides basic constructs. These include the concepts of plan, stages, triggering and final points, and path functions, as well as the problem formulation. Section IV describes in detail the planning-scheduling. Section V presents the results and showcases the performances, and Section VI concludes and provides future perspectives. Appendices provide supplementary information.

II. Problem Formulation

Before defining the problem of energy-aware planning-scheduling in Section II-B, Section II-A provides necessary preliminaries. For CPP and, e.g., pattern detections in the agricultural use case, we assume that aerial robot travels a *plan* with information about the path and the schedule.

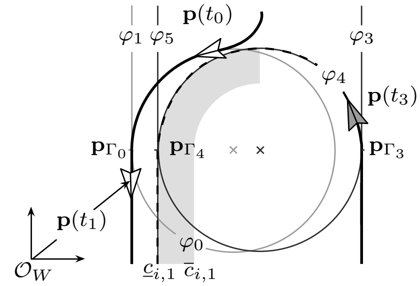


Fig. 2: Definitions II.1–II.4 on a slice of the plan Γ. $\mathbf{p}_{\Gamma_1}, \ldots$ are triggering points, in which proximity happens change of stages $\Gamma_1, \ldots$. Each contains a path function $\varphi_1, \ldots$ and parameters to alter the path and schedule $c_{1,1}, \ldots$.

A. Preliminaries

Definition II.1 (Plan)**.** Given a generic point $\mathbf{p}(t) \in \mathbb{R}^2$ w.r.t. a reference frame $\mathcal{O}_W$ of the aerial robot flying at a given altitude $h \in \mathbb{R}_{>0}$, the *plan* is a finite state machine (FSM) Γ, where the state-transition function $s : \bigcup_i \Gamma_i \times \mathbb{R}^2 \to \bigcup_i \Gamma_i$ maps a stage and a point to the next stage

$$s(\Gamma_i, \mathbf{p}(t)) := \begin{cases} \Gamma_{i+j} & \exists j \in \mathbb{Z}, \text{ if } \|\mathbf{p}(t) - \mathbf{p}_{\Gamma_i}\| < \varepsilon_i \\ \Gamma_i & \text{otherwise} \end{cases}.$$

Here, Γ_i are *stages*, and $\mathbb{Z}$ indicates the set of integers. At each stage, the aerial robot travels a path and runs a schedule on the computing hardware. Both are to be altered in Section IV within given boundaries with *path-* and *computations-*specific *parameters*. ε_i is a stage-dependent value detailed after Definition II.4.

Definition II.2 (Stage)**.** The ith *stage* Γ_i at time instant t of a plan Γ is

$$\Gamma_i := \{\varphi_i(\mathbf{p}(t), c_i^\rho), c_i^\sigma \mid \forall j \in [\rho]_{>0},\, c_{i,j} \in \mathcal{C}_{i,j}, \\ \forall k \in [\sigma]_{>0},\, c_{i,\rho+k} \in \mathcal{S}_{i,k}\},$$

where c_i^ρ and c_i^σ are ρ *path* and σ *computations parameters*. $\mathcal{C}_{i,j} := [\underline{c}_{i,j}, \overline{c}_{i,j}] \subseteq \mathbb{R}$ is the jth path parameter constraint set, and $\mathcal{S}_{i,k} := [\underline{c}_{i,\rho+k}, \overline{c}_{i,\rho+k}] \subseteq \mathbb{Z}_{\geq 0}$ is the kth computation parameter constraint set.

The notation $[\cdot]$ indicates positive naturals up to $\cdot$, i.e., $\{0, 1, \ldots, \cdot\}$. For a set $\mathbb{X}$, $\mathbb{X}_{\geq 0}$ then indicates it is positive, $\mathbb{X}_{>0}$ strictly positive.

The function φ_i is the *path function*, specifying the path. These are stage-dependent mathematical functions the aerial robot tracks as it travels the motion for the coverage. The notation $[\underline{\cdot}, \overline{\cdot}]$ indicates the upper/lower bound of a parameter $\cdot$, i.e.,

$$\underline{\cdot} \leq \cdot \leq \overline{\cdot}. \tag{1}$$

Definition II.3 (Path functions)**.** $\varphi_i : \mathbb{R}^2 \times \mathbb{R}^\rho \to \mathbb{R}$ are *path functions*, forming the path. They are a function of $\mathbf{p}(t)$ and path parameters $c_i^\rho(t)$ and are continuous and twice differentiable.

The change of stages happens in the proximity of given points termed *triggering points*, whereas the plan is complete at the occurrence of the *final point*.

Definition II.4 (Triggering and final points)**.** The *triggering point* $\mathbf{p}_{\Gamma_i}$ allows the transition between stages. The *final point* is the last triggering point $\mathbf{p}_{\Gamma_l}$ relative to the last stage Γ_l.

The stage-dependent value $\varepsilon_i \in \mathbb{R}_{\geq 0}$ in Definition II.1 expresses the radius of an imaginary circle over $\mathbf{p}_{\Gamma_i}$.

Figure 2 illustrates the concepts in Definitions II.1–II.4. $\varphi_1, \ldots, \varphi_6$ are path functions. φ_1 and φ_5 are circles, while φ_2, φ_4, and φ_6 are lines. They are all relative to different stages $\Gamma_1, \Gamma_2, \ldots$. The constraint sets $\mathcal{C}_{1,1}, \mathcal{C}_{2,1}, \ldots$ forms the area where it is possible to alter the paths $\varphi_1, \varphi_2, \ldots$ with the parameters $c_{1,1}, c_{2,1}, \ldots$; the gray area in the figure. The area relative to Γ_5 is bounded by $\underline{c}_{5,1}\overline{c}_{5,1}$.

A convenient way of defining Γ is specifying a set of stages, a shift, and a final point. The set is termed *primitive stages* and iterated with the shift up to reaching the final point.

Definition II.5 (Primitive stages)**.** Given the number of *primitive stages* $n \in \mathbb{Z}_{>0}$, a *shift* $\mathbf{d} \in \mathbb{R}^2$, and a final point $\mathbf{p}_{\Gamma_l}$, the stages $\Gamma_1, \Gamma_2, \ldots, \Gamma_n$ are primitive if they form the remainder of the plan with $\mathbf{d}$ up to $\mathbf{p}_{\Gamma_l}$.

Here, c_i indicates a row vector that contains both the path and computations parameters in sequence, i.e., $c_i := [\,c_i^\rho \quad c_i^\sigma\,]'$, where $\cdot'$ is the transpose of $\cdot$. The path parameters in the definition have a constant distance per each value in $[n]_{>0}$, i.e.,

$$\varphi_{(i-1)n+j}(\mathbf{p} + (i-1)\mathbf{d}, c_1^\rho) - \varphi_{in+j}(\mathbf{p} + i\mathbf{d}, c_1^\rho) = e_j, \quad (2)$$

holds $\forall i \in [l/n - 1]_{>0}, j \in [n]_{>0}$ assuming the total number of stages is known and is $l \in \mathbb{Z}_{>0}$. $e_j \in \mathbb{R}$ given a shift $\mathbf{d}$, initial point $\mathbf{p}$, and initial value of path parameters c_1^ρ.

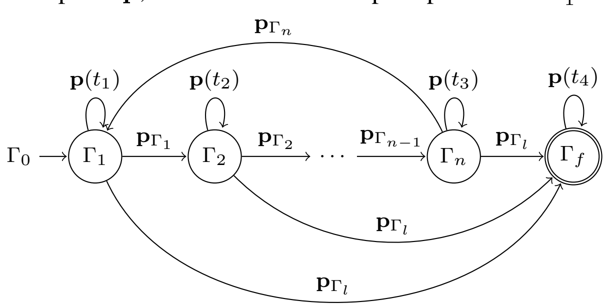


Fig. 3: Definition of a plan Γ with periodic patterns. Stages $\Gamma_1, \Gamma_2, \ldots, \Gamma_n$ containing primitive paths $\varphi_1, \varphi_2, \ldots, \varphi_n$ are iterated with a shift $\mathbf{d}$.

Figure 3 illustrates the concepts in Definition II.5. A plan composed of n stages $\Gamma_1, \Gamma_2, \ldots, \Gamma_n$ (containing primitive paths $\varphi_1, \varphi_2, \ldots, \varphi_n$) is reiterated with the shift $\mathbf{d}$. $t_1 < \cdots < t_4$ are time instant $\in \mathbb{R}_{\geq 0}$. Γ_f is the accepting stage, indicating the plan is completed, Γ_0 the initial stage where the aerial robot, e.g., awaits the starting command.

B. Problem formulation

The problem of planning-scheduling is composed of two sub-problems. One is to form a static plan that visits each point in space, the other to re-plan and re-schedule the plan in-flight in an energy-aware way.

Problem II.1 (Coverage problem)**.** Consider a finite set of vertices of a polygon $v := \{v_1, v_2, \ldots\}$ where each vertex $v_i := (x_{v_i}, y_{v_i}), \forall i \in [|v|]_{>0}$ is a point w.r.t. $\mathcal{O}_W$. Let $\underline{r} \in \mathbb{R}_{\geq 0}$, the minimum turning radius, and $\mathbf{p}(t_0)$, the starting point at the time instant t_0, be given. The *coverage problem* is the problem of finding a plan Γ that covers the polygon.

For a set $\mathbb{X}$, the notation $|\mathbb{X}|$ indicates the cardinality of $\mathbb{X}$.

Problem II.2 (Planning problem)**.** Consider an initial plan Γ in Definition II.1. The *planning problem* is the problem of finding the optimal configuration of path and computations parameters $c_i(t)$, $\forall i \in \{1, 2, \ldots\}$ under energy constraints and uncertainty at each time step t.

III. Energy Models

The solution to Problem II.2 requires accurate energy models, predicting the impact of given path and computations parameters on the battery at future time instants. To this end, Sections III-A–III-C provide models for the motion and computations energies and battery.

A. Energy model for the motion

Let us return to the collected energy data and spectrum analysis in Figure 1, illustrating the energy of a coverage plan Γ with four primitive path functions iterated with a shift. Energy data exhibit periodic behavior, an observation further backed by the power spectrum analysis. It indicates that to model the energy, three frequencies are adequate.

An intuitive way of modeling the energy data is thus a Fourier series of a given order $r \in \mathbb{Z}_{\geq 0}$ and period $T \in \mathbb{R}_{>0}$

$$h(t) = a_0/T + (2/T)\sum_{j=1}^{r}(a_j \cos\omega j t + b_j \sin\omega j t), \quad (3)$$

where $h : \mathbb{R}_{\geq 0} \to \mathbb{R}$ maps time to the instantaneous energy consumption, $\omega := 2\pi/T$ is the angular frequency, and $a, b \in \mathbb{R}$ the series coefficients.

The model in Equation (3) does not the quantify the contribution of path and computations parameters c_i, where, e.g., different schedules results in different instantaneous energy. For this latter purpose, we use another model

$$\dot{\mathbf{q}}(t) = A\mathbf{q}(t) + B\mathbf{u}(t), \quad (4a)$$

$$y(t) = C\mathbf{q}(t), \quad (4b)$$

where $y(t) \in \mathbb{R}$ is the instantaneous energy consumption. The state $\mathbf{q} \in \mathbb{R}^m$ with $m := 2r + 1$ contains energy coefficients

$$\mathbf{q}(t) = \begin{bmatrix}\alpha_0(t) & \alpha_1(t) & \beta_1(t) & \cdots & \alpha_r(t) & \beta_r(t)\end{bmatrix}'. \quad (5)$$

The state transition matrix

$$A = \begin{bmatrix} 0 & 0^{1\times 2} & \ldots & 0^{1\times 2} \\ 0^{2\times 1} & A_1 & \ldots & 0^{2\times 2} \\ \vdots & \vdots & \ddots & \vdots \\ 0^{2\times 1} & 0^{2\times 2} & \ldots & A_r \end{bmatrix}, \; A_j := \begin{bmatrix} 0 & \omega j \\ -\omega j & 0 \end{bmatrix}, \quad (6)$$

where $A \in \mathbb{R}^{m\times m}$ contains r sub-matrices A_j and $0^{i\times j}$ is a zero matrix of i rows and j columns. In matrix A, the top left entry is zero, the diagonal entries are $A_1, \ldots, A_r$, the remaining entries are zeros.

The output matrix

$$C = (1/T)\Big[1 \;\overbrace{1 \quad 0 \quad \cdots \quad 1 \quad 0}^{2r}\Big], \quad (7)$$

where $C \in \mathbb{R}^m$ (the first value in the first column is one, the pattern one–zero is then repeated $2r$ times).

Under some conditions, the models in Equations (3–4) are equal.

Lemma III.1 (Signal, output equality)**.** Given $\mathbf{u}$ a zero vector, matrices A, C described by Equations (6–7), and the initial guess $\mathbf{q}(t_0) = \mathbf{q}_0$ at initial time instant t_0

$$\mathbf{q}_0 = \begin{bmatrix} a_0 & a_1/2 & b_1/2 & \cdots & a_r/2 & b_r/2 \end{bmatrix}'.$$

h in Equations (3) is equal to y in Equation (4).

To define the nominal control and the output matrix, we exploit the effect of variation of path and computations parameters on the energy.

Lemma III.2 (Parameters, energy relation)**.** Given $c_i(t)$ parameters at two following time instants $t \in \{t_j, t_{j+1}\} \subset \mathbb{R}_{\geq 0}$ s.t. $t_j < t_{j+1}$ for an arbitrary stage Γ_i, a change in parameters $c_i(t_j) \neq c_i(t_{j+1})$ results in different overall and instantaneous energies for path and computations parameters respectively.

Appendices A–B contains proofs of Lemmas III.1–III.2.

Using the same notation from Lemma III.2, the nominal control in Equation (4)

$$\mathbf{u}(t_{j+1}) := \hat{\mathbf{u}}(t_{j+1}) - \hat{\mathbf{u}}(t_j), \tag{8}$$

for all time instants. $\hat{\mathbf{u}}$ is then a scale transformation

$$\hat{\mathbf{u}}(t) := \mathrm{diag}(\nu_i) c_i(t) + \tau_i \tag{9}$$

where $\mathrm{diag}(\,\cdot\,)$ is a diagonal matrix with items of a set $\cdot$ on the diagonal and zeros elsewhere. $\nu_i := \begin{bmatrix}\nu_{i,1} & \cdots & \nu_{i,n}\end{bmatrix}'$ and $\tau_i := \begin{bmatrix}\tau_{i,1} & \cdots & \tau_{i,n}\end{bmatrix}'$ are scaling factors with $n := \rho + \sigma$ that transform parameters domain (see Definition II.2) to time and power domains.

Let us assume that the time evolves linearly. Path parameters c_i^ρ can be transformed into a time measure with scaling factors

$$\nu_{i,j} = \left((\bar{t} - \underline{t})/(\bar{c}_{i,j} - \underline{c}_{i,j})\right)/\rho, \tag{10a}$$
$$\tau_{i,j} = \left(\underline{c}_{i,j}(\underline{t} - \bar{t})/(\bar{c}_{i,j} - \underline{c}_{i,j}) + \underline{t}\right)/\rho, \tag{10b}$$

$\forall j \in [\rho]_{>0}$ where $\bar{t}, \underline{t}$ are time measures needed to complete the coverage with configurations $\underline{c}_i^\rho, \bar{c}_i^\rho$.

Similarly to Equation (10), computations parameters c_i^σ can be transformed into an instantaneous energy measure with scaling factors

$$\nu_{i,j} = (g(\bar{c}_{i,j}) - g(\underline{c}_{i,j}))/(\bar{c}_{i,j} - \underline{c}_{i,j}), \tag{11a}$$
$$\tau_{i,j} = \underline{c}_{i,j}(g(\underline{c}_{i,j}) - g(\bar{c}_{i,j}))/(\bar{c}_{i,j} - \underline{c}_{i,j}) + g(\underline{c}_{i,j}), \tag{11b}$$

$\forall j \in [\rho+1, n]$. The function g is detailed in Section III-B and quantifies the power of the computing hardware.

The input matrix then includes the change in energy, i.e.,

$$B = \begin{bmatrix} 0^{1\times\rho} & 1 & \cdots & 1 \\ 0^{1\times\rho} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0^{1\times\rho} & 0 & \cdots & 0 \end{bmatrix}, \tag{12}$$

where $B \in \mathbb{R}^{m\times n}$ contains zeros except the first row where the first ρ columns are still zeros and the remaining σ are ones.

B. Energy model for the computations

Models for heterogeneous computing hardware in the literature often rely on analytical expressions [42]–[45] or different techniques including regressional analysis [37], [46], [47], aiding the selection of hardware or software specific parameters. This section summarizes an energy model from our early studies [37], [38] that relies on regressional analysis aiding to quantify the computations energy of any combination of computations c_i^σ withing the bounds (see Definition II.2).

The model relies on an automatic modeling and profiling tool [37] named `powprofiler` distributed [48] under the open-source M.I.T. license. It is segmented into two layers. In the *measurement layer* the tool measures a discrete set of computations parameters and infers the energy of the remaining in the *predictive layer* via a piece-wise linear regression.

Let us assume there is at least one measuring device, i.e., shunt or internal power resistor, multimeter, or amperometer, quantifying the power drain of a specific component, e.g., CPU, GPU, memory, etc., or of the entire computing hardware.

Definition III.1 (Measurement layer)**.** Given a measuring device, computations parameters, and initial and final time instants, the *measurement layer* is the function $\mathbf{g} : \mathbb{Z}_{>0} \times \mathbb{Z}^\sigma \times \mathcal{T} \to \mathbb{R}$ that returns an energy measure.

Here, the notation $\mathcal{T}$ encloses all the time intervals from initial t_0 to final t_f, i.e., $\mathcal{T} := [t_0, t_f]$.

Definition III.2 (Predictive layer)**.** Given a measuring device and computations parameters the *predictive layer* is the function $\mathbf{g} : \mathbb{Z}_{>0} \times \mathbb{Z}^\sigma \to \mathbb{R}$ that returns and energy measure.

The energy measures in Definitions III.1–III.2 can be either average or overall energy. Additionally, the tool supports the measures of battery state of charge (SoC) detailed in Section III-C. The function g in Definition III.2 is contained in the computations scaling factors in Equation (11) assuming the computations energy behaves linearly between $\underline{c}_i^\sigma$ and $\bar{c}_i^\sigma$, otherwise

$$\begin{aligned} g(c_i^\sigma) = (&\mathbf{g}(\lceil c_i^\sigma \rceil, \mathcal{T}_1) - \mathbf{g}(\lfloor c_i^\sigma \rfloor, \mathcal{T}_2)) \\ &(c_i^\sigma - \lfloor c_i^\sigma \rfloor)/(\lceil c_i^\sigma \rceil - \lfloor c_i^\sigma \rfloor) + \mathbf{g}(\lfloor c_i^\sigma \rfloor, \mathcal{T}_2), \end{aligned} \tag{13}$$

where the notation $\lceil c_i^\sigma \rceil, \lfloor c_i^\sigma \rfloor$ indicates two adjacent measurement layers, and $\mathcal{T}_1, \mathcal{T}_2$ are the corresponding two time intervals. The measuring device in both $\mathbf{g}$ and g is implicit.

C. Battery model

The battery model predicts the battery SoC in function of a given load for future time instants. There are multiple possible models in the literature [49], with varying accuracy, complexity, and easy of implementation ranging from accurate but costly physical models [50], [51], to abstract models [39]–[41], [52] that have compelling trade-offs in terms of ease of implementation and accuracy.

We model a Li-ion battery of an aerial robot flying with an abstract model referred to as "Rint" equivalent circuit model (ECM) in the literature [39]–[41].

Lemma III.3 (Battery SoC)**.** Given the internal battery voltage $V \in \mathbb{R}$ measured in volts, resistance R_r in ohms, constant

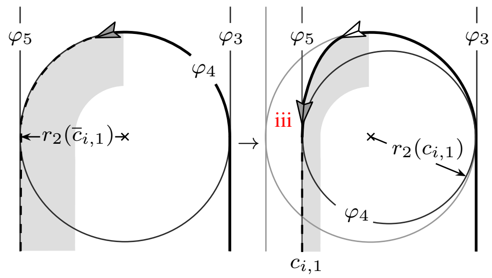


Fig. 4: The alteration of the path parameter $c_{1,1}$, the radius of the circle (it corresponds to the alteration of the plan in Figure 1).

nominal capacity Q_c in amperes per hour, and a battery coefficient k_b, the *battery SoC* evolves

$$\dot{b}(y(t)) = -k_b\left(V - \sqrt{V^2 - 4R_r y(t)}\right)/(2R_r Q_c).$$

Appendix C contains a proof of Lemma III.3.

Equation (4) states that the output y evolves in $\mathbb{R}$. Let us re-evaluate the output constraint with Lemma III.3.

Definition III.3 (Output constraint)**.**

$$\mathcal{Y}(t) := \{y \mid y \in [0, b(y(t))Q_c V] \subseteq \mathbb{R}_{\geq 0}\},$$

is the *output constraint*, where $b(y(t))Q_c V$ is the maximum instantaneous energy consumption.

IV. Planning-Scheduling

This section solves Problems II.1–II.2, i.e., derives a plan in Section IV-A, and re-plans the plan energy-wise in Section IV-B.

A. Coverage path planning

There are various approaches in the literature to solve CPP problems, such as Problem II.1. Approaches that ensure the completeness of the cover are NP-hard [53] and use cellular decomposition, dividing the free-space into sub-regions to be easily covered [2], [3].

An intuitive way to solve the problem is with a back-and-forth motion, sweeping the space delimited by v we term $\mathcal{Q}^v$. Although abundant in both mobile ground-based [2], [54], [55] and aerial [56]–[59] robotics literature, the motion, called *boustrophedon motion* [2], is unsuitable for aerial robots broadly, especially for fixed-wing aerial robots. These robots have reduced maneuverability [60]–[63] and are generally unable to fly quick turns [64].

To address fixed wings and aerial robots generally, this section details a different motion with a wide turning radius. It is similar to another motion in the literature termed *Zamboni motion* [56], but additionally allows variable CPP at the very core of this work. The motion, termed *Zamboni-like motion*, is composed of four primitive paths (see Definition II.5): two lines φ_1, φ_2 and two circles φ_3, φ_4 (see Figure 5). Let us assume the vertices $v_1, v_2, \ldots$ are ordered from the top-left-most vertex in clockwise order, the aerial robot can overfly the edges formed by the vertices, and ${}^{v_x}|_{v_y}$ indicates an edge formed by vertices v_x, v_y.

Algorithm 1 illustrates the procedure to generate the plan Γ that covers $\mathcal{Q}^v$ per each discretized time step, i.e., $\mathcal{T} := \{t_0, t_0 + h, \ldots, t_f\}$ for a given step $h \in \mathbb{R}_{>0}$.

To implement the variable CPP, the radius r_2 of the second circle $\varphi_{|\Gamma|+4}$ on Line 13

$$r_2(c_{i,1}) := \sqrt{r^2 + c_{i,1}}, \quad (14)$$

Algorithm 1 Zamboni-like motion for CPP

1: **for all** $i \in \mathcal{T}$ **do**
2: **if** $\mathbf{p}(i) = \mathbf{p}_{\Gamma_l}$ in Def. II.4 **then return** Γ
3: **if** $\mathbf{p}(i) = \mathbf{p}_{\Gamma_j}$ **then**
4: $j \leftarrow j + 1$
5: **if** $j \notin [n]_{>0}$ **then**
6: $j \leftarrow 1$
7: $\varphi_{|\Gamma|+1} \leftarrow$ line in Def. II.3 parallel to ${}^{v_1}|_{v_{|v|}}$ that intersects $\mathbf{p}_{|\Gamma|}$
8: $\mathbf{p}_{|\Gamma|+1} \leftarrow$ other intersection of $\varphi_{|\Gamma|+1}$ and v
9: $\varphi_{|\Gamma|+2} \leftarrow$ circle whose left most point lays on $\mathbf{p}_{|\Gamma|+1}$
10: $\mathbf{p}_{|\Gamma|+2} \leftarrow$ other inter. of $\varphi_{|\Gamma|+2}$ and v
11: $\varphi_{|\Gamma|+3} \leftarrow$ line par. to $\varphi_{|\Gamma|+1}$ that inter. $\mathbf{p}_{|\Gamma|+2}$
12: $\mathbf{p}_{|\Gamma|+3} \leftarrow$ other inter. of $\varphi_{|\Gamma|+3}$ and v
13: $\varphi_{|\Gamma|+4} \leftarrow$ circle in Eq. (15) whose right most point lays on $\mathbf{p}_{|\Gamma|+3}$
14: $\mathbf{p}_{|\Gamma|+4} \leftarrow$ other inter. of $\varphi_{|\Gamma|+4}$ and v
15: $\Gamma \leftarrow \Gamma \cup \{\Gamma_{|\Gamma|+1}, \ldots, \Gamma_{|\Gamma|+4}\}$ in Defs. II.1–II.2

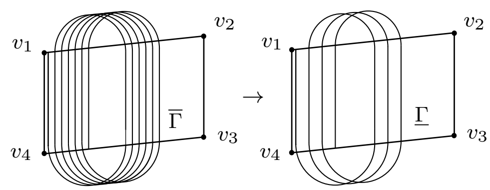


Fig. 5: Zamboni-like motion: ($\bar{\Gamma}$) with four primitive paths (Lines 9–14 in Alg. 1) can be (re)planned ($\underline{\Gamma}$) with r_2 in Eq. (14).

is in function of a path parameter $c_{i,1} \in (\underline{r}^2 - r^2, 0]$ and r is a given ideal turning radius along with the minimum radius (see Problem II.1). The center than also changes

$$\varphi_{|\Gamma|+4} := (x - x_{\mathbf{p}_{|\Gamma|+3}} + r_2)^2 + (y - y_{\mathbf{p}_{|\Gamma|+3}})^2 - r_2^2, \quad (15)$$

where $(x_{\mathbf{p}}, y_{\mathbf{p}}) =: \mathbf{p}$ for any point $\mathbf{p}$. Figure 4 illustrates the concept of $c_{i,1}$ altering the CPP. The radius of the first circle on Line 9 is then $r_1 := r + x_{\mathbf{d}}/2$ (i.e., the radiuses of the two circles ensure that the primitive paths are shifted of $\mathbf{d}$ in Definition II.5).

Algorithm 1 initialize j to minus one and builds the first four primitive functions $\varphi_1, \ldots, \varphi_4$. The remaining Γ is built with the shift $\mathbf{d}$ up to the given final point $\mathbf{p}_{\Gamma_l}$. Initial point is $\mathbf{p}_{\Gamma_1}$, placed s.t. the line φ_1 is at the same distance from an eventual previous line, e.g., $x_{\mathbf{p}_{\Gamma_1}} = x_{v_1} + x_{\mathbf{d}}/2$ in Figure 5.

B. Coverage (re)planning-scheduling

Past literature in planning-scheduling often relies on optimal control and optimization related approaches [18], [20], [21], [33]. We similarly derive an optimal control problem returning the trajectory of parameters $c_i(\mathcal{T})$ with $\mathcal{T} := [t_0, t_f]$ (see Definition III.1). Since the final time instant and the exact value of the state $\mathbf{q}$ are not known, we use a technique in the literature named output model predictive control (MPC) that derives the configuration for a finite horizon on an estimated state $\hat{\mathbf{q}}$, i.e., $t_f := t_0 + N$ for a given $N \in \mathbb{R}_{>0}$.

An optimal control problem (OCP) that selects the highest configuration of c_i and respects the constraints, with $\mathbf{q}(t)$ and $c_i(t)$ the state and parameters trajectories, $l : \mathbb{R}^m \times \mathcal{C}_i \times \mathcal{S}_i \times$

Algorithm 2 Coverage (re)planning-scheduling

1: **for all** $i \in \mathcal{T}$ **do**
16: $\quad \mathbf{q}(\mathcal{K} \setminus \{i+N\}), c_j(\mathcal{K}) \leftarrow$ solve NLP $\arg\max_{\mathbf{q}(k),c_j(k)} l_f(\mathbf{q}(i+N), i+N) + \sum_{k\in\mathcal{K}} l_d(\mathbf{q}(k), c_j(k), k)$ in Eq. (16) on $\mathcal{K} = \{i, i+h, \ldots, i+N\}$
17: $\quad k \leftarrow i$
18: $\quad$ **while** $b_d(C\mathbf{q}(k)) > 0$ **do**
19: $\quad\quad$ **if** $k \in \mathcal{K}$ **then**
20: $\quad\quad\quad \mathbf{q}(k+h) \leftarrow$ solve model in Eq. (4a)
21: $\quad\quad b_d(C\mathbf{q}(k+h)) \leftarrow$ solve model in Lem. III.3
22: $\quad\quad k \leftarrow k+h$
23: $\quad t_b \leftarrow k - i$
24: $\quad t_s \leftarrow (\mathrm{diag}(\nu_j^\rho) c_j^\rho(i) + \tau_j^\rho)\overbrace{[1 \quad 1 \quad \cdots \quad 1]}^{\rho}$
25: $\quad t_r \leftarrow (t_s/\bar{t})(\bar{t} - i)$
26: $\quad$ **if** $t_r < t_b$ **then**
27: $\quad\quad c_j^\rho(i) \leftarrow$ find c_j^ρ with $t_c \in [0, t_b]$, otherwise take $\underline{c}_j^\rho$
28: $\quad \hat{\mathbf{q}}(i+h) \leftarrow$ estimate $\mathbf{q}$ in Eq. (4a) with energy sensor $y(i)$
29: $\quad \hat{y}(i+h) \leftarrow$ derive y from Eq. (4b) with est. state $\hat{\mathbf{q}}(i+h)$

$\mathbb{R}_{\geq 0} \to \mathbb{R}$ and $l_f : \mathbb{R}^m \times \mathbb{R}_{>0} \to \mathbb{R}$ given initial and final cost functions

$$\max_{\mathbf{q}(t),c_i(t)} l_f(\mathbf{q}(t_f), t_f) + \int_{t_0}^{t_f} l(\mathbf{q}(t), c_i(t), t)\, dt, \tag{16a}$$
$$\text{s.t. } \dot{\mathbf{q}} = f(\mathbf{q}(t), c_i(t), t), \tag{16b}$$
$$c_{i,j}(t) \in \mathcal{C}_{i,j},\ c_{i,\rho+k}(t) \in \mathcal{S}_{i,k}\ \forall j \in [\rho]_{>0},\ k \in [\sigma]_{>0}, \tag{16c}$$
$$\mathbf{q}(t) \in \mathbb{R}^m,\ y(t) \in \mathcal{Y}(t), \tag{16d}$$
$$\mathbf{q}(t_0) = \hat{\mathbf{q}}_0 \text{ given (last estimated state), and} \tag{16e}$$
$$b(y(t_0)) = b_0 \text{ given.} \tag{16f}$$

Here, Equation (16b) is the differential periodic energy model in Equation (4). Equation (16c) are the parameters constraints sets in Definition II.2. Equation (16d) is the state and output constraints in Lemma III.3 that requires to evolve the battery model. Equation (16e) is the state guess estimated via state estimationwith the very first estimate given. Equation (16f) is the battery SoC, estimated via a flight controller.

Algorithm 2 implements Equation (16) for the purpose of energy-aware (re)planning-scheduling of Γ from Algorithm 1, i.e, Lines 16–29 continue after Line 15 in Algorithm 1.

Line 16 in Algorithm 2 contains a transformed version of the OCP in Equation (16) into a non-linear program (NLP) that can be easily solved with available NLP solvers [65]. Its solution leads to both trajectories of parameters and states for future N instants. Here, the sets $\mathcal{K}, \mathcal{T}$ have possibly a different step h, tuning the precision. The functions l_d, b_d are merely the discretized versions of Equation (16a) and Lemma III.3, with, e.g., Runge-Kutta methods, Euler method, etc [66].

Lines 17–23 estimate the time needed to completely drain the battery, exploiting the SoC already predicted previously on Line 16. The coverage is then replanned accordingly on Lines 24–27 using Lemma III.2 and scaling factors from Equation (10). Lines 28–29 estimate the energy model's state with current energy sensors readings y (e.g., linear Kalman filter [67]).

V. Results

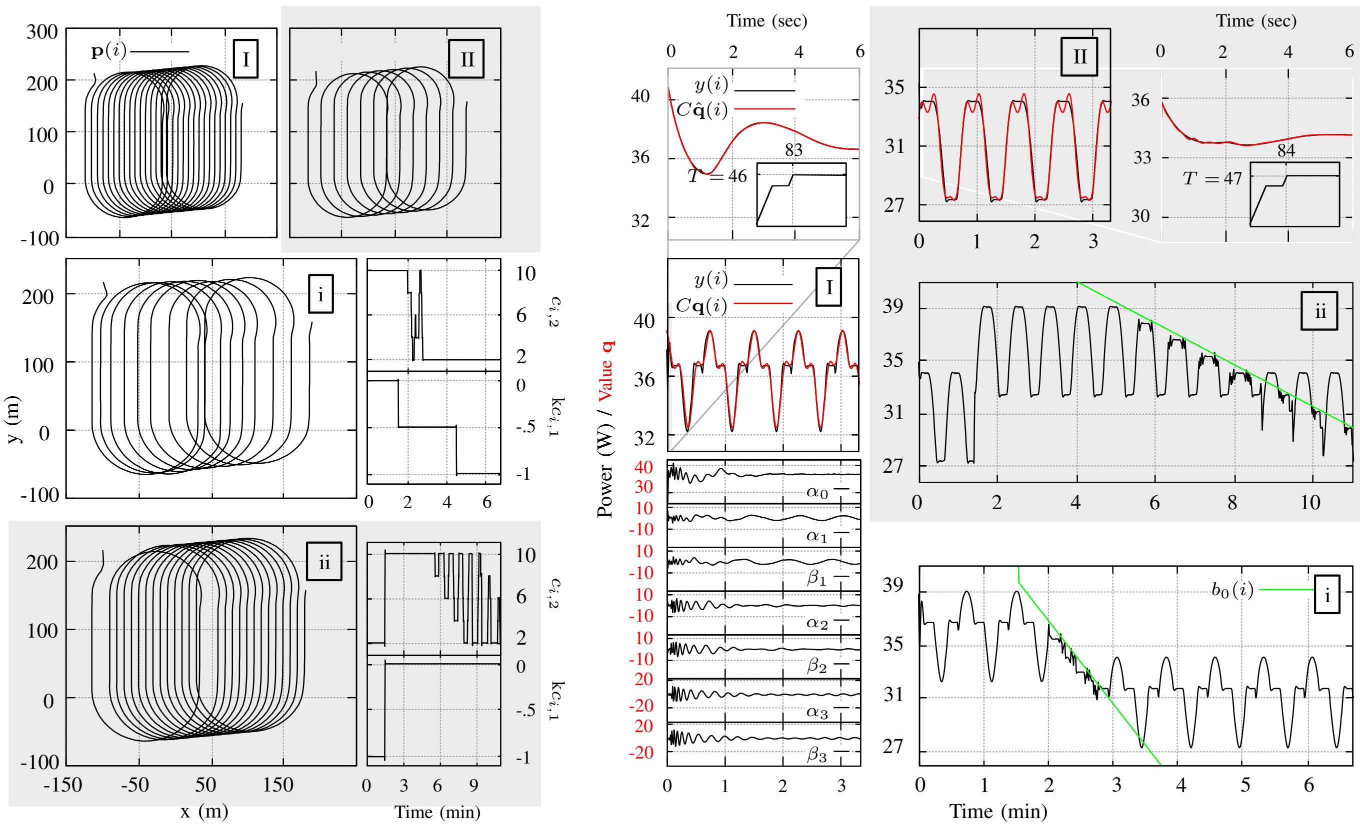
Figure 1 details the data of a physical flight in standard atmospheric conditions. Figure 6 extends the flight with NVIDIA (R) Jetson Nano heterogeneous computing hardware aided by a flight simulation implemented in MATLAB (R). Upper-case roman numerals I,II indicate the plans are static (i.e., solely Algorithm 1), lower-case i,ii exploit planning-scheduling in the letter. The computing hardware caries a camera as a peripheral and is evaluated independently of the aerial robot with `powprofiler` (see Section III-B). The scheduler varies a computation parameter $c_{i,2}$ relative to ground hazards detection rate from two to ten frames-per-second (FPS). The detection uses PedNet, a Convolutional Neural Network (CNN) [68], implemented through Robot Operating System (ROS) middleware [69] as well as the scheduler itself. The planner varies the path parameter $c_{i,1}$ in Equation (14) between zero and -1000 (i.e., the planner-scheduler is the concrete implementation of Algorithms 1–2).

Figure 6a illustrate same plan Γ under different conditions. Flights I–i have a constant wind speed of five meters per second, wind direction of zero degrees, and an initial parameters $c_{i,1}, c_{i,2}$ values of zero and ten (i.e., full r_2 and detection). Flights II–ii (see added gray background for clarity) the same but a wind direction of 90 degrees and the initial parameters values of -1000 and two (i.e., minimum r_2 and detection).

Figure 6b illustrate the power ($y(i)$ on Line 28 in Algorithm 2), and the energy model ($C\mathbf{q}(i)$ on Line 20). Flight i simulates a battery (green line) drop at approximately one minute and a half and four minutes and a half. Planner-scheduler optimizes the path in the proximity of the drops to ensure that the flight is completed, whereas it maximizes the parameter $c_{i,2}$ when the battery is discharging, respecting the output constraint (Definition III.3). Flight ii simulates the opposite scenario. The lowest configuration of parameters and no battery defects. The path parameter increases as soon as the algorithm estimated enough data (two periods T) and the computation parameter decreases mapping the battery discharge rate. For both cases scaling factors are derived empirically, the horizon N is set to six seconds similarly to relevant literature, oder r is three (see Figure 1) and the costs l_d, l_f are merely squared controls. The figure further details the energy model (see detail view for I–II) on an initial slice of the model ($C\hat{\mathbf{q}}(i)$), power ($y(i)$), and period (T).The bottom detail of I illustrates the evolutions of the state $\mathbf{q}$ in time, concluding that approximately two periods are sufficient to obtain a consistent state estimate.

VI. Conclusions and Future Directions

The letter provides a planning-scheduling approach for autonomous aerial robots powered by a limited power source, extending relevant past literature. It proposes a novel coverage motion for variable CPP robust to aerial robots constraints such as the turning radius of fixed wings. Energy modeling in the letter exploits collected empirical data of the fixed-wing aerial robot flying static CPP and further incorporates energy of the computing hardware via the `powprofiler` tool. The approach compromises two algorithms: one derives a static coverage



(a) Path, parameters $(c_{i,1}, c_{i,2})$ evolutions

(b) Energies, model states $(\alpha_0, \dots \in \mathbf{q})$ and parameters (T, b_0) evolutions

Fig. 6: CPP with novel Zamboni-like motion (I,II) and Planning-scheduling of CPP and ground hazards detections with PedNet CNN (i,ii) in terms of the path, energies, and plans-schedules under different conditions (I–i,II–ii), i.e., wind speed and direction, battery behavior, and parameters initial values.

plan, whereas the other (re)plans-schedules the plan on a finite horizon via MPC. It evolves the state of the energy model while optimizing the battery usage. The plan compromise multiple stages, where at each the aerial robot flies a path and does some computations, allowing further extensibility in terms of constructs and approaches.

Indeed, we are currently extending the results to a standard flight controller. The guidance on the coverage needs alike further investigation, as well as the study of the implications of planning-scheduling on other energy-critical mobile robots. Here, our initial investigation led to possible savings [70], in line with relevant literature [20], [21].

References

[1] T. a. M. Cabreira, L. B. Brisolara, and P. R. Ferreira Jr., "Survey on coverage path planning with unmanned aerial vehicles," *Drones*, vol. 3, no. 1, 2019. [Online]. Available: https://www.mdpi.com/2504-446X/3/1/4 1

[2] H. Choset, "Coverage for robotics–a survey of recent results," *Annals of Mathematics and Artificial Intelligence*, vol. 31, pp. 113–126, 2001. 1, 5

[3] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258–1276, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S092188901300167X 1, 5

[4] S. S. H. Hajjaj and K. S. M. Sahari, "Review of research in the area of agriculture mobile robots," in *8th International Conference on Robotic, Vision, Signal Processing & Power Applications*. Springer, 2014, pp. 107–117. 1

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

[6] 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. 1

[7] 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. 1

[8] 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)*. IEEE, 2010, pp. 105–108. 1

[9] 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. 1

[10] 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. 1

[11] 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. 1

[12] 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 *Conference Series: Earth and Environmental Science*, vol. 275, no. 1. IOP Publishing, 2019, p. 012022. 1

[13] 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 *12th International Conference on Advanced Robotics (ICAR)*. IEEE, 2005, pp. 492–497. 1, 2

[14] W. Andrew, C. Greatwood, and T. Burghardt, "Aerial animal biometrics: Individual friesian cattle recovery and visual identification via an autonomous UAV with onboard deep inference," in *International*

Conference on Intelligent Robots and Systems (IROS), 2019, pp. 237–243. 1

[15] T. Peng, D. Zhang, R. Liu, V. K. Asari, and J. S. Loomis, "Evaluating the power efficiency of visual SLAM on embedded GPU systems," in *National Aerospace and Electronics Conference (NAECON)*. IEEE, 2019, pp. 117–121. 1

[16] L. Wang, X. Ye, H. Xing, Z. Wang, and P. Li, "YOLO nano underwater: A fast and compact object detector for embedded device," in *Global Oceans Conference*. IEEE, 2020, pp. 1–4. 1

[17] G. Alexey, V. Klyachin, K. Eldar, and A. Driaba, "Autonomous mobile robot with AI based on Jetson Nano," in *Future Technologies Conference (FTC)*. Springer, 2021, pp. 190–204. 1

[18] J. Brateman, C. Xian, and Y.-h. Lu, "Energy-effcient scheduling for autonomous mobile robots," in *IFIP International Conference on Very Large Scale Integration*. IEEE, 2006, pp. 361–366. 1, 2, 5

[19] S. Sudhakar, S. Karaman, and V. Sze, "Balancing actuation and computing energy in motion planning," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4259–4265. 1, 2

[20] M. Lahijanian, M. Svorenova, A. A. Morye, B. Yeomans, D. Rao, I. Posner, P. Newman, H. Kress-Gazit, and M. Kwiatkowska, "Resource-performance tradeoff analysis for mobile robots," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1840–1847, 2018. 1, 2, 5, 7

[21] P. Ondrúška, C. Gurău, L. Marchegiani, C. H. Tong, and I. Posner, "Scheduled perception for energy-efficient path following," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 4799–4806. 1, 2, 5, 7

[22] 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. 1, 2

[23] D.-K. Ho, K. Ben Chehida, B. Miramond, and M. Auguin, "Learning-based adaptive management of QoS and energy for mobile robotic missions," *International Journal of Semantic Computing*, vol. 13, no. 04, pp. 513–539, 2019. 1, 2

[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 *4th International Conference on Robotic Computing (IRC)*. IEEE, 2020, pp. 294–297, https://adamseewald.cc/short/energy2020. [Online]. Available: https://adamseewald.cc/short/energy2020 1

[25] Y. Mei, Y.-H. Lu, Y. C. Hu, and C. G. Lee, "Energy-efficient motion planning for mobile robots," in *International Conference on Robotics and Automation (ICRA)*, vol. 5. IEEE, 2004, pp. 4344–4349. 1

[26] M. Wahab, F. Rios-Gutierrez, and A. El Shahat, "Energy modeling of differential drive robots," in *Southeast Conference (SoutheastCon)*. IEEE, 2015. 1

[27] C. H. Kim and B. K. Kim, "Energy-saving 3-step velocity control algorithm for battery-powered wheeled mobile robots," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2005, pp. 2375–2380. 1

[28] H. Kim and B.-K. Kim, "Minimum-energy translational trajectory planning for battery-powered three-wheeled omni-directional mobile robots," in *10th International Conference on Control, Automation, Robotics and Vision*. IEEE, 2008, pp. 1730–1735. 1

[29] S. T. H. Rizvi, G. Cabodi, D. Patti, and M. M. Gulzar, "A general-purpose graphics processing unit (GPGPU)-accelerated robotic controller using a low power mobile platform," *Journal of Low Power Electronics and Applications*, vol. 7, no. 2, p. 10, 2017. 1

[30] A. Abramov, K. Pauwels, J. Papon, F. Worgotter, and B. Dellen, "Real-time segmentation of stereo videos on a portable system with a mobile GPU," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 9, pp. 1292–1305, 2012. 1

[31] M. T. Satria, S. Gurumani, W. Zheng, K. P. Tee, A. Koh, P. Yu, K. Rupnow, and D. Chen, "Real-time system-level implementation of a telepresence robot using an embedded GPU platform," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2016, pp. 1445–1448. 1

[32] U. Jaramillo-Avila, J. M. Aitken, and S. R. Anderson, "Visual saliency with foveated images for fast object detection and recognition in mobile robots using low-power embedded GPUs," in *19th International Conference on Advanced Robotics (ICAR)*. IEEE, 2019, pp. 773–778. 1

[33] W. Zhang and J. Hu, "Low power management for autonomous mobile robots using optimal control," in *46th Conference on Decision and Control (CDC)*. IEEE, 2007, pp. 5364–5369. 2, 5

[34] D.-K. Ho, K. Ben Chehida, B. Miramond, and M. Auguin, "QoS and energy-aware run-time adaptation for mobile robotic missions: A learning approach," in *3rd International Conference on Robotic Computing (IRC)*. IEEE, 2019, pp. 212–219. 2

[35] ——, "Towards a multi-mission QoS and energy manager for autonomous mobile robots," in *2nd International Conference on Robotic Computing (IRC)*. IEEE, 2018, pp. 270–273. 2

[36] A. Seewald, H. García de Marina, H. S. Midtiby, and U. P. Schultz, "Mechanical and computational energy estimation of a fixed-wing drone," in *4th International Conference on Robotic Computing (IRC)*. IEEE, 2020, pp. 135–142, https://adamseewald.cc/short/mechanical2020. [Online]. Available: https://adamseewald.cc/short/mechanical2020 2

[37] 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*, vol. 49, no. 2, pp. 136–157, 2021, https://adamseewald.cc/short/coarse2019. [Online]. Available: https://adamseewald.cc/short/coarse2019 2, 4

[38] A. Seewald, U. P. Schultz, J. Roeder, B. Rouxel, and C. Grelck, "Component-based computation-energy modeling for embedded systems," in *SIGPLAN International Conference on Systems, Programming, Languages, and Applications: Software for Humanity (SPLASH)*. ACM, 2019, pp. 5–6, https://adamseewald.cc/short/component2019. [Online]. Available: https://adamseewald.cc/short/component2019 2, 4

[39] H. He, R. Xiong, and J. Fan, "Evaluation of lithium-ion battery equivalent circuit models for state of charge estimation by an experimental approach," *Energies*, vol. 4, no. 4, pp. 582–598, 2011. [Online]. Available: https://www.mdpi.com/1996-1073/4/4/582 2, 4

[40] H. Hinz, "Comparison of lithium-ion battery models for simulating storage systems in distributed power generation," *Inventions*, vol. 4, 2019. [Online]. Available: https://www.mdpi.com/2411-5134/4/3/41 2, 4

[41] S. Mousavi G. and M. Nikdel, "Various battery models for various simulation studies and applications," *Renewable and Sustainable Energy Reviews*, vol. 32, pp. 477–485, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032114000598 2, 4

[42] A. Marowka, "Energy-aware modeling of scaled heterogeneous systems," *International Journal of Parallel Programming*, vol. 45, no. 5, pp. 1026–1045, 2017. 4

[43] M. Goraczko, J. Liu, D. Lymberopoulos, S. Matic, B. Priyantha, and F. Zhao, "Energy-optimal software partitioning in heterogeneous multiprocessor embedded systems," in *45th ACM/IEEE Design Automation Conference*. IEEE, 2008, pp. 191–196. 4

[44] E. Calore, S. F. Schifano, and R. Tripiccione, "Energy-performance tradeoffs for HPC applications on low power processors," in *European Conference on Parallel Processing*. Springer, 2015, pp. 737–748. 4

[45] T.-J. Yang, Y.-H. Chen, and V. Sze, "Designing energy-efficient convolutional neural networks using energy-aware pruning," in *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 5687–5695. 4

[46] P. E. Bailey, D. K. Lowenthal, V. Ravi, B. Rountree, M. Schulz, and B. R. De Supinski, "Adaptive configuration selection for power-constrained heterogeneous systems," in *43rd International Conference on Parallel Processing*. IEEE, 2014, pp. 371–380. 4

[47] K. Ma, X. Li, W. Chen, C. Zhang, and X. Wang, "GreenGPU: A holistic approach to energy efficiency in GPU-CPU heterogeneous architectures," in *41st International Conference on Parallel Processing*. IEEE, 2012, pp. 48–57. 4

[48] A. Seewald, U. P. Schultz, E. Ebeid, and H. S. Midtiby, "`powprofiler` computations energy modeling tool," https://doi.org/10.5281/zenodo.5562457, oct 2021. [Online]. Available: https://doi.org/10.5281/zenodo.5562457 4

[49] R. Rao, S. Vrudhula, and D. N. Rakhmatov, "Battery modeling for energy aware system design," *Computer*, vol. 36, no. 12, pp. 77–87, 2003. 4

[50] S. J. Moura, F. B. Argomedo, R. Klein, A. Mirtabatabaei, and M. Krstic, "Battery state estimation for a single particle model with electrolyte dynamics," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 2, pp. 453–468, 2017. 4

[51] J. Marcicki, M. Canova, A. T. Conlisk, and G. Rizzoni, "Design and parametrization analysis of a reduced-order electrochemical model of graphite/LiFePO4 cells for SoC/SoH estimation," *Journal of Power Sources*, vol. 237, pp. 310–324, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378775313000694 4

[52] Y. Xing, W. He, M. Pecht, and K. L. Tsui, "State of charge estimation of lithium-ion batteries using the open-circuit voltage at various ambient temperatures," *Applied Energy*, vol. 113, pp. 106–115, 2014.

[Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261913005746 4
[53] E. M. Arkin, S. P. Fekete, and J. S. Mitchell, "Approximation algorithms for lawn mowing and milling," *Computational Geometry*, vol. 17, no. 1, pp. 25–50, 2000. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925772100000158 5
[54] H. M. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, S. Thrun, and R. C. Arkin, *Principles of robot motion: Theory, algorithms, and implementation*. MIT press, 2005. 5
[55] S. M. LaValle, *Planning algorithms*. Cambridge University Press, 2006. [Online]. Available: http://lavalle.pl/planning/ 5
[56] J. Araújo, P. Sujit, and J. Sousa, "Multiple UAV area decomposition and coverage," in *Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*. IEEE, 2013, pp. 30–37. 5
[57] O. Artemenko, O. J. Dominic, O. Andryeyev, and A. Mitschele-Thiel, "Energy-aware trajectory planning for the localization of mobile devices using an unmanned aerial vehicle," in *25th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 2016, pp. 1–9. 5
[58] T. a. M. Cabreira, C. D. Franco, P. R. Ferreira, and G. C. Buttazzo, "Energy-aware spiral coverage path planning for UAV photogrammetric applications," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3662–3668, 2018. 5
[59] C. Di Franco and G. Buttazzo, "Energy-aware coverage path planning of UAVs," in *International Conference on Autonomous Robot Systems and Competitions*. IEEE, 2015, pp. 111–117. 5
[60] M. Dille and S. Singh, "Efficient aerial coverage search in road networks," in *Guidance, Navigation, and Control (GNC) Conference*. AIAA, 2013, pp. 1–20. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2013-5094 5
[61] R. Mannadiar and I. Rekleitis, "Optimal coverage of a known arbitrary environment," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2010, pp. 5525–5530. 5
[62] A. Xu, C. Viriyasuthee, and I. Rekleitis, "Optimal complete terrain coverage using an unmanned aerial vehicle," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 2513–2519. 5
[63] ——, "Efficient complete coverage of a known arbitrary environment with applications to aerial operations," *Autonomous Robots*, vol. 36, no. 4, pp. 365–381, 2014. 5
[64] X. Wang, P. Jiang, D. Li, and T. Sun, "Curvature continuous and bounded path planning for fixed-wing UAVs," *Sensors*, vol. 17, no. 9, 2017. [Online]. Available: https://www.mdpi.com/1424-8220/17/9/2155 5
[65] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model predictive control: Theory, computation, and design*. Nob Hill Publishing, 2017, vol. 2. 6
[66] A. Iserles, *A first course in the numerical analysis of differential equations*. Cambridge University Press, 2009, no. 44. 6
[67] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 03 1960. [Online]. Available: https://doi.org/10.1115/1.3662552 6
[68] M. Ullah, A. Mohammed, and F. Alaya Cheikh, "PedNet: A spatio-temporal deep convolutional neural network for pedestrian segmentation," *Journal of Imaging*, vol. 4, no. 9, p. 107, 2018. 6
[69] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: An open-source robot operating system," in *ICRA Workshop on Open Source Software*, vol. 3, no. 3.2, 2009, p. 5. 6
[70] A. Seewald, "Beyond traditional energy planning: The weight of computations in planetary exploration," in *IROS Workshop on Planetary Exploration Robots: Challenges and Opportunities (PlanRobo)*. ETH Zürich, Department of Mechanical and Process Engineering, 2020, p. 3, https://doi.org/10.3929/ethz-b-000450120. [Online]. Available: https://adamseewald.cc/short/beyond2020 7
[71] B. Kuo, *Automatic control systems*, ser. Electrical engineering series. Prentice-Hall, 1967. 9
[72] K. Ogata, *Modern control engineering*. Prentice-Hall, 2002. 9
[73] C. Moler and C. Van Loan, "Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later," *SIAM review*, vol. 45, no. 1, pp. 3–49, 2003. 9

Appendix A
Proof of Lemma III.1

We propose a formal proof of Lemma III.1. The proof justifies the choice of the items of the matrices A, C and of the initial guess $\mathbf{q}_0$ in Equation (**??**). We write these elements such that the coefficients of the series $a_0, \dots, b_r$ are the same as the coefficients of the state $\alpha_0, \dots, \beta_r$.

Let us re-write the Fourier series expression in Equation (3) in its complex form with the well-known Euler's formula $e^{it} = \cos t + i \sin t$. With $t = \omega j t$, we find the expression for $\cos \omega j t = (e^{i\omega j t} + e^{-i\omega j t})/2$ and $\sin \omega j t = (e^{i\omega j t} - e^{-i\omega j t})/(2i)$ by substitution of $\sin \omega j t$ and $\cos \omega j t$ respectively. This leads [71]

$$\begin{aligned} h(t) = a_0/T + (1/T) \sum_{j=1}^{r} e^{i\omega j t}(a_j - i b_j) + \\ (1/T) \sum_{j=1}^{r} e^{-i\omega j t}(a_j + i b_j), \end{aligned} \tag{17}$$

where i is the imaginary unit.

The solution at time t can be expressed $\mathbf{q} = e^{At}\mathbf{q}_0$. Both the solution and the system in Equation (4) are well established expressions derived using standard textbooks [71], [72]. To solve the matrix exponential e^{At}, we use the eigenvectors matrix decomposition method [73].

The method works on the similarity transformation $A = VDV^{-1}$. The power series definition of e^{At} implies $e^{At} = Ve^{Dt}V^{-1}$ [73]. We consider the non-singular matrix V, whose columns are eigenvectors of A; $V := \begin{bmatrix} v_0 & v_1^0 & v_1^1 & \dots & v_r^0 & v_r^1 \end{bmatrix}$. We then consider the diagonal matrix of eigenvalues $D = \operatorname{diag}(\lambda_0, \lambda_1^0, \lambda_1^1, \dots, \lambda_r^0, \lambda_r^1)$. λ_0 is the eigenvalue associated to the first item of A. λ_j^0, λ_j^1 are the two eigenvalues associated with the block A_j. We can write $Av_j = \lambda_j v_j \ \ \forall j = \{1, \dots, m\}$, and $AV = VD$.

We apply the approach in terms of Equation (4), under the assumptions made in the lemma (the control is a zero vector); $\dot{\mathbf{q}} = A\mathbf{q}$. The linear combination of the initial guess and the generic solution

$$\begin{aligned} F\mathbf{q}(0) = \gamma_0 v_0 + \sum_{k=0}^{1} \sum_{j=1}^{r} \gamma_j v_j^k \\ F\mathbf{q}(t) = \gamma_0 e^{\lambda_0 t} v_0 + \sum_{k=0}^{1} \sum_{j=1}^{r} \gamma_j e^{\lambda_j t} v_j^k \end{aligned} \tag{18}$$

where $F = \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix}$ is a properly sized vector of ones.

Let us consider the second expression in Equation (18). It represents the linear combination of all the coefficients of the state at time t. It can also be expressed in the following form

$$\begin{aligned} F\mathbf{q}(t)/T = \gamma_0 e^{\lambda_0 t} v_0/T + (1/T) \sum_{j=1}^{r} \gamma_j e^{\lambda_j^0 t} v_j^0 + \\ (1/T) \sum_{j=1}^{r} \gamma_j e^{\lambda_j^1 t} v_j^1. \end{aligned} \tag{19}$$

We proof that the eigenvalues λ and eigenvectors V are such that Equation (19) is equivalent to Equation (17).

The matrix A is a block diagonal matrix, so we can express its determinant as the multiplication of the determinants of its blocks $\det(A) = \det(0) \times \det(A_1) \times \cdots \times \det(A_r)$. We proof the first determinant and the others separately.

Thereby we start by proofing that the first terms of the Equation (17) and (19) match. We find the eigenvalue from $\det(0) = 0$, which is $\lambda_0 = 0$. The corresponding eigenvector can be chosen arbitrarily $(0-\lambda_0)v_0 = \begin{bmatrix}0 & \cdots & 0\end{bmatrix}$ $\forall v_0$, thus we choose $v_0 = \begin{bmatrix}1 & 0 & \cdots & 0\end{bmatrix}$. We find the value γ_0 of the vector γ so that the terms are equal, $\gamma_0 = \begin{bmatrix}a_0 & 0 & \cdots & 0\end{bmatrix}$.

Then, we proof that all the terms in the sum of both the Equations (17) and (19) match.

For the first block A_1, we find the eigenvalues from $\det(A_1 - \lambda I) = 0$. The polynomial $\lambda^2 + \omega^2$, gives two complex roots–the two eigenvalues $\lambda_1^0 = i\omega$ and $\lambda_1^1 = -i\omega$. The eigenvector associated with the eigenvalue λ_1^0 is $v_1^0 = \begin{bmatrix}0 & -i & 1 & 0 & \cdots & 0\end{bmatrix}^T$. The eigenvector associated with the eigenvalue λ_1^1 is $v_1^1 = \begin{bmatrix}0 & i & 1 & 0 & \cdots & 0\end{bmatrix}^T$. Again, we find the values γ_1 of the vector γ such that the equivalences

$$\begin{cases} e^{i\omega t}(a_1 - ib_1) & = \gamma_1 e^{i\omega t} v_1^0 \\ e^{-i\omega t}(a_1 + ib_1) & = \gamma_1 e^{i\omega t} v_1^1 \end{cases}$$

hold. They hold for $\gamma_1 = \begin{bmatrix}b_1 & a_1\end{bmatrix}$.

The proof for the remaining $r-1$ blocks is equivalent.

The initial guess is build such that the sum of the coefficients is the same in both the signals. In the output matrix, the frequency $1/T$ accounts for the period in Equation (17) and (19) and (3). At time instant zero, the coefficients b_j are not present and the coefficients a_j are doubled for each $j = 1, 2, \ldots, r$ (thus we multiply by a half the corresponding coefficients in $\mathbf{q}_0$). To match the outputs $h(t) = y(t)$–or equivalently $F\mathbf{q}(t)/T = C\mathbf{q}(t)$–we define $C = (1/T)\begin{bmatrix}1 & 1 & 0 & \cdots & 1 & 0\end{bmatrix}$. We thus conclude that the signal and the output are equal, hence the lemma holds. ■

We note for practical reasons that the signal would still be periodic with another linear combination of coefficients (for instance, $C = d\begin{bmatrix}1 & 0 & 1 & \cdots & 0 & 1\end{bmatrix}$, or $d\begin{bmatrix}1 & \cdots & 1\end{bmatrix}$ for a constant value $d \in \mathbb{R}$).

APPENDIX B
PROOF OF LEMMA III.2

APPENDIX C
PROOF OF LEMMA III.3