# 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 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 along with strictly limited energy budgets–with the typical instance being a UAV used to inform its grounded counterparts of patterns detected while flying. Currently, UAVs flying outdoors are often semi-autonomous, in the sense that the mission is static and usually defined using a 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 attempts to combine and generalize some of the past body of knowledge on the mobile robot planning problem, and highlights the increasing *computational demands* and their relation to energy consumption, path, and autonomy.

Planning algorithms for mobile robots broadly are not a new concept, in that they are correlated to such topics as trajectory generation and path planning. Generally, these algorithms select an energy-optimized trajectory [2], by e.g., maximizing the operational time [3], but in practice apply to few robots [4], and focus on optimizing motion control for these robots [5], despite compelling evidence for the systems' energy being influenced by the computations over bare motion [6]. For UAVs specifically, rotorcrafts have equally gained research interest in terms of algorithms for energy-optimized trajectory generation [7], [8]. Furthermore, past mission planning algorithms–which include a broader concept of a mission being a set of tasks along with a motion plan–also focus on the trajectory [6], [9], and apply to few robots [10], [11]. Yet, computations of such systems are only expected to increase in the near future.

The proposed algorithm alters the energy consumption dynamically by means of mission-specific parameters: the Quality of Service (QoS) of the computations, and the trajectory-explicit equations (TEEs) adjustments. In the remainder of the paper, we strictly adopt the following notation. We refer to the values of mission-specific QoS and TEEs parameters as computations and adjustments, to the constraints sets that delimit such computations and adjustments as QoS and TEEs sets, and to the current trajectory as TEE. Our goal is a mission extension by optimizing both computations and adjustments as the UAV flies and its batteries drain. First, the algorithm optimizes computations requiring the UAV to include robot operating system (ROS) nodes. Then, it optimizes adjustments–a way to alter the trajectory–and guides 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*. The periodicity is directly observed, by e.g., the UAV flying in repetitive patterns, and the uncertainty accounts for the environmental interference with e.g., a fixed-wing UAV drifting due to windy weather. It addresses the periodicity modeling the energy with Fourier analysis–being the mission periodic, we expect the energy to evolve also periodically–and the uncertainty with a state estimator. It output the controls sequence (computations and adjustments) using robust output feedback 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 consists of a UAV flying in circles and lines covering a polygon, detecting obstacles using a convolutional neural network (CNN), and informing grounded mobile robots employed for future harvesting–a survey mission optimized for the craft's dynamics. The algorithm plans the mission controlling the processing rate, the radius of the circles, and the distance between the lines. Data indicates a potential extension of up to 13 minutes over an hour by merely switching to the lowest computations.

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 mobile

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.

robot 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

Let a mission's stage be a generic continuous twice differentiable TEE $\varphi$ and a set of tasks $\psi_1, \ldots, \psi_\sigma$. Moreover, let $t_f$ be the final time, $[\cdot]$ the set $\{0, 1, \ldots, \cdot\}$, $[\cdot]^+$ the set $\{0\}/[\cdot]$, and $\underline{\cdot}, \overline{\cdot}$ the lower and upper bound of $\cdot$. If $\cdot$ is a value, the bounds are user defined and retrieved from a lookup table; if a set, they correspond to minimum and maximum of the set.

**Definition II.1** (Stage and mission)**.** At a time instant $k \in [0, t_f)$, the *stage* is defined as the ordered list

$$\begin{aligned}\mathcal{M}_k := \{(\varphi_k(\cdot), \psi_1(\cdot), ..., \psi_\sigma(\cdot)) \mid \\ \exists\ \varphi_k(\cdot) \in \mathbb{C}_k, \\ \psi_i(\cdot) \in \mathbb{S}_{k,i}\ \forall i \in [\sigma]\},\end{aligned} \tag{1}$$

where $\mathbb{C}_k := [\underline{c}_k, \overline{c}_k] \subseteq \mathbb{R}$ is the TEEs set, and $\mathbb{S}_{k,i} := [\underline{s}_{k,i}, \overline{s}_{k,i}] \subseteq \mathbb{Z}_{\geq 0}$ the $i$-th task QoS set. The parameters of the TEE $\varphi_k$ are defined in Subsection II-A, of the tasks $\psi_1, \ldots, \psi_\sigma$ in II-C.

The *mission* is the union of all the stages, along with a function $\lambda : \mathbb{R}^2 \to \mathcal{M}_k$ which maps a point $\mathbf{p} \in \mathbb{R}^2$ of a UAV flying at an assigned altitude $h \in \mathbb{R}_{>0}$ w.r.t. some inertial navigation frame $\mathcal{O}_W$ to a specific stage. If for simplicity the system is sampled discrete-time

$$\mathcal{M} := (\lambda(\mathbf{p}), \bigcup_{i \in [t_f]} \mathcal{M}_i), \tag{2}$$

the algorithm inputs $\mathcal{M}_k$ and outputs the position, the instantaneous energy consumption, and the controls sequence–an action performed evolving the mission state.

### A. State: position and energy

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

Consider the position $\mathbf{p}$, the set

$$\mathcal{P}_k := \{\mathbf{p}_k \mid \varphi_k(\mathbf{p}_k, c_{k,1}, \ldots, c_{k,\rho}) \in \mathbb{C}_k\}, \tag{3}$$

delimits the area where the $k$-th TEE $\varphi_k : \mathbb{R}^2 \times \mathbb{R}^\rho \to \mathbb{R}$ is free to evolve using $\rho$ adjustments $\mathbf{c}_k := c_{k,1}, ..., c_{k,\rho}$ (being the TEE satisfied for all the approaching points $\varphi_k \to \mathbb{C}_k$).

The algorithm uses the concept to select the adjustments s.t. $\varphi_k(\mathbf{p}_k, \mathbf{c}_k^0)$ has the highest energy value. It guides the UAV to the new position $\mathbf{p}_{k+1}$ using the vector field of $\mathit{\Phi} := \varphi_k(\mathbf{p}_k, \mathbf{c}_k^0)$, deriving the direction to follow–the desired velocity vector

$$\dot{\mathbf{p}}_d(\mathbf{p}_k) := E\nabla\mathit{\Phi} - k_e\mathit{\Phi}\nabla\mathit{\Phi}, \;\; E = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \tag{4}$$

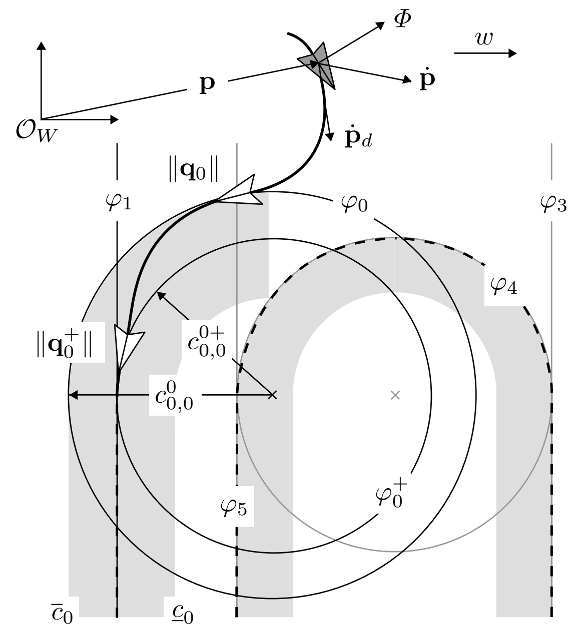


Fig. 1. Overview

where $\nabla\mathit{\Phi} \in \mathbb{R}^2$ is the gradient, $E$ 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 $w \in \mathbb{R}$.

The algorithm models the energy using a state $\mathbf{q} \in \mathbb{R}^j$ derived from Fourier analysis (the meaning of $j$ is clarified to the reader shortly) and decompose such evolution in the energy due to the trajectory, and computations–an approach adapted from our earlier work on computational energy analysis [21], [22], and energy estimation of a fixed-wing UAV [23].

### B. Energy evolution due to trajectory

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

$$h(t) = \sum_{n=0}^{r} a_n \cos(nt/\xi) + b_n \sin(nt/\xi), \tag{5}$$

where $\xi \in \mathbb{R}$ is the characteristic time, and $a_n, b_n \in \mathbb{R}$ the Fourier series coefficients.

Suppose uncertainty in the form of $\mathbf{w}_k \in \mathbb{R}^j, v_k \in \mathbb{R}$ accounting for the unknown state and output is absent. The non-linear model in Equation (5) can be expressed using an equivalent linear discrete time-invariant state-space model

$$\begin{cases} \mathbf{q}_{k+1} &= A\mathbf{q}_k + B\mathbf{u}_k + \mathbf{w}_k \\ y_k &= C\mathbf{q}_k + v_k \end{cases}, \tag{6}$$

where $y_k \in \mathbb{R}_{\geq 0}$ is the instantaneous energy consumption.

The state $\mathbf{q}$ mimics the original Fourier series coefficients

$$\mathbf{q}_k = \begin{bmatrix} \alpha_0 & \alpha_1 & \beta_1 & \cdots & \alpha_r & \beta_r \end{bmatrix}^T,$$
$$A = \begin{bmatrix} 1 & & & \\ & A_1 & & \\ & & \ddots & \\ & & & A_r \end{bmatrix}, \; A_n = \begin{bmatrix} 0 & \frac{n}{\xi} \\ -\frac{n^2}{\xi^2} & 0 \end{bmatrix}, \tag{7}$$
$$C = \begin{bmatrix} 1 & 1 & 0 & \cdots & 1 & 0 \end{bmatrix},$$

where $\mathbf{q}_k \in \mathbb{R}^j$ given $j := 2r+1$, $A \in \mathbb{R}^{j \times j}$ is the state transmission matrix, and $C \in \mathbb{R}^j$ is the output matrix. In matrix $A$, the first value is one, $A_n$ is later on the diagonal, and zero in the remainder.

The control $\mathbf{u}$ along with the input matrix

$$\mathbf{u}_k = \begin{bmatrix} g(\mathbf{s}_k) - g(\mathbf{s}_{k-1}) & \mathbf{c}_k - \mathbf{c}_{k-1} \end{bmatrix}^T,$$
$$B = \begin{bmatrix} 1 & \omega_{k,1} & \cdots & \omega_{k,\rho} \\ & 0 & & \\ & & \ddots & \\ & & & 0 \end{bmatrix}, \tag{8}$$

where $\mathbf{s}_k$ is defined in Subsection II-C, $\mathbf{u}_k \in \mathbb{R}^l$ is the control given $l := 1+\rho$, $\mathbf{u}_1 = \begin{bmatrix} 0 & \cdots & 0 \end{bmatrix}^T$ and $B \in \mathbb{R}^{j \times l}$. Moreover, the first item is one, while the others on the first row are gain factors $\omega_k \in \mathbb{R}$, quantifying the contribution of a given adjustment to the instantaneous energy consumption.

Note that the controls sequence $\mathbf{u}_k^a := (\mathbf{c}_k, \mathbf{s}_k)$–an output of the algorithm–differs from the nominal control in Equation (6) $\mathbf{u}_k(\mathbf{u}_k^a, \mathbf{u}_{k-1}^a)$.

The energy evolution analysis necessitates the following realistic assumption.

**Assumption II.1** (Energy evelution periodicity)**.** Given two time instants $k_1, k_2 \in [t_f]$ s.t. $k_1 > k_2$ and a constant value $n \in \mathbb{R}_{>0}$, there exist an arbitrary constant displacement $e \in \mathbb{R}$

$$|y_k - y_{k+n}| = e \;\; \forall k \in [k_1, k_2]. \tag{9}$$

Physically, the time evolution of the instantaneous energy consumption is assumed periodic, in the sense that it presents repetitive patterns. We show in Section IV the assumption being eased in practice to a set $\mathbb{E} \subset \mathbb{R}$, or omitted under specific conditions.

Equation (8) accounts for the energy due to the computations. The energy due to the adjustments is merely a linear combination of the gain factor and the adjustment. Nevertheless, the change updates the path which will hence affect the reading from the sensors and adjust the energy evolution. For instance, a decrement in the adjustment radius of a circle when the TEE is a circle, adds a negative contribution, thus simulates the lowering of instantaneous energy consumption.

In the case of the system behaving ideally (i.e., with no uncertainty), we expect a state (energy evolution) evolving accordingly to its output (instantaneous energy consumption). An observation summarized in the following Lemma.

**Lemma II.2** (State, output proportionality)**.** Suppose the system of Equation (6) evolves with no uncertainty ($\mathbf{w} = \mathbf{0}, v = 0$) and Assumption II.1 holds. Given two time instants $k_1, k_2 \in [t_f]$

$$\|\mathbf{q}_{k_1}\| \geq \|\mathbf{q}_{k_2}\| \iff y_{k_1} \geq y_{k_2}. \tag{10}$$

*Proof.* * ∎

*C. Energy evolution due to computations*

The energy cost of the computations is assessed using `powprofiler`, an open-source modeling tool presented in our previous work [21], that measures software configurations empirically and builds a computational energy consumption model. Specifically, the tool builds a linear interpolation, one per each task. It requires the user to implement the mission as a ROS system with one or more ROS nodes changing the computational load by node-specific ROS parameters. A way to simulate the change of the computations.

The plan $\mathcal{M}$ contains a set of ordered lists with $\sigma$ tasks each (recall Definition II.1). These tasks are simulated by $\sigma$ ROS nodes $\Psi(\mathbf{s}_k) := (\psi_1(s_{k,1}), \ldots, \psi_\sigma(s_{k,\sigma}))$, in fact they input the desired and output the actual computations

$$\mathbf{s}_k := \{s_{k,1}, \ldots, s_{k,\sigma} \mid \psi_i(s_{k,i}) \in \mathbb{S}_k \, \forall i \in [\sigma]\}, \tag{11}$$

where $s_{k,i} : \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$ returns the $i$-th computation, $\mathbf{s}_k \in \mathbb{S}_k \subseteq \mathbb{Z}_{\geq 0}^\sigma$ the set of $\sigma$ computations at time $k$ given $\mathbb{S}_k := \bigcup_{i \in [\sigma]} \mathbb{S}_{k,i}$.

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

$$y_k^s := g\left(\mathrm{QoS}_{k,0}, \ldots, \mathrm{QoS}_{k,\sigma-1}\right) = g\left(\mathbf{s}_k\right). \tag{12}$$

## III. Algorithm

Given an initial stage $\mathcal{M}_0$ (TEE $\varphi_0$, tasks $\Psi$, computations $\mathbf{s}_0 \in \mathbb{S}_0$, and adaptations $\mathbf{c}_0 \in \mathbb{C}_0$), the algorithm produce a valid mission.

**Definition III.1** (Valid mission)**.** A mission is valid if for every stage $\mathcal{M}_{k-1}$ in $k \in [t_f]^+$ there exist a control sequence $\mathbf{u}_k^a$ which produce the stage $\mathcal{M}_k$

$$\begin{aligned} \mathbf{u}_k^a = \{(\mathbf{c}_k, \mathbf{s}_k) \mid &(\varphi(\mathbf{p}_{k-1}, \mathbf{c}_{k-1}), \Psi(\mathbf{s}_{\mathbf{k-1}})) \in \mathcal{M}_{k-1} \\ &\implies (\varphi(\mathbf{p}_k, \mathbf{c}_k), \Psi(\mathbf{s}_k)) \in \mathcal{M}_k\}. \end{aligned} \tag{13}$$

Let us proof that if the mission is valid, the instantaneous energy consumption is a linear combination of the state from the Equation (6).

**Theorem III.1** (State output linearity)**.** Consider the mission from Definition II.1, the valid mission from III.1, and assume Assumption II.1 holds. Likewise in Lemma II.2, the model behaves ideally. Then, the instantaneous energy consumption $y_k$ is a linear combination of the state $\mathbf{q}_k$ and $\mathcal{M}_k$ produce a valid mission $\mathcal{M}_{k+1} \, \forall k \in [t_f - 1]$.

*Proof.* *

∎

### A. Output constraints set

Recall the model in Equation (6). We stated the output $y_k$–the instantaneous energy consumption–evolves in $\mathbb{R}_{\geq 0}$. This is generally untrue. Physical UAVs are bounded by strict battery limitations.

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

$$\mathrm{SoC}_k = -\left(V - \sqrt{V^2 - 4R\tilde{V}y_k V^{-1}}\right)/2RQ_c, \qquad (14)$$

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

$$\mathbb{Y}_k := \{y_k \mid y_k \in [0, \mathrm{SoC}_k Q_c V] \subseteq \mathbb{R}_{\geq 0}\}, \qquad (15)$$

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

### B. Deployment algorithm

1: **procedure** STEP($\mathcal{M}_{k-1}$, $\mathbf{q}_{k-1}$, $\mathbf{u}^a_{k-1}$, $\mathbf{u}^a_{k-2}$, $P_{k-1}$)
2: $\quad\mathbf{u}_{k-1} \leftarrow \mathbf{u}_{k-1}(\overline{\mathbf{u}}^a_{k-1}, \mathbf{u}^a_{k-2})$
3: $\quad\mathbf{u}^0_{k-1} \leftarrow \arg\max_{\mathbf{u}} \sum_{i=k-1}^{k+N-2} l(\mathbf{q}_i, \mathbf{u}_i) + V_f(\mathbf{q}_{k+N-1})$
4: $\quad\hat{\mathbf{q}}^-_k \leftarrow A\hat{\mathbf{q}}_{k-1} + B\mathbf{u}^0_{k-1}$
5: $\quad$**if** $C\hat{\mathbf{q}}^-_k \notin \mathbb{Y}_k$ **then**
6: $\quad\quad\mathbf{u}^a_{k-1} \leftarrow \mathbf{u}^a_{k-1}/\{\overline{\mathbf{u}}^a_{k-1}\}$
7: $\quad\quad$**return** STEP($\mathcal{M}_{k-1}$, $\mathbf{q}_{k-1}$, $\mathbf{u}^a_{k-1}$, $\mathbf{u}^a_{k-2}$,$P_{k-1}$)
8: $\quad$**else**
9: $\quad\quad$**if** $|y^m_k - C\hat{\mathbf{q}}^-_k| \leq \varepsilon$ **then**
10: $\quad\quad\quad\hat{\mathbf{q}}_k \leftarrow \hat{\mathbf{q}}^-_k$
11: $\quad\quad$**else**
12: $\quad\quad\quad P^-_k \leftarrow AP_{k-1}A^T + Q$
13: $\quad\quad\quad K \leftarrow P^-_k C^T/(CP^-_k C^T + R)$
14: $\quad\quad\quad\hat{\mathbf{q}}_k \leftarrow \hat{\mathbf{q}}^-_k + K(y^m_k - C\hat{\mathbf{q}}^-_k)$
15: $\quad\quad\quad P_k \leftarrow (I + KC)P^-_k$
16: $\quad\quad$**end if**
17: $\quad\quad\mathbf{u}^a_k \leftarrow \mathbf{u}^a_{k-1}$
18: $\quad\quad$**return** $(\hat{\mathbf{q}}_k, \mathbf{u}^a_k)$
19: $\quad$**end if**
20: **end procedure**
21: **procedure** EADMPA($\mathcal{M}_0$, $\mathbf{p}_0$, $\mathbf{q}_0$)
22: $\quad k \leftarrow 0$
23: $\quad\mathbf{u}^a_{k-1} \leftarrow \{\emptyset\}$
24: $\quad\mathbf{u}^a_k \leftarrow \{(\mathbf{c}_0, \mathbf{s}_0) \mid (\varphi_0(\mathbf{p}_0, \mathbf{c}_0), \Psi(\mathbf{s_0})) \in \mathcal{M}_0\}$
25: $\quad\mathcal{M}_k \leftarrow \mathcal{M}_0$
26: $\quad\mathbf{p}_k \leftarrow \mathbf{p}_0$
27: $\quad\mathbf{q}_k \leftarrow \mathbf{q}_0$
28: $\quad$**while** $k \leq t_f$ **do**
29: $\quad\quad(\mathbf{q}_k, \mathbf{u}^a_k) \leftarrow$STEP($\mathbf{q}_k$, $\mathbf{u}^a_k$, $\mathbf{u}^a_{k-1}$)
30: $\quad\quad\mathbf{p}_k = \mathbf{p}_k \dot{\mathbf{p}}_d(\mathbf{p}_k)/v$
31: $\quad\quad\mathbf{u}^a_{k-1} \leftarrow \mathbf{u}^a_k$
32: $\quad\quad k \leftarrow k + 1$
33: $\quad$**end while**
34: **end procedure**

Line 2 selects the maximum possible control from the control sequence at the time step $k-1$. Line 3 uses robust output feedback model predictive control (MPC) [24] to select the optimal control $\mathbf{u}^0$ for a given horizon $N$ from the cost function

$$\begin{aligned} l(\mathbf{q}_k, \mathbf{u}_k) &:= (1/2)(\mathbf{q}_k^T Q\mathbf{q}_k + \mathbf{u}_k^T R\mathbf{u}_k), \\ V_f(\mathbf{q}_k) &:= (1/2)(\mathbf{q}_k^T P_f \mathbf{q}_k) \end{aligned} \qquad (16)$$

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

Follows a check if the mission can finish without the eventuality of the battery discharge (output constraints satisfaction) at line 5. The control sequence is then eventually updated.

Before the algorithm returns the energy evolution and the control sequence, a state estimator–the discrete-time Kalman filter [25] (lines 12–15)–predicts the state $\mathbf{q}$ if the modeled instantaneous energy consumption diverging from the sensor's value $y^m_k \in \mathbb{R}$ more than a given $\varepsilon \in \mathbb{R}_{\geq 0}$.

Note the position evolution can be computed directly from Equation (4). If the velocity is $v \in \mathbb{R}_{\geq 0}$, and the starting point $\mathbf{p}_0$, $\mathbf{p}_{k+1} = \mathbf{p}_k\dot{\mathbf{p}}_d(\mathbf{p}_k)/v$.

## IV. Evaluation

*

## 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.

[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.

[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 4th IEEE International Conference on Robotic Computing (IRC)*. IEEE, 2020, p. to appear. [Online]. Available: https://adamseew.bitbucket.io/short/mechanical2020

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

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