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 in outdoor scenarios such as precision agriculture, search and rescue, and surveillance is a challenging task. It requires 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 that highlights the increasing *computational demands* and their relation to the energy consumption, path, and autonomy.

Planning algorithms for mobile robots broadly are not a new concept and have been extensively studied, 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 a narrow class of robots [4], and focus on optimizing motion control for such robots [5], despite compelling evidence for these 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 the narrow class of robots [10], [11]. Yet, computations of such systems are only expected to increase in the near future. Unlike the past approaches, the algorithm presented here attempts to combine and generalize some of these technical breakthroughs, correlate the extensively studied energy planning to the growing computational demands, and validate its findings using UAVs.

The proposed algorithm alters the energy consumption dynamically by means of mission-specific parameters: the Quality of Service (QoS) of the onboard computations, and the trajectory-explicit equations (TEEs) of the path. Our goal is a mission extension by optimizing both QoS and TEEs as the UAV flies and its batteries drain. The algorithm optimizes QoS requiring the UAV to include robot operating system (ROS) nodes. It optimizes TEEs–an abstraction of the path to follow–and guides the UAV using a vector field that converges smoothly to the path. Physically, one can use a feature detection node and a set of ellipses, with the algorithm controlling the processing rate and the length of the semi-major and -minor axis. Data using a commercial fixed-wing UAV indicates a potential extension of up to 13 minutes over an hour by merely switching to the lowest QoS.

The algorithm relies on the assumptions of the mission–a time-varying motion plan and a set of tasks the UAV is supposed to perform along the TEEs and QoS parameters constraints–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. We propose the Fourier analysis to address the periodicity assumption–being the mission periodic, we expect the energy to evolve also periodically–and a state estimator to cope with the uncertainty assumption. The algorithm selects the controls (the QoS and TEEs parameters) using a robust output feedback model predictive control (MPC).

The remainder of the paper is organized as follows. The dynamic mission planning problem is set in Section II. A suitable model for the energy and position of the UAV in Section III. The algorithm that uses the model and solves the problem is proposed in Section IV. Section V presents a fixed-wing UAV flying in an agricultural scenario showcasing the performance. The paper finishes with some conclusions in Section VI.

II. Problem Statement

We consider the problem of dynamic mission planning for a precision agriculture fixed-wing UAV. Precision agriculture

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.

is rising in demand for reducing waste, costs, and resources while increasing outputs and utilization [12], with mobile robots being increasingly used for e.g., harvesting [13]–[18]. The mission consist of a UAV flying in ellipses shifted in time, 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. Such a scenario is frequently put into practice, with complex monitoring strategies being often implemented utilizing UAVs for preventing damage and ensuring better crop quality [19].

The algorithm inputs the mission

$$\mathcal{M} := \left\{ \varphi_k,\ (\underline{c}_k, \overline{c}_k),\ \bigcup_{i=1,\ldots,\rho} (\underline{\mathrm{QoS}}_{k,i}, \overline{\mathrm{QoS}}_{k,i}) : \forall k \in \mathbb{T} \right\}, \tag{1}$$

where $\mathbb{T} := [0, t_f) \subseteq \mathbb{R}_{\geq 0}$ is the mission time, t_f the final instant, $\varphi_k \in \mathbb{R}$ is a generic continuous twice differentiable TEE along its upper and lower bounds $(\underline{c}_k, \overline{c}_k) \in \mathbb{R}^2$ (the motion plan), and a set of upper and lower bounds for the σ tasks $(\underline{\mathrm{QoS}}_{k,i}, \overline{\mathrm{QoS}}_{k,i}) \in \mathbb{Z}^2_{\geq 0}$.

It outputs the correction to the position and the instantaneous energy consumption, while it adapts the controls $\mathbf{u}_k := (c_k, \mathrm{QoS}_{k,1}, \ldots, \mathrm{QoS}_{k,\rho})$, the TEE and QoS parameters.

III. Model

The state is the UAV's position in space and the energy evolution in time. Despite we show a linear relation between instantaneous energy and energy evolution, 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.

Starting with a position $\mathbf{p} \in \mathbb{R}^3$ in the 3D Euclidean space w.r.t. some inertial navigation frame $\mathcal{O}_W$, we build a guidance action–which allows the *position evolution* along the 3-axis–to the path to follow using a vector field [20]. We build the path with the TEE, being such function satisfied $\varphi(\mathbf{p}) \to 0$ for all the approaching points.

We build the *energy evolution* $\mathbf{q} \in \mathbb{R}^j$ using Fourier analysis (the meaning of j is clarified to the reader in Subsection III-B) and decompose such evolution in the energy due to the trajectory (depending on TEEs), and computations (on QoS)–an approach adapted from our earlier work on computational energy analysis [21], [22], and energy estimation of a fixed-wing UAV [23].

A. Position evolution

We consider a non-holonomic 2D model of the UAV flying at an assigned altitude $h \in \mathbb{R}_{>0}$ (the requirement is eased in Section IV to a set of altitudes $\mathbf{h}$)

$$\begin{cases} \dot{\mathbf{p}}(t) &= s\Psi(\psi(t)) + \mathbf{d}(t) \\ \dot{\psi}(t) &= u(\mathbf{p}(t)) \end{cases}, \tag{2}$$

where $\mathbf{p} \in \mathbb{R}^2$, $s \in \mathbb{R}$ is the airspeed (assumed constant), $\psi \in (-\pi, \pi]$ the attitude yaw angle and $\Psi(\psi) = \begin{bmatrix} \cos\psi & \sin\psi \end{bmatrix}^T$, $\mathbf{d} \in \mathbb{R}^2$ is the wind vector (assumed $\|\mathbf{d}\| < s$, i.e., the constant airspeed is greater than the norm of the wind), and $u \in \mathbb{R}$ the guidance action–the angular velocity $\dot{\psi}$.

Let us define $\mathcal{P}_k$, the set that delimits all the possible deviations form the k-th TEE

$$\mathcal{P}_k := \{\mathbf{p} : \underline{c}_k \leq \varphi_k(\mathbf{p}, c_{k,1}, \ldots, c_{k,\rho}) \leq \overline{c}_k\}, \tag{3}$$

where $\underline{c}_k, \overline{c}_k$ are the bounds where the TEE $\varphi_k : \mathbb{R}^2 \times \mathbb{R}^\sigma \to \mathbb{R}$ is free to evolve using controls $\mathbf{c}_k := c_{k,1}, ..., c_{k,\sigma}$.

The algorithm uses the concept to select $\varphi(\mathbf{p}, \mathbf{c}_k) = c_k \in \mathbb{R}$ s.t. $\mathbf{p} \in \mathcal{P}_k$ with the highest energy value under the energy budget constraints. We design a guidance action u that heads to such TEE by minimizing the norm $\|\varphi_k(\mathbf{p}) - c_k\|$.

Let us define $\mathit{\Phi} := \varphi_k(\mathbf{p}) - c_k$. The direction to follow can be expressed as the desired velocity vector

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

where $\nabla\mathit{\Phi} \in \mathbb{R}^2$ is defined as the gradient of $\mathit{\Phi}$ at the point $\mathbf{p}$ (i.e., its vector field), 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 $\chi \in (-\pi, \pi]$ due to the atmospheric interference.

Let us further define $\hat{\mathbf{p}} := \mathbf{p}/\|\mathbf{p}\|$, the desired course heading rate $\dot{\chi}_d$ is computed by sensing the position $\mathbf{p}$, the ground velocity $\dot{\mathbf{p}}$, and is expressed

$$\begin{aligned} \dot{\chi}_d(\mathbf{p}) = -E\frac{\dot{\mathbf{p}}_d}{\|\dot{\mathbf{p}}_d\|^2} \cdot \\ \left(E\hat{\dot{\mathbf{p}}}_d\hat{\dot{\mathbf{p}}}_d^T E \left((E - k_e\mathit{\Phi})H(\mathit{\Phi})\dot{\mathbf{p}} - k_e\nabla\mathit{\Phi}^T\dot{\mathbf{p}}\nabla\mathit{\Phi} \right) \right)^T, \end{aligned} \tag{5}$$

where $H(\cdot)$ is defined as the Hessian operator and $\dot{\mathbf{p}}_d := \dot{\mathbf{p}}_d(\mathbf{p})$ for brevity; the physical meaning is that the curvature of the desired trajectory has to be known in order to be tracked.

The guidance action can be expressed

$$u(\mathbf{p}, \psi) = \frac{\|\dot{\mathbf{p}}\|}{s\cos\gamma} \left(\dot{\chi}_d + k_d\hat{\dot{\mathbf{p}}}^T E\hat{\dot{\mathbf{p}}}_d \right), \tag{6}$$

where $\gamma = \cos^{-1}\left(\hat{\dot{\mathbf{p}}}^T \Psi(\psi) \right)$ is the sideslip angle, $k_d \in \mathbb{R}_{\geq 0}$ adjusts the speed of convergence of $\dot{\mathbf{p}}_d$, and $\dot{\chi}_d := \dot{\chi}_d(\mathbf{p})$ is given in Equation (5).

—from here on work in progress—

B. Energy evolution due to trajectory

The energy evolution analysis necessitates the following realistic assumption.

Assumption 3.1: The position evolution is periodic, in the sense that it presents repetitive patterns.
Formally, given two time instants $k_1, k_2 \in \mathbb{R}_{\geq 0}$ such that $k_1 > k_2$ and a value $n \in \mathbb{R}_{\geq 0}$, there exist an arbitrary constant displacement $e \in \mathbb{R}$ such that

$$\|\mathbf{p}(t) - \mathbf{p}(t+n)\| = e\ \ \forall t \in \{k_1, ..., k_2\}. \tag{7}$$

We show in Section V the assumption being eased in practice to a set $\mathbb{E} \subset \mathbb{R}$.

Let us consider a Fourier series $f : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$ of an arbitrary order $r \in \mathbb{Z}_{\geq 0}$

$$f(t) = \sum_{n=0}^{r} a_n \cos \frac{nt}{\xi} + b_n \sin \frac{nt}{\xi}, \tag{8}$$

where $\xi \in \mathbb{R}$ is the characteristic time, and $a_n, b_n \in \mathbb{R}$ for $n \in \{0, \ldots, r\}$ the Fourier series coefficients.

The non-linear model in Equation (8) can be expressed using an equivalent linear time-invariant state-space model

$$\begin{cases} \dot{\mathbf{q}}(t) & = A\mathbf{q}(t) + B\mathbf{u}(t) \\ y(t) & = C\mathbf{q}(t) \end{cases}, \tag{9}$$

where $y(t) \in \mathbb{R}_{\geq 0}$ is the instantaneous energy consumption. We prove formally in the Theorem 4.1 the instantaneous energy being obtained as a linear combination of the state. The control $\mathbf{u}$ along with the input matrix B are defined later in Subsection III-D, the state $\mathbf{q}$ mimics the original Fourier series coefficients, and

$$\begin{aligned} \mathbf{q}(t) &= \left[\begin{array}{cccccc} \alpha_0 & \alpha_1 & \beta_1 & \cdots & \alpha_r & \beta_r \end{array} \right]^T, \\ A &= \left[\begin{array}{cccc} 1 & & & \\ & A_1 & & \\ & & \ddots & \\ & & & A_r \end{array} \right], \ A_n = \left[\begin{array}{cc} 0 & \frac{n}{\xi} \\ -\frac{n^2}{\xi^2} & 0 \end{array} \right], \\ C &= \left[\begin{array}{cccccc} 1 & 1 & 0 & \cdots & 1 & 0 \end{array} \right], \end{aligned} \tag{10}$$

where $\mathbf{q}(t) \in \mathbb{R}^j$ 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 remaining.

We further model the system of Equation (9) plus disturbances with the following discrete time model

$$\begin{cases} \mathbf{q}_{k+1} & = A\mathbf{q}_k + B\mathbf{u}_k + \mathbf{w}_k \\ y_k & = C\mathbf{q}_k + v_k \end{cases}, \tag{11}$$

where $\mathbf{w}_k \in \mathbb{R}^j, v_k \in \mathbb{R}$ account for unknown state and output disturbances. In the case of the system behaving ideally (i.e., with zero disturbances), we expect a set of states evolving accordingly to their output. Such observation is summarized in the following Lemma.

Lemma 3.2: Suppose the system of Equation (11) evolves with no disturbances ($\mathbf{w} = \mathbf{0}, v = 0$) and Assumption 3.1 holds, then

$$\|\mathbf{q}_{k,0}\| \geq \|\mathbf{q}_{k,1}\| \iff y_{k,0} \geq y_{k,1}. \tag{12}$$

Proof:

"*The easy proof is trivial and is left as an exercise to the reader :P*"

■

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 an energy model. Specifically, the tool builds a multivariate linear interpolation from the mission specification. It hence requires the user to implement a ROS system with one or more computationally expensive ROS nodes which change the computational load by node-specific ROS parameters (QoS).

Suppose the system is composed of σ computationally expensive ROS nodes. Let us define the computational control action

$$\mathcal{C}_k := \{u : u \in \mathrm{QoS}_n(k) \ \forall n \in \{0, \ldots, \sigma - 1\}\}, \tag{13}$$

where $\mathrm{QoS}_n(k) : \mathbb{Z}_{\geq 0} \rightarrow \mathbb{Z}_{\geq 0}$ returns the n-th QoS value at time instant k, and $\mathcal{C}_k \in \mathbb{Z}^{\sigma}_{\geq 0}$ the set of σ QoS values the system is composed of. Let us further define $g : \mathbb{Z}_{\geq 0} \times \mathbb{Z}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ as the instantaneous energy value obtained interrogating `powprofiler`. The instantaneous computational energy component can be defined

$$y^c_k : = g\left(\mathrm{QoS}_0\left(k\right), \ldots, \mathrm{QoS}_{\sigma-1}\left(k\right)\right) = g\left(\mathcal{C}_k\right). \tag{14}$$

The QoS parameters $\mathcal{C}_k$ can be subject to different constraints at different states. Physically, this means that the UAV can perform the ROS nodes within different QoS ranges in time while flying

$$\begin{aligned} \underline{\mathrm{QoS}}_n(k) \leq \mathrm{QoS}_n(k) \leq \overline{\mathrm{QoS}}_n(k), \\ \forall n \in \{0, \cdots, \sigma - 1\}, \end{aligned} \tag{15}$$

where the values $\underline{\mathrm{QoS}}_n(k), \overline{\mathrm{QoS}}_n(k)$ are retrieved from the mission specification.

—I'll move this subsec under eq 9, it doesn't make sense now and my brain stopped working for this evening—

D. Control action

We build the control action $\mathbf{u}$ from Equation (9) and (11) in two steps. In Equation (13) we defined the control due to the computations, here we do the same for the trajectory.

Let us re-define the TEE introduced first in Subsection III as a function $\varphi_k : \mathbb{R}^3 \times \mathbb{R}^\rho \rightarrow \mathbb{R}$ of the position and ρ TEE parameters $\mathcal{M} \in \mathbb{R}^\rho$, e.g., the constants of a linear function, the radius of a circle, and semi-major and minor axis of an ellipse. Given $\mathbf{u}_k := \{u_{k,0}, \ldots, u_{k,\rho-1}\}$, the set of these parameters at time instant k can be expressed

$$\mathcal{M}_k := \{u_k \in \mathbf{u}_k : \varphi_k(\mathbf{p}, \mathbf{u}_k) \in \mathcal{P}_k\}, \tag{16}$$

where $\mathbf{p} \in \mathcal{P}_k$ is any optimal point which let the trajectory explicit function φ_k converge under any optimal TEE parameter u^0_k. Physically, $\mathcal{M}_k$ contains all the controls generating the points over the area $\mathcal{P}$ defined in Equation (3).

The explicit trajectory equation φ_k can be different at different states k, meaning the vector field and guidance action, from Equation (4) and (6) respectively, will account for the sudden change of trajectory during the mission.

It is worth considering that the number of parameters at state k is a parameter of the state. This is for the sake of generality, as the mission specification might contain different explicit equations for different states. For instance, the fixed-wing craft might follow an ellipse function throughout the mission and heading a linear function while landing.

The TEE parameters and QoS, $\mathcal{M}$ and $\mathcal{C}$ defined in Equation (16) and (13), are incorporated in the system in Equation (11) using the input matrix

$$\mathbf{u}_k = \left[\frac{g\left(\mathcal{C}_k\right) - g\left(\mathcal{C}_{k-1}\right)}{\mathcal{M}_k} \right], \quad B = \left[\begin{array}{c|ccc} 1 & 0 & \cdots & 0 \\ \hline 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{array} \right], \tag{17}$$

where $\mathbf{u}_k \in \mathbb{R}^l$ is the control given $l := 1+\rho$, and $B \in \mathbb{R}^{j \times l}$ is the input matrix from Equation (9) and (11). Moreover, the first column in the first row of the input matrix is 1, while all the other items are 0. This adds the computational model component to the energy evolution in the system. The energy due to the change of explicit trajectory equation parameters is not directly added to the system, which however will update the reading from the sensors y^s defined in Subsection IV-A and adjust the energy evolution accordingly.

IV. Algorithm

The main goal of the algorithm is to find an optimal control action $\mathbf{u}^0$ for the current state $\hat{\mathbf{q}}$–estimated from sensors' measurements–from the optimal control law $\mathbf{u}^0 =: \kappa(\hat{\mathbf{q}})$. This is achieved by solving online a finite horizon optimal control problem by the hand of a modification of a model predictive control (MPC) algorithm [24].

Let us proof formally an important finding from Section III extensively used in the algorithm.

Theorem 4.1: Consider a continuously differentiable function $\varphi_k : \mathbb{R}^3 \to \mathbb{R}$ at a time instant $k \in \mathbb{Z}_{>0}$. Assume Assumption 3.1 holds, the robots is free to move in $\mathcal{P}$ defined in (3), and is following φ with the direction $\hat{\mathbf{p}}_d$ defined in 4. Likewise in Lemma 3.2, the model behaves ideally. Then, the instantaneous energy consumption is a linear combination of the state

$$y_k = C\mathbf{q}_k = \sum_{n=0}^{r} \alpha_n, \tag{18}$$

where $\alpha_n \in \mathbf{q}_k$ are the $r+1$ state's components at k with r being a preassigned arbitrary order from (8), and C is described in Equation (10).

Proof: *

■

A. State estimation

As the environment uncertainty and measurement error evolve in a normal distribution, we use a Kalman filter [25], [26] for the purpose of state estimation.

The prediction is done using

$$\hat{\mathbf{q}}^-_{k+1} = A\hat{\mathbf{q}}_k + B\mathbf{u}_k, \tag{19a}$$

$$P^-_{k+1} = AP_kA^T + Q, \tag{19b}$$

where $\hat{\mathbf{q}}^-_k, \hat{\mathbf{q}}_k \in \mathbb{R}^j$ depicts the estimate of the state before and after measurement (or simply estimate), and $P_k, P^-_k \in \mathbb{R}^{j\times j}$ the error covariance matrix (i.e., the variance of the estimate).

The estimation of the state and the update of the predicted output is done using

$$K_k = (CP^-_{k+1}C^T + R)^{-1}(P^-_{k+1}C^T), \tag{20a}$$

$$\hat{\mathbf{q}}_{k+1} = \hat{\mathbf{q}}^-_{k+1} + K_k(y^s_k + y^c_k - C\hat{\mathbf{q}}^-_{k+1}), \tag{20b}$$

$$P_{k+1} = (I - K_kC)P^-_{k+1}, \tag{20c}$$

$$\hat{y}_k = C\hat{\mathbf{q}}_{k+1}, \tag{20d}$$

where $K_k \in \mathbb{R}^j$ is the gain of the Kalman filter, and I the identity matrix. y^s_k, y^c_k are the instantaneous energy readings: $y^s_k \in \mathbb{R}_{\geq 0}$ the robot sensor, i.e., the energy due to the trajectory, and y^c_k the energy of a given software configuration described in Equation (14). The noise covariance matrices $Q \in \mathbb{R}^{j\times j}, R \in \mathbb{R}$ indicates the uncertainty and measurement error covariance respectively, and $\hat{y}_k \in \mathbb{R}_{\geq 0}$ is the estimated energy.

Equations (19–20) converge to the predicted energy evolution as follows. An initial guess of the values $\hat{\mathbf{q}}_0, P_0$ is derived empirically from collected data. It is worth considering that an appropriate guess of these parameters allows the algorithm to converge to the desired energy evolution in a shorter amount of time. The tuning parameters Q, R are also derived from the collected data, and may differ due to i.e., different sensors used to measure the instantaneous energy consumption, or different atmospheric conditions accounting for the process noise.

At time $k = 0$, the initial estimate before measurement of the state and of the error covariance matrix is updated in Equation (19a) and (19b) respectively. The value of $\hat{\mathbf{q}}^-_1$ is then used in Equation (20b) to estimate the current state along with the data from the sensor y_0 (e.g., the energy sensor of the flight controller of the fixed-wing craft), where the sensor noise covariance matrix R accounts for the amount of uncertainty in the measurement. The estimated output $\hat{y}_0$ is then obtained from Equation (20d). The algorithm is iterative. At time $k = 1$ the values $\hat{\mathbf{q}}_1, P_1$ computed at previous step are used to estimate the values $\hat{\mathbf{q}}_2, P_2$, and y_1.

B. Optimal control action

*

C. Deployment algorithm

*

V. Evaluation

*

VI. 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] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[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] R. F. Stengel, *Optimal control and estimation*. Courier Corporation, 1994.

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