***

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

Precision agriculture is rising in demand for reducing waste, costs, and resources while increasing outputs and utilization [1]. Researchers have fulfilled some of these demands applying mobile robots in agricultural scenarios such as harvesting [2]–[7], where complex monitoring strategies–often implemented utilizing unmanned aerial vehicles (UAVs)–are usually required for preventing damage and ensuring better crop quality [8]. Planning a mission for UAVs operating in such scenarios is a challenging task. It usually requires advanced levels of autonomy, with the UAV being, e.g., used to inform its grounded counterparts of obstacles detected while flying, along with strictly limited energy budgets. Currently, UAVs in precision agriculture are semi-autonomous, in the sense that the path is static and defined using a mission planning software [9]. The presented approach proposes an *energy-aware dynamic mission planning algorithm* for UAVs for precision agriculture that addresses the broader problem of increasing *computational demands* and their relation to the UAV's energy consumption, path, and autonomy.

Although underrepresented in precision agriculture literature, energy planning algorithms for mobile robots are not a new concept and have been extensively studied, being these correlated to such topics as trajectory generation and path planning. Generally, they account for the energy due to the trajectory, by e.g., maximizing the operation time [10], but practically restrict to a narrow class of mobile robots [11], and focus on optimizing motion control for such robots [12]. In a similar setting, rotorcrafts have been an obvious object of research interest, with such approaches as the energy-efficient trajectory generation [13], [14]. Conversely, the presented approach attempts to combine and generalize some of these technical breakthroughs and correlate the extensively studied energy planning to the growing computational demands of precision agriculture UAVs through a mission-aware perspective. Of particular inspiration to the approach is the technique of coupling the energy due to the computations performed and trajectory traveled, although being, likewise the others, limited to the trajectory [15]–[17], or to the class of mobile robots [18], [19].

The algorithm adapts the energy dynamically, replanning mission-specific parameters: the Quality of Service (QoS) of the onboard computations, and the trajectory-explicit equations (TEEs) of the path. It ideally accounts for a mission extension by adapting both QoS and TEEs as the UAV flyes and its batteries drain. Firstly, the algorithm adapts QoS requiring the UAV to comprise robot operating system (ROS) with computationally expensive ROS nodes controlled through ROS parameters. Secondly, it adapts TEEs–a mathematical abstraction of the path to follow–and implements the change updating a vector field that converges smoothly to such 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 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 path and a set of tasks the UAV is supposed to perform along the TEEs and QoS parameters boundaries–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 (QoS and TEEs) dynamically using a modified tube-based output model predictive control (MPC), ensuring that state estimation error does not result in transgression of the control and state constraints.

The remainder of the paper is organized as follows. A suitable energy model for the path and the computations is showed in Section II. The algorithm that uses the energy model to address dynamic mission planning is proposed in Section III. Section IV presents a fixed-wing UAV flying

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.

in an agricultural scenario showcasing the performance. The paper finishes with some conclusions in Section V.

II. Model

The algorithm utilizes the model to estimate the robot's position in space and its energy evolution in time.

Starting with a position $\mathbf{p} \in \mathbb{R}^3$ in the 3D Euclidean space, with respect to some inertial navigation frame $\mathcal{O}_W$, a guidance action–which allows the motion along the 3-axis–is built upon the direction to follow and derived using vector field [20]. To define the direction to follow, we use generic continuously differentiable functions TEEs $\varphi : \mathbb{R}^3 \to \mathbb{R}$, a mathematical abstraction representing the desired trajectory (being the function satisfied $\varphi(\mathbf{p}) \to 0$ for all the points approaching such trajectory).

The energy evolution $\mathbf{q} \in \mathbb{R}^j$ spent to perform such guidance action is derived using Fourier analysis (the meaning of j will be explained to the reader in Subsection II-B) and is decomposed in the energy due to the trajectory (TEEs), and the computations (QoS)–an approach that has been adapted from authors' earlier work on computational energy analysis [21], [22], and energy estimation of a fixed-wing craft [23].

A. Position in space and guidance action

For exemplification, we consider a non-holonomic 2D model of a fixed-wing craft flying at an assigned altitude $h \in \mathbb{R}_{>0}$. We will show, after the results, the requirement being eased to an arbitrary mobile robot model

$$\begin{cases} \dot{\mathbf{p}}(t) &= s\Psi(\psi(t)) + d(t) \\ \dot{\psi}(t) &= u(\mathbf{p}(t)) \end{cases}, \tag{1}$$

where $\mathbf{p}(t) \in \mathbb{R}^2$, $s \in \mathbb{R}$ describes the airspeed assumed constant, $\psi(t) \in (-\pi, \pi]$ the attitude yaw angle and $\Psi(\psi(t)) = \begin{bmatrix} \cos\psi(t) & \sin\psi(t) \end{bmatrix}^T$, d is the wind vector where we assume $\|d\| < s$, and $u \in \mathbb{R}$ the guidance action which denotes the angular velocity $\dot{\psi}$ of the craft.

Let us define $\mathcal{P}$ the area which discloses all the possible paths within the boundaries $\underline{c} \in \mathbb{R}_{\leq 0}, \overline{c} \in \mathbb{R}_{\geq 0}$

$$\mathcal{P} := \{\mathbf{p} : \underline{c} \leq \varphi(\mathbf{p}) \leq \overline{c}\}, \tag{2}$$

where $\underline{a}, \overline{a}$ returns the upper bound and lower bound limit of a from the mission specification, which is a lookup table.

The concept of area between the bounds is used later to design a controller that selects c with the highest energy value under the energy budget constraints. Here we design a guidance action u that allows following φ in such area by the means of minimizing the norm $\|\varphi(\mathbf{p})\|$.

Let us define $\Phi := \varphi(\mathbf{p})$. The direction to follow is expressed as the desired velocity vector

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

where $\nabla\Phi \in \mathbb{R}^2$ is defined as the gradient of φ 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 which adjusts the speed of convergence.

The direction the velocity vector $\dot{\mathbf{p}}$ 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

$$\begin{aligned} \dot{\chi}_d(\mathbf{p}) = -E\frac{\dot{\mathbf{p}}_d}{\|\dot{\mathbf{p}}_d\|^2} \cdot \\ \left(E\hat{\dot{\mathbf{p}}}_d\hat{\dot{\mathbf{p}}}_d^T E\left((E - k_e\Phi)H(\Phi)\dot{\mathbf{p}} - k_e\nabla\Phi^T\dot{\mathbf{p}}\nabla\Phi\right)\right)^T, \end{aligned} \tag{4}$$

where $H(\cdot)$ is defined as the Hessian operator; the physical meaning is that the curvature of the desired trajectory has to be known in order to be tracked.

Under the assumption of the airspeed $s > \|w(t)\|$ for $t > 0$ (i.e., the constant airspeed is greater than the norm of the wind), the guidance action can be expressed

$$u(\mathbf{p}, \psi) = \frac{\|\dot{\mathbf{p}}\|}{s\cos\gamma}\left(\dot{\chi}_d(\mathbf{p}) + k_d\hat{\dot{\mathbf{p}}}^T E\hat{\dot{\mathbf{p}}}_d\right), \tag{5}$$

where $k_d \in \mathbb{R}_{\geq 0}$ is the gain which adjusts the speed of the convergence of $\dot{\mathbf{p}}_d$, $\gamma = \cos^{-1}\left(\hat{\dot{\mathbf{p}}}^T\Psi(\psi)\right)$ is the sideslip angle, and $\dot{\chi}_d$ is given in Equation (4).

B. Energy evolution due to trajectory

To evaluate the energy we formalize the following realistic assumption introduced in Section I.

Assumption 2.1: The mission the mobile robots perform is periodic, in the sense that it presents repetitive patterns.

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

$$f(t) = \sum_{n=0}^{r} a_n\cos\frac{nt}{\xi} + b_n\sin\frac{nt}{\xi}, \tag{6}$$

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 (6) can be expressed using an equivalent time-varying state-space model in the following form

$$\begin{cases} \dot{\mathbf{q}}(t) &= A\mathbf{q}(t) + B\mathbf{u}(t) \\ y(t) &= C\mathbf{q}(t) \end{cases}, \tag{7}$$

where $y(t) \in \mathbb{R}_{\geq 0}$ is the instantaneous energy evolution of the system being controlled. We will see later in the Theorem 3.1 that the instantaneous energy can be obtained from the model as a linear combination of the state. The control $\mathbf{u}$ along with the input matrix B are defined later in Subsection II-D, the state $\mathbf{q}$ mimics the original Fourier series coefficients, and

$$\begin{aligned} \mathbf{q}(t) &= \left[\begin{array}{c|ccccc} \alpha_0 & \alpha_1 & \beta_1 & \cdots & \alpha_r & \beta_r \end{array}\right]^T, \\ A &= \left[\begin{array}{c|ccc} 1 & \mathbf{0} & \cdots & \mathbf{0} \\ \hline \mathbf{0} & A_1 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & A_r \end{array}\right], \; A_n = \begin{bmatrix} 0 & \frac{n}{\xi} \\ -\frac{n^2}{\xi^2} & 0 \end{bmatrix}, \\ C &= \left[\begin{array}{c|ccccc} 1 & 1 & 0 & \cdots & 1 & 0 \end{array}\right], \end{aligned} \tag{8}$$

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. Furthermore, the first row and column of the matrix A contain zeros row vectors and column vectors respectively.

Motivated by the fact that the system of interest is sampled discrete-time, we consider the discretized version of Equation (7) and add the uncertainty

$$\begin{cases} \mathbf{q}_{k+1} & = A\mathbf{q}_k + B\mathbf{u}_k + w_k \\ y_k & = C\mathbf{q}_k + v_k \end{cases}, \tag{9}$$

where $w_k \in \mathbb{R}$ accounts for the environment uncertainty, and $v_k \in \mathbb{R}$ for the measurement error.

The instantaneous energy can be obtained from the model (9) as a linear combination of the state.

Lemma 2.2: Suppose the system behaves ideally (with no environment uncertainty and measurement error). The magnitude of two arbitrary states $\mathbf{q}_{k,0}, \mathbf{q}_{k,1}$ described in (8) along their evolution (9) at time instant k depends on the instantaneous energy consumption

$$\|\mathbf{q}_{k,0}\| \geq \|\mathbf{q}_{k,1}\| \iff y_{k,0} \geq y_{k,1}. \tag{10}$$

Proof: *

■

C. Energy evolution due to computations

A computational energy model is built using `powprofiler`, an open-source modeling tool that measures empirically software configurations and builds an energy model, presented in authors' previous work [21]. Specifically, the tool builds a multivariate linear interpolation which is accessed online at the hand of a lookup table in the optimal control algorithm. The system is modeled as follows. An existing ROS system composes several computationally expensive ROS nodes, allowing to vary the number of computations changing some node-specific quality of service (QoS) values via ROS parameters. The tool builds the energy model using mission specification which, besides other mission parameters, specifies per each ROS node a QoS range.

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

$$\mathcal{C}_k := \{u : u \in \mathrm{QoS}_n(k) \, \forall n \in \{0, \ldots, \sigma - 1\}\}, \tag{11}$$

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

$$y^c_k := g\left(\mathrm{QoS}_0(k), \ldots, \mathrm{QoS}_{\sigma-1}(k)\right) = g\left(\mathcal{C}_k\right). \tag{12}$$

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

$$\begin{aligned} \underline{\mathrm{QoS}}_n(k) \leq \mathrm{QoS}_n(k) \leq \overline{\mathrm{QoS}}_n(k), \\ \forall n \in \{0, \cdots, \sigma - 1\}, \end{aligned} \tag{13}$$

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

The control action is built in two steps. Equation (11) defines the control due to the computations. The control due to the trajectory is specific to the system being analyzed, for instance on the model of the fixed-wing craft in (1), along its control in Equation (4) and (5). A generalization which accounts for the variations in the TEE φ of a generic model by means of the TEEs parameters is derived in the following subsection.

D. Control action

Given a generic trajectory equation φ, the trajectory can be modeled by 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. Let us define $\mathbf{u}_{k,n} := \{u_{k,0}, \ldots, u_{k,n-1}, u^0_{k,n}, u_{k,n+1}, \ldots, u_{k,\rho_k-1}\}$, the set of these parameters can be expressed

$$\begin{aligned} \mathcal{M}_k := \{u_{k,n} : \varphi_k(\mathbf{p}^0_{k,n}, \mathbf{u}_{k,n}) \in \mathcal{P} \\ \forall n \in \{0, \ldots, \rho_k - 1\}\}, \end{aligned} \tag{14}$$

where $\mathbf{p}^0_k$ 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 (2).

The explicit trajectory equation φ_k can be different at different states k, meaning the vector field and guidance action, from Equation (3) and (5) 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 (14) and (11), are incorporated in the system in Equation (9) using the input matrix

$$\mathbf{u}_k = \left[\frac{g\left(\mathcal{C}_k\right)}{\mathcal{M}_k}\right], \; B = \left[\begin{array}{c|ccc} 1 & 0 & \cdots & 0 \\ \hline 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{array}\right], \tag{15}$$

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 (7) and (9). 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 III-A and adjust the energy evolution accordingly.

III. 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 II extensively used in the algorithm.

Theorem 3.1: Consider a continuously differentiable function $\varphi_k : \mathbb{R}^3 \to \mathbb{R}$ at a time instant $k \in \mathbb{Z}_{>0}$. Assume Assumption 2.1 holds, the robots is free to move in $\mathcal{P}$ defined in (2), and is following φ with the direction $\hat{\mathbf{p}}_d$ defined in 3. Likewise in Lemma 2.2, the model behaves ideally. Then, the instantaneous energy consumption is a linear combination of the state

$$y_k = C\mathbf{q}_k = \sum_{n=0}^{r} \alpha_n, \tag{16}$$

where $\alpha_n \in \mathbf{q}_k$ are the $r+1$ state's components at k with r being a preassigned arbitrary order from (6), and C is described in Equation (8).

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

$$\hat{\mathbf{q}}_{k+1}^- = A\hat{\mathbf{q}}_k + B\mathbf{u}_k, \tag{17a}$$

$$P_{k+1}^- = AP_kA^T + Q, \tag{17b}$$

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

$$K_k = (CP_{k+1}^-C^T + R)^{-1}(P_{k+1}^-C^T), \tag{18a}$$

$$\hat{\mathbf{q}}_{k+1} = \hat{\mathbf{q}}_{k+1}^- + K_k(y_k^s + y_k^c - C\hat{\mathbf{q}}_{k+1}^-), \tag{18b}$$

$$P_{k+1} = (I - K_kC)P_{k+1}^-, \tag{18c}$$

$$\hat{y}_k = C\hat{\mathbf{q}}_{k+1}, \tag{18d}$$

where $K_k \in \mathbb{R}^j$ is the gain of the Kalman filter, and I the identity matrix. y_k^s, y_k^c are the instantaneous energy readings: $y_k^s \in \mathbb{R}_{\geq 0}$ the robot sensor, i.e., the energy due to the trajectory, and y_k^c the energy of a given software configuration described in Equation (12). 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 (17–18) 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 (17a) and (17b) respectively. The value of $\hat{\mathbf{q}}_1^-$ is then used in Equation (18b) 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 (18d). 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

*

IV. Evaluation

*

V. Conclusion and Future Work

*

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] ——, "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.

[17] ——, "Deployment of mobile robots with energy and timing constraints," *IEEE Transactions on robotics*, vol. 22, no. 3, pp. 507–522, 2006.

[18] 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.

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

[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.