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 robot planning, 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) values of the onboard computations, and the trajectory-explicit equations (TEEs) adjustments of the motion plan. 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 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 selects the controls (QoS and TEEs parameters) using robust output feedback model predictive control (MPC).

In the spirit of reducing waste, 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 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 monitoring mission optimized for the craft's dynamics. The algorithm plans the mission controlling the processing rate and the length of the semi-major and -minor axis. Data indicates a potential extension of up to 13 minutes over an hour by merely switching to the lowest QoS.

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 energy and position. The algorithm that uses the model and solves the dynamic mission planning problem is proposed in Section III. Section IV presents the result and showcase the performance. The paper finishes with some conclusions in Section V.

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.

II. Mission Planning Overview

Let the mission be defined by a generic continuous twice differentiable TEE φ (being such function satisfied $\varphi \to 0$ for all the approaching points) and a set of σ tasks. Moreover, let t_f be the mission final time, $[\cdot]$ the set $\{1, 2, \ldots, \cdot\}$, and $\underline{\cdot}, \overline{\cdot}$ the upper and lower bound of $\cdot$. At a time instant $k \in [0, t_f) \subseteq \mathbb{R}_{\geq 0}$, the mission is formally defined as the ordered list

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

where $\mathbb{C}_k := [\underline{c}_k, \overline{c}_k] \subseteq \mathbb{R}$ is a formulation for the TEE φ adjustments constraints, and $\mathbb{S}_{k,i} := [\underline{\mathrm{QoS}}_{k,i}, \overline{\mathrm{QoS}}_{k,i}] \subseteq \mathbb{Z}_{\geq 0}$ for the i-th task QoS value constraints.

The overall plan is the union of all the missions: a set of TEEs, tasks, and TEEs and QoS parameters. If the system is sampled discrete-time for simplicity

$$\mathcal{M} := \bigcup_{i \in [tf]} \mathcal{M}_i, \tag{2}$$

the algorithm inputs $\mathcal{M}$ and outputs the position and the instantaneous energy consumption, while it adapts the control in function of the TEE and QoS parameters–an action performed evolving the mission state.

A. Mission state

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

Consider the position $\mathbf{p} \in \mathbb{R}^2$ of a UAV flying at an assigned altitude $h \in \mathbb{R}_{>0}$ w.r.t. some inertial navigation frame $\mathcal{O}_W$, the set

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

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 ρ controls $\mathbf{c}_k := c_{k,1}, ..., c_{k,\rho}$.

The algorithm uses the concept to select $\varphi_k(\mathbf{p}, \mathbf{c}_k^0)$ with the highest energy value under the constraints, and *evolves the position* using the vector field of $\mathit{\Phi} := \varphi_k(\mathbf{p}, \mathbf{c}_k^0)$ and deriving the direction to follow. It 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 at the point $\mathbf{p}$, 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.

The algorithm *evolves the energy* using a state $\mathbf{q} \in \mathbb{R}^j$ derived from Fourier analysis (the meaning of j is clarified to the reader in the next subsection) 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 $f : \mathbb{R}_{\geq 0} \to \mathbb{R}$ of an arbitrary order $r \in \mathbb{Z}_{\geq 0}$ for the purpose of energy estimation of the mission

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

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.

Suppose no disturbances in form of $\mathbf{w}_k \in \mathbb{R}^j, v_k \in \mathbb{R}$ accounting otherwise for the unknown state and output. 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. We prove formally in the Theorem 3.1 the instantaneous energy being obtained as a linear combination of the state. 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 remaining.

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

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

where $\mathcal{C}_k$ 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 TEE parameter to the instantaneous energy.

The energy evolution analysis necessitates the following realistic assumption.

Assumption 2.1: 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 change of explicit trajectory equation parameters is merely a linear combination of the gain factor and the value of the TEE. Nevertheless, the change updates the trajectory which will hence affect the reading from the sensors and adjust the energy evolution accordingly. The linearity simulates how a variation affects the energy, for instance, a decrement in the radius of a circular TEE will add a negative contribution, the simulate the lowering of instantaneous energy consumption.

In the case of the system behaving ideally (i.e., with zero uncertainty), we expect a set of states evolving accordingly to their outputs. Such observation is summarized in the following Lemma.

Lemma 2.2: Given two time instants $k_1, k_2 \in [t_f]$, suppose the system of Equation (6) evolves with no uncertainty ($\mathbf{w} = \mathbf{0}, v = 0$) and Assumption 2.1 holds, then

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

Proof:

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

■

—from here on work in progress—

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 linear interpolation, one per each task. It hence 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 (QoS).

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

$$\mathcal{C}_k := \left\{u : u \in \mathrm{QoS}_{k,n} \;\; \forall n \in \{0, \dots, \sigma-1\}, \forall k \in \mathbb{T}\right\}, \tag{11}$$

where $\mathrm{QoS}_{k,n} : \mathbb{Z}_{\geq 0} \to \mathbb{Z}_{\geq 0}$ 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. 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

$$y^c_k := g\left(\mathrm{QoS}_{k,0}, \dots, \mathrm{QoS}_{k,\sigma-1}\right) = g\left(\mathcal{C}_k\right). \tag{12}$$

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

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

where the values $\underline{\mathrm{QoS}}_{k,n}, \overline{\mathrm{QoS}}_{k,n}$ are retrieved from the mission $\mathcal{M}$.

III. Algorithm

Let us proof formally an important finding from Section II-A extensively used in the algorithm.

Theorem 3.1: Consider a continuously differentiable function $\varphi_k : \mathbb{R}^2 \times \mathbb{R}^\sigma \to \mathbb{R}$ at a time instant $k \in \mathbb{T}$. Assume Assumption 2.1 holds, the robots is free to move in $\mathcal{P}$ defined in (3), and is following φ with the direction $\dot{\mathbf{p}}_d$ defined in 4. Likewise in Lemma 2.2, the model behaves ideally. Then, the instantaneous energy consumption is a linear combination of the state

$$y_k = C\mathbf{q}_k = \sum_{n=0}^{r} \alpha_n, \tag{14}$$

where $\alpha_n \in \mathbf{q}_k$ are the $r+1$ state's components at k with r being a preassigned arbitrary order from (5), and C is described in Equation (7).

Proof: *

■

A. State estimation

As the environment uncertainty and measurement error evolve in a normal distribution, we use a Kalman filter [24], [25] for the purpose of state estimation.

The prediction is done using

$$\hat{\mathbf{q}}^-_{k+1} = A\hat{\mathbf{q}}_k + B\mathbf{u}_k, \tag{15a}$$

$$P^-_{k+1} = AP_kA^T + Q, \tag{15b}$$

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

$$K_k = (CP^-_{k+1}C^T + R)^{-1}(P^-_{k+1}C^T), \tag{16a}$$

$$\hat{\mathbf{q}}_{k+1} = \hat{\mathbf{q}}^-_{k+1} + K_k(y^s_k + y^c_k - C\hat{\mathbf{q}}^-_{k+1}), \tag{16b}$$

$$P_{k+1} = (I - K_kC)P^-_{k+1}, \tag{16c}$$

$$\hat{y}_k = C\hat{\mathbf{q}}_{k+1}, \tag{16d}$$

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 (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 (15–16) 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 (15a) and (15b) respectively. The value of $\hat{\mathbf{q}}_1^-$ is then used in Equation (16b) 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 (16d). 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] 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] R. F. Stengel, *Optimal control and estimation*. Courier Corporation, 1994.

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