



Article Development of Path Planning Tool for Unmanned System Considering Energy Consumption

Sunghun Jung

Department of Electric Vehicle Engineering, Dongshin University, Jeollanam-do 58245, Korea; jungx148@gmail.com; Tel.: +82-61-330-7651

Received: 29 January 2019; Accepted: 13 August 2019; Published: 14 August 2019



Abstract: Increasing the flight endurance of unmanned aerial vehicles (UAVs) has received attention recently. To solve this problem, two research topics have generally appeared: Shortest-path planning (SPP) and remaining-flying-range estimation. In this work, energy-efficient path planning by considering the distance between waypoint nodes, the minimum and maximum speed of the UAV, the weight of the UAV, and the angle between two intersecting edges is proposed. The performances of energy-efficient path planning (EEPP) and generic shortest-path planning are compared using extended-Kalman-filter-based state-of-charge and state-of-power estimation. Using this path-planning tool and considering energy consumption during flight operation, two different path plans can be obtained and compared in advance so that the operator can decide which path to choose by consulting a comparison chart. According to the experimental results, the EEPP algorithm results in 0.96% of improved SOC leftover and 11.03 (W) of lowered SOP compared to the SPP algorithm.

Keywords: EEPP; EKF; path planning; SOC; SOP

1. Introduction

As unmanned aerial vehicle (UAV) technology matures, the longtime flight endurance capability of UAVs is gaining attention [1,2]. Because the energy density (W/kg) of the battery pack (in particular, Li-ion) is closely correlated with the overall flight time of the UAV, it is recommended to choose cells possessing as high an energy density as possible [3,4]. However, in contrast to electric vehicles (EVs), the operation times of electric aircraft, including the UAV, drastically decrease as the system weights increase. Hence, technology development for increasing UAV flight time has been delayed for a long time.

Lately, the EV industry has been growing quickly, and automotive industry researchers have focused their attention on accurately predicting the energy leftover so that the driving range can be extended. Accurate energy leftover monitoring (state of charge (SOC) estimation in the EV industry), aims to extend the driving range, and it is related to shortest-path planning (SPP), which is essential to saving energy. Both accurate SOC estimation and SPP should ultimately enhance the fuel economy of EVs, and the same philosophy applies to UAVs.

There have been numerous studies regarding path planning. Chang et al. proposed a method to achieve an effective delivery route for trucks carrying delivery drones with moving shift-weights-based K-means clustering and traveling-salesman-problem modeling [5]. Rubio et al. presented adaptive path planning for multiple UAVs using evolutionary computation combined with market-based cooperation strategies [6].

Further, there has been research regarding the accurate estimation of the remaining driving or flying range. Oliva et al. suggested a model-based approach by combining unscented filtering and Markov chains to predict the remaining driving range (RDR) [7]. Ondruska et al. proposed the efficient computation of RDR confidence level using a feature-based linear regression framework with a varying

probabilistic attainability map in real-time [8]. Ferreira et al. presented an interaction method between a driver and EV to estimate EV range using a regression-model-based data mining approach [9].

Unlike the previously listed works, this study contributes to the energy-efficient path planning (EEPP) of the UAV to ultimately increase the flight endurance compared with the general SPP. To verify the improved fuel efficiency, the concept of the extended Kalman filter (EKF)-based SOC estimation method is adopted. To mimic cell behavior, a physics-based equivalent circuit model (ECM) is used [10–13].

The overall mission hierarchy to achieve an energy-efficient path proposed in this paper is as follows.

First, with the given satellite map, the EEPP algorithm generates the most energy-efficient path with step-by-step processes; (1) obstacle detection, (2) extract obstacle, (3) obstacle separation, (4) add buffer zone, (5) Voronoi, (6) filter out useless candidates, (7) 1D optimal control, and (8) SPP or EEPP selection. Here, the format of the output path has both Euclidean and GPS coordinates. In contrast to the SPP algorithm, which uses Dijkstra algorithm, the EEPP algorithm uses three step-by-step processes; (1) find all trajectory candidates, (2) calculate total unit energy consumption (UEC) of all trajectory candidates, and (3) find a trajectory having the minimum UEC.

Second, either flight simulation with equivalent circuit model (ECM) of the Li-Ion battery pack or the flight experiment with a real Li-Ion battery pack is performed using the previously obtained energy-efficient path and gives out the voltage and current profiles of each cell.

Third, a battery state estimator uses the voltage and current profiles of cells to calculate the SOC. In particular, the battery state estimator uses the EKF to calculate the SOC.

The flow of this paper is as follows: In Section 2, the overall mission hierarchy, including four subparts (path planning, flight simulation, flight experiment, and battery state estimation), is described. In Section 3, the path planning, including the SPP and EEPP algorithms, is explained. Section 4 covers the mathematical formulation of the UAV and battery pack. Section 5 presents the simulation and experimental setups, while Section 6 shows the simulation and experiment results. Section 7 contains the conclusion of this paper and describes future works.

2. Mission Hierarchy

Figure 1 depicts the overall mission hierarchy to achieve the most energy-efficient path and corresponding estimated SOC value.



Figure 1. Mission hierarchy.

First, using a Google Earth satellite photograph, and the takeoff and landing locations, the most energy-efficient path is determined by the path-planning part. Here, the path is saved in both the Cartesian system unit, $Trajectory_{xyz}$, and the geographic coordinate system unit, $Trajectory_{GPS}$.

Second, the achieved path is utilized for the indoor flight simulation and outdoor flight experiment to calculate the cell open-circuit voltage (OCV), v_{ocv} , and load current, i_o , data.

Third, the achieved cell voltage and current data are utilized for the battery state estimation to calculate the SOC. Both SOC_{Sim} and SOC_{Exp} are calculated to compare results.

With the achieved SOC value, one can finally determine whether the EEPP algorithm results in an energy-efficient path.

3. Path Planning

Figure 2 displays a magnified version of the path-planning part of Figure 1. This part results in either the shortest path or energy-efficient path. The detailed description of each subpart is given elsewhere [14].



Figure 2. Flowchart of path planning.

3.1. SPP

The Dijkstra algorithm is used for the SPP algorithm, which utilizes the weights of each edge between nodes to calculate the minimum sum of the total distance from the start node to the goal node [15].

The overall flowchart of the SPP algorithm is shown in Figure 3.



Figure 3. Flowchart of shortest-path planning (SPP) algorithm.

3.2. EEPP

In contrast to the general SPP algorithm, which only accounts for the distance between nodes, the EEPP algorithm calculates the energy-efficient path by considering factors including the distance between nodes, the minimum and maximum speed of the UAV, and the weight of the UAV as shown in Figure 4.

The overall amount of battery consumption of the UAV, B_o , would be

$$B_o = B_t + B_h + B_a + B_n + B_d + B_c + B_l,$$
(1)

where B_t is the battery consumption during takeoff (W), B_h is the battery consumption during hovering (W), B_a is the battery consumption during acceleration (W), B_n is the battery consumption during normal flight (W), B_d is the battery consumption during deceleration (W), B_c is the battery consumption during the circular flight (W), and B_l is the battery consumption during landing (W).

Because the takeoff and landing occur in a short period and the amount of battery consumption is insignificant compared with the total consumption, B_t and B_l are assumed to be zero, and Equation (1) becomes

$$B_o \approx B_h + B_a + B_n + B_d + B_c. \tag{2}$$

If there are n_e number of edges in the given path,

$$B_o \approx \sum_{i=1}^{n_e} (B_h + B_a + B_n + B_d + B_c).$$
 (3)

From the equation of a single rotor helicopter in hover [16,17], the required hovering power of the UAV can be derived as

$$B_h = m^{\frac{3}{2}} \sqrt{\frac{g^3}{2\rho\zeta n_r}},\tag{4}$$

where *m* is the mass of the UAV (kg), *g* is the gravitational acceleration (m/s²), ρ is the fluid density of air (kg/m³), ζ is the total area of the spinning rotor disk (m²), and *n_r* is the number of rotors (no unit).

The required power for the acceleration can be derived as

$$B_{a} = \frac{1}{2t_{a}} \left(\int_{0}^{t_{a}} a_{\max} dt \right)^{2} = \frac{1}{2} m a_{\max}^{2} t_{a},$$
(5)

where a_{max} is the maximum acceleration (m/s²), and t_a is the time taken for the acceleration (s).

The required power for the deceleration can be derived as

$$B_d = \frac{1}{2t_d} \left(\int_0^{t_d} a_{\min} dt \right)^2, \tag{6}$$

where a_{min} is the minimum acceleration (m/s²), and t_d is the time taken for the deceleration (s). Here, with assumptions that $t_a = t_d$ and $a_{max} = -a_{min}$, one can conclude $B_a = B_d$.

The required power for the normal (constant speed) flight can be derived as

$$B_n = \frac{1}{2t_n} m v_{\max}^2.$$
 (7)

Here, v_{max} is the maximum velocity (m/s²), and t_n is the time taken for the normal flight (s).

The required power for the circular flight can be derived as

$$B_c = \frac{1}{2t_c} m v_{\min}^2 \tag{8}$$

where v_{min} is the minimum velocity (m/s²), and t_c is the time taken for the circular flight (s).

The unknown value, t_a , in Equation (5) can be calculated as

$$t_a = t_d = \frac{v_{\max} - v_{\min}}{a_{\max}}.$$
(9)

The unknown value, t_n , in Equation (7) can be calculated as

$$\frac{1}{2}t_c = \frac{1}{v_{\min}} \left\{ \frac{v_{\min}^2}{a_{\max}} \tan\left(\frac{\pi}{2} - \frac{\alpha}{2}\right) \right\},\tag{10}$$

therefore,

$$t_c = 2\frac{v_{\min}}{a_{\max}} \tan\left(\frac{\pi}{2} - \frac{\alpha}{2}\right),\tag{11}$$

where α is the intersection angle of two edges, which satisfies $0 \le \alpha \le \pi$.

Because

$$b_{1} = t_{a}v_{\min} + \frac{1}{2}t_{a}(v_{\max} - v_{\min}), b_{2} = \frac{v_{\min}^{2}}{a_{\max}^{m}} \tan\left(\frac{\pi}{2} - \frac{\alpha}{2}\right),$$
(12)

the unknown value, t_c , in Equation (8) can be calculated as

$$t_n = \frac{1}{v_{\max}} \{ \|W_{i-1}, W_i\| - 2(b_1 + b_2) \},$$
(13)

where W_i is the waypoint (no unit), b_1 is the distance in which the UAV accelerates or decelerates (m), and b_2 is the distance at which the UAV flies in a circular motion (m).



Figure 4. Analysis of flight path using the Dubins path method [18–20].

If Equations (4)–(13) are combined into Equation (3), then

$$B_{0} = \sum_{i=1}^{n_{e}} \left[m^{\frac{3}{2}} \sqrt{\frac{g^{3}}{2\rho\zeta n_{r}}} + ma_{\max}(v_{\max} - v_{\min}) + \frac{mv_{\max}^{3}}{2\{||W_{i-1}, W_{i}|| - 2(b_{1} + b_{2})\}} + \frac{mv_{\min}a_{\max}}{4\tan\left(\frac{\pi}{2} - \frac{\alpha}{2}\right)} \right].$$
(14)

To calculate the total unit energy consumption (UEC) (Wh), U_b , Equation (14) is converted as

$$B_{0} = \frac{t_{t}}{3600} \sum_{i=1}^{n_{e}} \left[m^{\frac{3}{2}} \sqrt{\frac{g^{3}}{2\rho\zeta n_{r}}} + ma_{\max}(v_{\max} - v_{\min}) + \frac{mv_{\max}^{3}}{2\{||W_{i-1}, W_{i}|| - 2(b_{1} + b_{2})\}} + \frac{mv_{\min}a_{\max}}{4\tan\left(\frac{\pi}{2} - \frac{\alpha}{2}\right)} \right]$$
(15)

where $t_t = t_a + t_n + t_d + t_c$.

The overall flowchart of the EEPP algorithm is shown in Figure 5.



Figure 5. Flowchart of the energy-efficient path planning (EEPP) algorithm.

4. Mathematical Modeling of the UAV and Battery Pack

4.1. UAV

The configuration and governing control inputs of the quadrotor are described in Figure 6, where X_E , Y_E , and Z_E represent earth coordinates, x_q , y_q , and z_q represent quadrotor coordinates, and F_i represents a thrust force generated by the *i*-th rotor in the indicated direction.



Figure 6. Quadrotor configuration.

To extract current and voltage profiles along the predetermined path obtained from the previous section, an enhanced quadrotor dynamic model can be represented as

$$w_{1}^{2} = b(T + \tau_{x} + \tau_{y} + \tau_{z})^{-1}$$

$$w_{2}^{2} = lb(-\tau_{x} + \tau_{z})$$

$$w_{3}^{2} = lb(-T + \tau_{y})$$

$$w_{4}^{2} = d(-T + \tau_{x} - \tau_{y} + \tau_{z})$$
(16)

where *T* is the total upward thrust of the UAV (N), *b* is the thrust factor, and w_i is the *i*-th rotor speed (rad/s), *l* is the lever length from the center of gravity (CG) to rotors (m), *d* is the drag factor, and τ_x , τ_y , and τ_z are torques applied on the UAV's body along *x*-, *y*-, and *z*-axes (N·m) [21,22].

Here, T, τ_x , τ_y , and τ_z can be calculated using proportional–integral–derivative (PID) controllers as

$$T = k_{p_T}(z^* - z) + k_{i_T} \int_{t_1}^{t_2} (z^* - z) dt + k_{d_T} (\dot{z}^* - \dot{z}) + \sqrt{W/4b},$$

$$\tau_x = k_{p_x}(\phi^* - \phi) + k_{i_x} \int_{t_1}^{t_2} (\phi^* - \phi) dt + k_{d_x} (\dot{\phi}^* - \dot{\phi}),$$

$$\tau_y = k_{p_y}(\theta^* - \theta) + k_{i_y} \int_{t_1}^{t_2} (\theta^* - \theta) dt + k_{d_y} (\dot{\theta}^* - \dot{\theta}),$$

$$\tau_z = k_{p_z}(\psi^* - \psi) + k_{i_z} \int_{t_1}^{t_2} (\psi^* - \psi) dt + k_{d_z} (\dot{\psi}^* - \dot{\psi}),$$

(17)

where k_p , k_i , and k_d are PID controller gains, ϕ , θ , and ψ are roll, pitch, and yaw angles (rad), uppercase letter * represents reference data, and W is the UAV weight (kg).

The full quadrotor UAV dynamic model can be written as

$$\begin{aligned} \ddot{x} &= (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)\frac{1}{m}b\Big(w_1^2 + w_2^2 + w_3^2 + w_4^2\Big), \\ \ddot{y} &= (\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)\frac{1}{m}b\Big(w_1^2 + w_2^2 + w_3^2 + w_4^2\Big), \\ \ddot{z} &= (\cos\phi\cos\theta)\frac{1}{m}b\Big(w_1^2 + w_2^2 + w_3^2 + w_4^2\Big) - g, \\ \ddot{\phi} &= \dot{\theta}\dot{\psi}\Big(\frac{I_y - I_z}{I_x}\Big) - \frac{I_r}{I_x}\dot{\theta}(w_1 + w_3 - w_2 - w_4) + \frac{1}{I_x}b\Big(w_4^2 - w_2^2\Big), \\ \ddot{\theta} &= \dot{\phi}\dot{\psi}\Big(\frac{I_z - I_x}{I_y}\Big) - \frac{I_r}{I_y}\dot{\phi}(w_1 + w_3 - w_2 - w_4) + \frac{1}{I_y}b\Big(w_3^2 - w_1^2\Big), \\ \ddot{\psi} &= \dot{\phi}\dot{\theta}\Big(\frac{I_x - I_y}{I_z}\Big) + \frac{1}{I_z}d\Big(w_2^2 + w_4^2 - w_1^2 - w_3^2\Big). \end{aligned}$$
(18)

Here, J_r is the rotor inertia (kg·m²).

Using the fact that power can be calculated by multiplying the thrust and velocity, one can calculate the required propulsive thrust power from the total upward thrust *T* in Equation (15) as

$$P = T \cdot \|v_0\|. \tag{19}$$

Then, the load current can be calculated using

$$\frac{P}{v_t} = i_0 \tag{20}$$

where v_t is the terminal voltage (V), and i_0 is the load current (A).

The above equations are used for the operation of flight simulation, as shown in Figure 7.



Figure 7. Flowchart of the EEPP algorithm.

4.2. Battery Pack

4.2.1. ECM

By using the two-ladder battery ECM and impedance voltage response, as shown in Figure 8, the v_t can be determined as

$$v_t = v_{ocv} - i_0 R_0 - v_{c1} - v_{c2} = v_{ocv} - i_0 R_0 - i_0 R_1 (1 - e^{-\Delta t/\tau_1}) - i_0 R_2 (1 - e^{-\Delta t/\tau_2})$$
(21)

where v_{ocv} is the OCV voltage (V), R_0 is the resistor (Ω), v_{ci} is the voltage across the *i*-th RC-network (V), τ_i is the time constant (s), which is calculated as $\tau_i = R_i C_i$ in which R_i is the resistor (Ω) and C_i is the capacitor (F) of the *i*-th ladder, and *t* is the time (s).



Figure 8. Battery modeling; (**a**) two-ladder battery equivalent circuit model (ECM); (**b**) impedance voltage response.

4.2.2. TBLI Parameter Identification

The ECM parameters, including R_0 , R_1 , C_1 , R_2 , and C_2 , are extracted at SOC 10% intervals using the high-performance pulse characterization current pattern, as shown in Appendix A [12]. Here, the table-based linear interpolation (TBLI) method is used to determine ECM parameters corresponding to varying SOC values. Although R_0 , R_1 , C_1 , R_2 , and C_2 are the functions of the SOC, C-rate, and temperature, only the SOC variance effect is considered in this research.

4.2.3. SOC and SOP State Estimation

The SOC estimation is performed using the EKF for each cell. For the EKF algorithm application, Equation (21) and SOC calculation based on current integration method are fused and converted into the discrete-time state space model as

$$\begin{bmatrix} v_{c1} \\ v_{c2} \\ z \\ z \\ z \\ z \end{bmatrix}_{k+1} = \begin{bmatrix} e^{-\Delta t/\tau_1} & 0 & 0 \\ 0 & e^{-\Delta t/\tau_2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_{c1} \\ v_{c2} \\ z \\ z \end{bmatrix}_k + \begin{bmatrix} R_1(1 - e^{-\Delta t/\tau_1}) \\ R_2(1 - e^{-\Delta t/\tau_2}) \\ \eta \Delta t/(3600Q_{norm}) \end{bmatrix}_k i_{0,k} + \begin{bmatrix} n_{w1} \\ n_{w2} \\ n_{w3} \end{bmatrix}_k, \quad (22)$$

$$\begin{bmatrix} v_{c1} \\ v_{c2} \\ z \\ z \end{bmatrix}_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_{c1} \\ v_{c2} \\ z \\ z \end{bmatrix}_k + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ z \end{bmatrix}_k + \begin{bmatrix} n_{v1} \\ n_{v2} \\ n_{v3} \end{bmatrix}_k,$$

where *z* is the SOC (%/100), Δt is 0.1 s, and n_{wi} and n_{vi} are system and measurement noises (V), respectively. Here, the term *z* can be calculated as $z(t) = Q_{init} + \eta i_0(t)/Q_{norm}(t)$ where Q_{init} is the initial battery capacity (Ah), η is the battery efficiency (assumed as 1), and Q_{norm} is the current battery capacity (Ah).

With an initial estimation vector, \hat{x}_0 , and covariance vector, P_0 , the estimation vector and covariance vector can be propagated using

$$\hat{x}_{k+1}^{-} = f(\hat{x}_k), P_{k+1}^{-} = A_{k+1} P_k A_{k+1}^T + Q_k.$$
(23)

The Kalman gain, K_k , can be calculated as

$$K_{k+1} = P_{k+1}^{-} H_{k+1}^{T} \left(H_{k+1} P_{k+1}^{-} H_{k+1}^{T} + R_{k+1} \right)^{-1}.$$
(24)

With the updated Kalman gain, K_{k+1} , and updated measurement, z_{k+1} , the updated estimation vector, \hat{x}_{k+1} , can be calculated as

$$\hat{x}_{k+1} = \hat{x}_{k+1} + K_{k+1} \Big(z_{k+1} - h_{k+1} \Big(\hat{x}_{k+1} \Big) \Big), \tag{25}$$

and the updated covariance vector, P_{k+1} , can be calculated as

$$P_{k+1} = P_{k+1}^{-} - K_{k+1}H_{k+1}P_{k+1}^{-}.$$
(26)

The final SOC value of the battery pack is conservatively chosen as the minimum value of each cell as

$$SOC_{pack,k} = \min(z_k^1, \dots, z_k^n) \times 100,$$
(27)

where $SOC_{pack,k}$ is the SOC (%) of a battery pack at the *k*-th time step.

Regarding the state of power (SOP) estimation, the maximum discharging current, *i*^{dis}_{max}, can be calculated as

$$i_{\max}^{dis} = C_{rate} Q_{norm} \tag{28}$$

where C_{rate} is the C-rate of the cell (no unit).

The maximum discharging current considering the SOC, $i_{max,k}^{dis,SOC(i)}$, can be calculated as

$$i_{\max,k}^{dis,SOC(i)} = \left\{ \left(z_k^{(i)} - 3\sigma_{z,k}^{(i)} \right) - z_{\min} \right\} C_{rate} Q_{norm}$$
⁽²⁹⁾

where σ is the standard deviation of the cell SOCs (%/100), z_{min} is the minimum SOC (%/100), η is the battery efficiency (assumed as one), and $T_k - T_o$ is the time flown from the beginning (s). Here, 3σ was used for the 99.7% confidence interval of the cell SOC estimation values. The maximum discharging current considering the cell voltage, $i_{max,k}^{dis,volt(i)}$, can be calculated as

$$i_{\max,k}^{dis,volt(i)} = |i_{0,k}| \left\{ v_{ocv} \left(z_k^{(i)} \right) - v_{\min}^{(i)} \right\} \left\{ \frac{1}{\left| v_{ocv} \left(z_k^{(i)} - v_{\min}^{(i)} \right) \right|} + \frac{1}{v_{\max}^{(i)} - \left\{ v_{ocv} \left(z_k^{(i)} \right) - v_{\min}^{(i)} \right\}} \right\},\tag{30}$$

where v_{ocv} is the open-circuit voltage (OCV) (V), and v_{min} is the minimum cell voltage (V). Here, $v_{min}^{(1)} = 3.679 \text{ V}, v_{min}^{(2)} = 3.660 \text{ V}, v_{min}^{(3)} = 3.663 \text{ V}, v_{min}^{(4)} = 3.626 \text{ V}, v_{max}^{(1)} = 4.171 \text{ V}, v_{max}^{(2)} = 4.176 \text{ V},$ $v_{max}^{(3)} = 4.174 \text{ V}, v_{max}^{(4)} = 4.174 \text{ V}, z_{min} = 0, C_{rate} = 18 \text{ C}, N_p = 1, N_s = 4, \text{ and } Q_{norm} = 3.3 \text{ Ah}.$ With the previously found $i_{max}^{dis}, i_{max,k}^{dis,SOC(i)}$, and $i_{max,k}^{dis,volt(i)}$, one can calculate $i_{max,k}^{dis}$ as

$$i_{\max,k}^{dis} = \min\left(i_{\max}^{dis}, i_{\max,k}^{dis,SOC(i)}, i_{\max,k}^{dis,volt(i)}\right)$$
(31)

and can calculate $P_{max,k}^{dis(i)}$ as

$$P_{\max,k}^{dis(i)} = i_{\max,k}^{dis} \left\{ v_{ocv} \left(z_k^{(i)} \right) - i_{\max,k}^{dis} \left| \frac{v_{ocv} \left(z_k^{(i)} \right) - v_{\min}}{i_{0,k}} \right| \right\}.$$
(32)

Then, the maximum power that each cell could perform, $P_{max}^{dis(i)}$, is calculated as

$$P_{\max}^{dis(i)} = v_{\max}^{(i)} i_{\max}^{dis}$$
(33)

and compared with the maximum power that each cell performs during the discharging process, $P_{max,k'}^{dis(i)}$ as

$$P_{\max,k}^{dis(i)} = N_p \min\left(N_s P_{\max}^{dis(i)}, \sum_{i=1}^{N_s} i_{\max,k}^{dis} \left(v_{ocv}\left(z_k^{(i)}\right) - i_{\max,k}^{dis} R_{dis}^{(i)}\right)\right)$$
(34)

where the total initial resistance of each cell, $R_{dis}^{(i)}$, can be calculated as

$$R_{dis}^{(i)} = \frac{v_{ocv}(z_k^{(i)}) - v_{\min}}{i_{0,k}}.$$
(35)

Figure 9 presents the flowchart of the battery state estimation [23].



Figure 9. Flowchart of battery state estimation.

5. Simulation and Experimental Setups

5.1. Simulation Setup

All simulations were performed with a quadrotor vehicle with initial system properties $P_o = [0,0,0] \text{ m}$, $\theta_o = [0,0,0] \text{ rad}$, $w_o = [0,0,0] \text{ rad/s}$, $v_o = [0,0,0] \text{ m/s}$, $a_o = [0,0,0] \text{ m/s}^2$, $||v_{min}||_2 = 0 \text{ m/s}$, $||v_{max}||_2 = 5 \text{ m/s}$, $||a_{max}||_2 = 0.5 \text{ m/s}^2$, $Alt_{min} = 3 \text{ m}$, and $Alt_n = 30 \text{ m}$, where P_o , θ_o , w_o , v_o , a_o , Alt_{min} , and Alt_n are initial position, attitude, angular velocity, velocity, acceleration, minimum flight altitude, and normal flight altitude, respectively. Also, the time step used in MATLAB/Simulink is 0.01 s, installed in a desktop with the following system specifications; Intel Core i5-4590 CPU 3.40-GHz processor, 64-bit operating system, and 4.00-GB RAM. Every simulation flight was performed with a fully charged 4s 3300-mAh battery pack.

5.2. Experiment Setup

All experiments were performed with the parts shown in Figure 1—a PX4 Pro powered F450 quadrotor, shown in Figure 10a; Pixhawk 2.1 flight controller, as in Figure 10b; QGroundControl (QGC), shown in Figure 10c; Pixhawk 2.1 Here + RTK, as in Figure 10d; eLogger V4 voltage and current data logging system, as shown in Figure 10e; and a 4s 3300-mAh battery pack, shown in Figure 10f—for the outdoor autopilot experiment with a previously achieved energy-efficient path. In particular, the Pixhawk 2.1 Here + RTK is used for the centimeter-level autopilot path following accuracy.



Figure 10. Flight experiment parts: (a) F450 quadrotor, (b) Pixhawk 2.1, (c) QGroundControl (QGC), (d) Pixhawk 2.1 Here+ RTK, (e) eLogger V4, and (f) 4s 3300-mAh battery pack.

The overall experiment setup is presented in Figure 11. On the day of the experiment, the wind speed was approximately 2 m/s.

For the experiment, the SPP and EEPP trajectories represented as the Cartesian system unit, $Trajectory_{xyz}$, obtained from the path-planning algorithm shown in Figure 2 are converted into the geographic coordinate system unit, $Trajectory_{GPS}$. The detailed description can be found in previous work [24].

In particular, the waypoint list of $Trajectory_{GPS}$ is saved as a text file following the Pixhawk waypoint file format, as depicted in Table 1 [25]. For the waypoints lists of SPP and EEPP trajectories, see Appendix B.



Figure 11. Flight experiment setup.

Table 1. Waypoint file format.

QGC WPL < VERSION>					
<index></index>	<current wp=""></current>	<coord frame=""></coord>	<command/>	<param1></param1>	<param2></param2>
<param3></param3>	<param4></param4>	<param5 <br="" x="">LONGITUDE></param5>	< PARAM6/Y/ LATITUDE >	< PARAM5/Z/ ALTITUDE >	<autocontinue></autocontinue>

6. Simulation and Experimental Results

6.1. Simulation Result

From the flight simulation, the obtained current and voltage profiles are shown in Figure 12. Here, negative current represents discharging.



Figure 12. Current and voltage profiles (simulation): (a) Current profile and (b) voltage profile.

Energy consumption comparison results of the SPP and EEPP algorithms are listed in Table 2 and Figure 13.

	*			
	SPP		EEPP	
	Prediction Value	Actual Value	Prediction Value	Actual Value
Total Trajectory Distance (m)	204.63	193.30	208.76	201.08
Min Speed of UAV (m/s)	0	0	0	0
Max Speed of UAV (m/s)	5	0.34	5	0.33
Weight of UAV (kg)	1.41	1.41	1.41	1.41
Total UEC (Wh)	4.47	11.54	5.71	11.99
Algorithm Run Time (s)	11.5	1	20.34	
SOC Leftover (%)	72.0	0	70.80	
SOP Peak (W)	566.2	20	543.	47

The percentage difference between the actual values of SPP and EEPP is shown in Table 3 and Figure 14. According to the result, the SOC leftover is larger and the SOP peak is lower when using SPP.



Figure 13. State of charge (SOC) and state of power (SOP) comparison (simulation): (**a**) SOC and (**b**) SOP.

	$\frac{SPP-EEPP}{SPP} \times 100 \ (\%)$
Total Trajectory Distance (m)	-4.02
Min. Speed of UAV (m/s)	0
Max. Speed of UAV (m/s)	2.94
Weight of UAV (kg)	0
Total UEC (Wh)	-3.90
Algorithm Run Time (s)	-76.72
SOC Leftover (%)	1.67
SOP Peak (W)	4.01

Table 3. Comparison results in percentage (simulation).

10

-10



Total Trajectory Distance Max Speed of UAV Total UEC Algorithm Run Time SOC Leftover SOP Peak

Figure 14. The comparison results in percentage (simulation).

6.2. Experimental Result

From the flight experiment, the obtained current and voltage profiles are shown in Figure 15.



Figure 15. Current and voltage profiles (experiment): (a) Current profile and (b) voltage profile.

Energy consumption comparison results of the SPP and EEPP algorithms are shown in Table 4 and Figure 16.

	SPP		EEPP	
	Prediction Value	Actual Value	Prediction Value	Actual Value
Total Trajectory Distance (m)	205.75	174.11	212.17	183.64
Min. Speed of UAV (m/s)	0	0	0	0
Max. Speed of UAV (m/s)	5	4.20	5	4.61
Weight of UAV (kg)	1.41	1.41	1.41	1.41
Total UEC (Wh)	4.47	7.94	5.71	7.41
Algorithm Run Time (s)	11.5	1	20.34	
SOC Leftover (%)	83.7	3	84.53	
SOP Peak (W)	448.8	85	498.37	

Table 4. Energy consumption comparison (experiment).

The percentage difference between the actual values of SPP and EEPP is shown in Table 5 and Figure 17. According to the result, the SOC leftover is larger and the SOP peak is lower when using EEPP.



Figure 16. SOC and SOP comparison (experiment): (a) SOC and (b) SOP.

	$\frac{SPP-EEPP}{SPP} \times 100 \ (\%)$
Total Trajectory Distance (m)	-5.47
Min. Speed of UAV (m/s)	0
Max. Speed of UAV (m/s)	-9.76
Weight of UAV (kg)	0
Total UEC (Wh)	6.68
Algorithm Run Time (s)	-76.72
SOC Leftover (%)	-0.96
SOP Peak (W)	-11.03

Table 5. Comparison result in percentage (experiment).



Figure 17. The comparison results in percentage (experiment).

The overall reference, simulation, and experiment flight trajectories of the SPP and EEPP algorithms are drawn on Google Earth, as shown in Figure 18. Here, yellow polygons represent imaginary obstacles that the UAV should avoid. The blue line represents the reference trajectory that the UAV should follow. The red line represents the simulation flight trajectory. The green line represents the experiment flight trajectory.



Figure 18. Reference, simulation, and experiment flight trajectories of: (**a**) SPP algorithm (2D), (**b**) EEPP algorithm (2D), (**c**) SPP algorithm (3D), and (**d**) EEPP algorithm (3D).

7. Conclusions

In this work, the EEPP algorithm is proposed and its performance is compared to the generic SPP algorithm using EKF-filter-based SOC estimation. Using the proposed path planning tool, two different path plans can be obtained and compared in advance. According to the experimental results, the EEPP algorithm results in an energy-efficient path.

Regarding the SOC, 0.96% is saved when using the EEPP algorithm, and this implies that the fuel economy is improved. Regarding the SOP, 11.03 (W) is lowered, and this implies that the internal shock of the battery pack is reduced. Although the rate of improved fuel economy and internal shock of the battery pack is not as good in this study, these two key factors can be enhanced in mission planning with longer trajectories. Here, the difference between simulation and experiment results appears to be caused by factors such as environmental effects and a less mature quadrotor simulation model.

In the future, instead of 2D path planning, 3D path planning with an enhanced EEPP algorithm are planned to greatly improve the fuel economy of the UAV. Also, rather than the offline method presented here, the development of online real-time energy-efficient path planning based on the probabilistic method is planned.

Acknowledgments: This work was supported by the JeonNam TechnoPark (JNTP) through the project (B0080802000131).

Conflicts of Interest: The author declares no conflict of interest.

Appendix A



Figure A1. ECM parameter values: (a) R0; (b) R1; (c) C1; (d) R2; (e) C2.

Appendix **B**



Figure A2. Waypoint list of SPP and EEPP trajectories: (a) SPP; (b) EEPP.

References

- 1. Kim, K.; Kim, T.; Lee, K.; Kwon, S. Fuel Cell System with Sodium Borohydride as Hydrogen Source for Unmanned Aerial Vehicles. *J. Power Sources* **2011**, *196*, 9069–9075. [CrossRef]
- Gao, X.; Hou, Z.; Guo, Z.; Fan, R.; Chen, X. The Equivalence of Gravitational Potential and Rechargeable Battery for High-Altitude Long-Endurance Solar-Powered Aircraft on Energy Storage. *Energy Convers. Manag.* 2013, 76, 986–995. [CrossRef]
- 3. Khaligh, A.; Li, Z. Battery, Ultracapacitor, Fuel Cell, and Hybrid Energy Storage Systems for Electric, Hybrid Electric, Fuel Cell, and Plug-In Hybrid Electric Vehicles: State of the Art. *IEEE Trans. Veh. Technol.* **2010**, *59*, 2806–2814. [CrossRef]
- 4. Noorden, R.A. Better Battery—Chemists are Reinventing Rechargeable Cells to Drive Down Costs and Boost Capacity. *Nature* 2014, 507, 26–28. [PubMed]
- 5. Chang, Y.S.; Lee, H.J. Optimal Delivery Routing with Wider Drone-Delivery Areas Along a Shorter Truck-Route. *Expert Syst. Appl.* **2018**, *104*, 307–317. [CrossRef]
- Rubio, J.C.; Vagners, J.; Rysdyk, R. Adaptive Path Planning for Autonomous UAV Oceanic Search Missions. In Proceedings of the AIAA 1st Intelligent Systems Technical Conference, Chicago, IL, USA, 20–22 September 2004.
- Oliva, J.A.; Weihrauch, C.; Bertram, T. A Model-Based Approach for Predicting the Remaining Driving Range in Electric Vehicles. In Proceedings of the Annual Conference of the Prognostics and Health Management Society, New Orleans, LA, USA, 14–17 October 2013.
- Ondruska, P.; Posner, I. Probabilistic Attainability Maps: Efficiently Predicting Driver-Specific Electric Vehicle Range. In Proceedings of the IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, MI, USA, 8–11 June 2014.
- Ferreira, J.C.; Monteiro, V.; Afonso, J.L. Dynamic Range Prediction for an Electric Vehicle. In Proceedings of the EVS27 International Battery, Hybrid and Fuel Cell Electric Vehicle Symposium, Barcelona, Spain, 17–20 November 2013.
- 10. Plett, G.L. *Battery Management Systems, II: Equivalent-Circuit Methods,* 2nd ed.; Artech House Publishers: Norwood, MA, USA, 2014.
- 11. Sharma, O.P. Analysis and Parameter Estimation of Li-Ion Batteries Simulations for Electric Vehicles. In Proceedings of the American Control Conference, San Francisco, CA, USA, 29 June–21 July 2011.
- 12. Thanagasundram, S.; Arunachala, R.; Makinejad, K.; Teutsch, T.; Jossen, A. A Cell Level Model for Battery Simulation. In Proceedings of the European Electric Vehicle Congress, Brussels, Belgium, 20–22 November 2012.
- Plett, G.L. Extended Kalman Filtering for Battery Management Systems of LiPB-based HEV Battery Packs–Part
 State and Parameter Estimation. *J. Power Sources* 2004, 134, 277–292. [CrossRef]
- 14. Jung, S.; Ariyur, K.B. Robustness for Scalable Autonomous UAV Operations. *Int. J. Aeronaut. Space Sci.* 2017, *18*, 767–779. [CrossRef]
- 15. Jung, S.; Ariyur, K.B. Scalable Autonomy for UAVs. In Proceedings of the AIAA Infotech@Aerospace, St. Louis, MO, USA, 29–31 March 2011.
- 16. Leishman, J.G. *Principles of Helicopter Aerodynamics*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2002.
- 17. Dorling, K.; Heinrichs, J.; Messier, G.G.; Magierowski, S. Vehicle Routing Problems for Drone Delivery. *IEEE Trans. Syst. Man Cybern. Syst.* 2017, 47, 70–85. [CrossRef]
- 18. Dubins, L.E. On Curves of Minimal Length with a Constraint on Average Curvature and with Prescribed Initial and Terminal Positions and Tangent. *Am. J. Math.* **1957**, *79*, 497–516. [CrossRef]
- 19. Tsourdos, A.; White, B.; Shanmugavel, M. *Cooperative Path Planning of Unmanned Aerial Vehicles*, 1st ed.; John Wiley & Sons: Chichester, UK, 2011.
- 20. Jung, S.; Ariyur, K.B. Enabling Operational Autonomy for UAVs with Scalability. *AIAA J. Aerosp. Inf. Syst.* **2013**, *10*, 516–529.
- 21. Robotics, P.C. Vision and Control: Fundamental Algorithms in MATLAB; Springer: Berlin/Heidelberg, Germany, 2011.
- Jung, S.; Ariyur, K.B. Strategic Cattle Roundup using Multiple Quadrotor UAVs. *Int. J. Aeronaut. Space Sci.* 2017, 18, 315–326. [CrossRef]

- 24. Steven, B.L.; Lewis, F.L. Aircraft Control and Simulation; John Wiley & Sons: New York, NY, USA, 2003.
- 25. QGroundControl—Waypoint Protocol. Available online: https://goo.gl/SqVVzd (accessed on 28 June 2018).



© 2019 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).