



Energy-Aware Multiflight Planning for an Unattended Seaplane: Flying Fish

Ryan D. Eubank*

Lincoln Laboratory, Massachusetts Institute of Technology Lexington, Massachusetts 02420

Justin M. Bradley†

University of Nebraska at Lincoln, Lincoln, Nebraska 68588

and

Ella M. Atkins‡

University of Michigan, Ann Arbor, Michigan 48109

DOI: 10.2514/1.I010484

Long-term unmanned vehicle operation requires autonomy capable of replanning activities responsive to changing vehicle and environment conditions. For unmanned aircraft systems, human handlers perform refueling/recharge and maintenance activities between flights, so the period of autonomy is typically limited to one flight. This paper investigates flight planning for a solar-energy-harvesting seaplane designed for persistent ocean surveillance without the need for human handling over a potentially long-term mission. A multiflight planner is introduced to generate energy-aware plans for persistent ocean surveillance. A novel heuristic is proposed to solve an asymmetric, nonmetric, negative-cost traveling salesman problem. Heuristic admissibility is demonstrated under specific conditions, and the characteristics of optimal multiflight plans are analyzed over a series of surveillance missions.

Nomenclature

A_i	=	area of the shape/object indicated by subscript i
a_{sun}	=	solar azimuth angle
b	=	aircraft wingspan
c	=	mean wing chord
e_{sun}	=	solar elevation angle
$f(\cdot)$	=	abbreviated notation for variables required to describe the sun-relative configuration of a solar array that includes position, attitude, and atmospheric conditions
\mathcal{P}_i	=	power of the system/process that is indicated by subscript i
$\mathcal{P}_{\text{spec}}$	=	power, per unit area, available from incident solar radiation
\bar{p}	=	satisfaction priority of mission goals or constraints
S	=	wing planform area
s_*	=	solar-incidence vector in the *coordinate frame
V, \check{V}_i	=	forward airspeed and trim/reference airspeed in trim/mode i
W	=	aircraft weight
\mathcal{E}_i	=	energy of the system/process that is indicated by subscript i
η_i	=	efficiency of the system/process that is indicated by subscript i
θ_{array}	=	solar array incidence (in vehicle frame, pitch axis)

I. Introduction

UNMANNED aircraft system (UAS) flight plans must remain within the vehicle's operational envelope and are subject to environment, payload, and energy storage constraints. Flight plans must achieve mission goals, avoid obstacles, and account for potentially hostile environmental conditions. Flight planners typically optimize single flights, assuming a "safe place" such as an airport or refueling perch will allow the system to stop and "reset" between flights. The unmanned seaplane engaged in persistent ocean surveillance (POS) may have no safe place to rest and reset; instead, it must cycle between periods of flight and open-water drift. For a long-duration POS mission, survival depends on careful energy management, which in turn depends on the environment. Successive single-flight plans will result in suboptimal long-term energy use and, potentially, energy depletion, necessitating planning over a multiflight horizon.

The University of Michigan designed, built, and tested the Flying Fish, which is an energy-harvesting autonomous seaplane-UAS (S-UAS) for POS. An initial phase 1 [1,2] vehicle provided proof of concept and motivated the larger solar-harvesting phase 2 vehicle (Fig. A1). This paper presents the solar-regenerative multiflight planner a S-UAS requires to factor energy collection and expenditure into plans that achieve mission goals without ever depleting energy stores. The greatest assets of the S-UAS (energy harvesting, flexible flight/drift profiles, and long-term deployment) also provide the greatest challenges for autonomous planning. A Flying Fish planner must support flexible mission duration ranging from a single surveillance flight to multiday or even season-long deployment.

The contributions of this work center around flight planning for our novel unmanned seaplane application. Traditional flight planners build an optimal plan from a specified takeoff site through a single landing or recovery site that may be adjusted in flight based on changes in the mission or environment. Because the seaplane drifts, for the first time, landing site selection is a function of expected postflight surface drift motion as well as

Received 16 June 2016; revision received 4 November 2016; accepted for publication 5 November 2016; published online 28 December 2016. Copyright © 2016 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved. All requests for copying and permission to reprint should be submitted to CCC at www.copyright.com; employ the ISSN 2327-3097 (online) to initiate your request. See also AIAA Rights and Permissions www.aiaa.org/randp.

*Technical Staff; eubankrd@umich.edu. Member AIAA.

†Assistant Professor, Computer Science and Engineering Department; justin.bradley@unl.edu. Senior Member AIAA.

‡Associate Professor, Aerospace Engineering Department; ematkins@umich.edu. Associate Fellow AIAA.

mission and in-flight environment conditions. This drift governs the takeoff location for the next flight, necessitating for the first time a multiflight planning horizon. Solar-energy harvesting also motivates multiflight planning because stored energy reserves plus expected capture during drift must balance or exceed cumulative energy expenditure over planned flights. Our multiflight planner must therefore integrate a suite of flight and drift vehicle dynamics, energy storage and harvesting, and environment models never before considered within a unified mission planning formulation.

In the following, the relevant background and baseline models to be integrated are presented. This paper presents energy and environmental models that supplement vehicle performance models presented in previous work [1,2]. A mission planner uses vehicle performance as well as energy collection and expenditure models to build surveillance mission plans. This mission planner models the planning problem as an asymmetric, nonmetric, negative-cost version of the nondeterministic polynomial-time (NP)-hard traveling salesman problem (TSP) [3]. To reduce search time, a novel application-motivated heuristic is developed based on a simplified TSP as an inner loop of the top-level planner. Unique challenges of the solar-regenerative S-UAS planning problem are examined in the context of multiday, overnight, and midday mission scenarios.

Example scenarios of each mission type for the Flying Fish platform are presented and analyzed. In this work, physics-based trajectory planning; vehicle performance; and models of energy harvesting, usage, and storage are all integrated into a discrete search engine for the determination of energy-optimal paths subject to solar-energy recovery dynamics. Although individual models are themselves adapted from the literature, the combination of vehicle performance, environment, and solar-energy-harvesting and usage models have never before been integrated into a system that optimizes energy use over multiple flights of a single aircraft capable of surveying targets while operating on the surface as well as while airborne.

II. Background and Related Work

The proposed multiflight planner draws from a variety of search-based modeling and inference techniques. An overview of pertinent planning literature is presented in the following, followed by a review of relevant applications.

A. Search and Planning Background

A planner translates mission goals, system and environment models, initial state, costs, and constraints into sequential or conditional plans of action [4]. Forward-chaining planners build a search tree from the initial state to the goal state(s) by exploring possible action sequences. Deterministic linear planners prescribe a fixed sequence of actions, nonlinear planners prescribe a set of actions with partial execution ordering constraints [5], and stochastic planners generate policies mapping observed states to goal-seeking actions [6].

Search-based planners are computationally intensive, given a sizable state and action space, resulting in the need for efficient abstractions, approximations, and heuristics. Abstraction can achieve substantial savings by grouping sequences of primitive actions into macroactions, as in Abstraction-Based Stanford Research Institute Problem Solver (ABSTRIPS) [7] and an adaptive planning system developed on top of FF (Macro-FF) [8]. Marvin [9] also exploits macroactions and employs a local hill-climbing search strategy to avoid the complexity associated with constructing a full search tree. Marvin is augmented with least-bad-first and greedy best-first search heuristics to address local search plateaus and local minima, respectively. Monte Carlo exploration has also been used to address plateau and local minima in local search strategies, although the inherent tradeoff between the efficiency of local exploration versus the completeness of global search remains.

Our paper adopts a global A^* search strategy to ensure long-term costs and utilities are appropriately considered. Given the associated complexity, emphasis is placed on identifying a good search heuristic. Heuristics have been frequently studied in the literature. An intuitive admissible heuristic for route planning is straight-line distance to the goal or destination. Straight-line distance is not universally applicable, as in the case of route planning with multiple goals for which other heuristics such as marginal utility have been proposed [10]. Marginal utility is the density of goal or high utility states in the search subtree to be expanded below a node within resource bounds. Because marginal utility may not be straightforward to compute, it must be inferred. Davidov et al. [10] defined partial marginal utility of a node as the number of goals found thus far over the number of states visited in this subtree: a domain-independent quantity that improves its estimate as the search space is further explored.

Search time and memory complexity are especially challenging to manage when planning must occur on board a platform with limited computational resources. Real-time constraints have been addressed with anytime and design-to-time [11] approaches. A design-to-time algorithm manages properties such as feature discretization level and model fidelity to regulate planner execution time. An anytime algorithm quickly identifies a baseline plan and then iteratively improves it over time, ideally allowing a result to be returned at any time. Because available planning time may be uncertain, several anytime strategies have emerged. For example, with “soft goals” that each increase overall mission utility but are not all required for success, partial satisfaction planning [12] can iteratively improve plans over time. Receding horizon formulations have proven effective the anytime approaches to optimal control [13,14]. An anytime heuristic search [15] may quickly identify a suboptimal solution that is improved (e.g., using A^* with a weighted heuristic) over time. Because an anytime search cannot always guarantee timely baseline response, researchers have also developed algorithms that return solutions in real time, regardless of problem size. For example, graph abstraction for local path planning can be used to manage search-space size along with planning horizon [16]. Partial-Refinement A^* [17] interleaves planning and execution, enabling an initial abstract map/grid to be refined over time.

Planner/scheduler formulations extend the basic search to manage real-time resource requirements and execution deadlines. For example, cooperative intelligent real-time control architecture (CIRCA) [18,19] uses a forward-chaining planner plus scheduler to build plans that execute with hard real-time guarantees. Sapa [20] is a metric temporal planner that can handle larger problem sizes with potential tradeoffs in optimality.

Motion planning is a special case of task planning in which paths or trajectories are computed a priori and updated online by reacting to incoming data [21]. Low-order models for a vehicle traversing a challenging environment have been used in roadmap [22–25] and rapidly exploring random tree [26–28] planners. Kinodynamic or trajectory planning builds feasible velocity profiles during planning using numerical methods such as sequential quadratic programming [29,30] or mixed-integer linear programming [31]. Analytic path planners such as the minimum-length Dubins algorithm [32] offer computational tractability and guaranteed convergence. Minimum-length paths are overlaid with vertical and feasible velocity profiles [21,33], as is done in our work.

B. Planning Applications

Flight planners for civil aviation specify plans as four-dimensional (4-D; latitude, longitude, altitude, and time) trajectories from a departure runway to an arrival runway with intermediate waypoint constraints. Flight plans must be followed to precision for safe and efficient coordination of air traffic, particularly in congested terminal areas and en route corridors. Flight trajectories are optimized to minimize a combination of total time and energy/fuel use [29,34,35]. Hazards such as bad weather introduce regions to avoid. Transiting UASs can also follow optimal 4-D flight plans. UASs may also pursue missions that require area coverage or flight between stationary or moving targets. The tools used for 4-D trajectory optimization can be extended to optimize area coverage or waypoint flight segments [36,37]. Traveling salesman problem solvers can optimize waypoint sequences [38].

Real-time flight planners reactively alter or rebuild plans in response to nearby aircraft, unanticipated obstacles, and changes in mission goals. Aircraft-to-aircraft collision avoidance, known as sense and avoid or detect and avoid [39], is important for manned and unmanned aviation [40,41]. A comprehensive survey and analysis of collision avoidance can be found F. J. M. Campo's 2010 thesis [42]. The traffic collision-avoidance system [43] and automatic dependent surveillance broadcast [44,45] assist with collision avoidance but do not yet provide a complete solution. The mission considered in this paper requires brief low-altitude "hops" in an open-water environment with a low likelihood of encounters with other aircraft, so obstacle avoidance considered in this work is with respect to known (detected or mapped) objects such as rocks, buoys, or slow-moving ships.

Long-duration flight planning has been primarily considered in the context of soaring or powered glider energy-harvesting applications. Gliders can remain aloft long term by periodically climbing at thermal (updraft) waypoints [46] and by optimizing traversal paths over gliding efficiency while additionally orienting any hosted solar panels to maximize energy collection [47,48]. This previous work keeps an aerodynamically efficient aircraft aloft long term, whereas our seaplane trades aerodynamic efficiency for the ability to land and drift on the water. Flying Fish therefore cannot remain aloft long term, instead requiring multiple drift-fly cycles to survive and achieve goals.

Although flight planning is a well-studied problem, long-duration mission planners have been more thoroughly investigated in other domains. For example, space exploration provides fertile territory for long-term autonomous operation because unmanned spacecraft and planetary surface vehicles must execute missions lasting from months to years without the possibility of repair or upgrade. Although mission costs result in a high level of conservatism, onboard planning has been successfully achieved. For example, the Mixed-Initiative Activity Plan Generator (MAPGEN) was infused into day-to-day Mars Exploration Rover operations. The MAPGEN builds daily activity schedules using constrained task planning and scheduling tools, focusing on task-level science and maintenance goals. ASPEN (which stands for automated *scheduling* and *planning* environment) [49], which is a highly successful planner-scheduler for space-based science, employs iterative repair to improve plan quality given highly constrained computational resources.

More recently, the goal-oriented autonomous controller (GOAC) [50] was proposed to demonstrate fully autonomous spacecraft operations. The GOAC incorporates the Teleo-Reactive Executive (T-REX) [51] interleaved planner-executor with correct-by-construction functional modules and a deadlock-centric verification module to manage function interaction timings. T-REX has also been applied to autonomous underwater vehicle applications requiring autonomous sense-deliberate-act cycles for science missions [51]. Long-duration autonomy for space and underwater applications remains an active area of research [52].

III. Energy, Environment, and Aircraft Models

Energy-aware flight planning requires models of energy storage, expenditure, and harvesting, as well as relevant environmental conditions: wind, water motion, and solar insolation. Each model is summarized in the following, followed by a discussion of characterizing these for the Flying Fish platform.

A. Environment Models

Energy collection is computed from locally incident solar-power density [$\mathcal{P}_{\text{spec}}(t, \cdot)$] and the solar array relative incidence angle. The solar incidence angle is resolved in an inertial frame defined by the solar azimuth a_{sun} and elevation e_{sun} , and then it is rotated into vehicle coordinates. The solar position and irradiance models used in this paper are derived from models published by the National Renewable Energy Laboratory (NREL) [53–55].

In the following, we use a dot notation [e.g., $\mathcal{P}_{\text{spec}}(t, \cdot)$] to represent the variables required to characterize incidence, including the Earth-relative position and attitude of the solar array, the Earth's rotation angles and orbital relationship to the sun, and the characteristics of the atmosphere between the array and sun. The NREL model uses the (North-referenced) azimuth and tilt (referenced to the Earth's rotational axis) to represent the solar panel attitude; the solar panel position is characterized by latitude, longitude, and mean-sea-level altitude. Atmospheric parameters are based on local dry-bulb temperature and surface pressure. The NREL model extrapolates relative Earth-sun coordinates from the Gregorian calendar date and the current Coordinated Universal Time (UTC). The current UTC of day is listed as distinct from the dot notation for reference.

The NREL solar position calculator uses a Fourier series to resolve Earth-centered Earth-fixed frame vectors [56]. The solar declination, right ascension, and local mean sidereal time are computed from the local date, time, and vehicle longitude [53]. The vehicle latitude determines the solar zenith, unrefracted solar elevation, and unrefracted azimuth [57] with correction based on the sun's proximity to the horizon, local temperature, and local pressure [58]. Refraction corrections are applied to the azimuth a_{sun} and elevation e_{sun} to extrapolate atmosphere-corrected solar-power density [$\mathcal{P}_{\text{spec}}(t, \cdot)$]. The inertial-frame solar-incidence vector is given by the following:

$$\mathbf{s}_I(t, \cdot) = \begin{bmatrix} \cos(a_{\text{sun}}(t, \cdot)) \cos(e_{\text{sun}}(t, \cdot)) \\ \sin(a_{\text{sun}}(t, \cdot)) \cos(e_{\text{sun}}(t, \cdot)) \\ \sin(e_{\text{sun}}(t, \cdot)) \end{bmatrix} \quad (1)$$

The solar insolation and incidence are input to the vehicle solar-power model in Eq. (2) to compute the expected solar power to be harvested. Figure 1 shows an example of solar energy accumulated over a day based on the solar-incidence angle (elevation only for a horizontal solar array), total incident solar power, and efficiency-scaled solar power that can be harvested by the Flying Fish 1.34 m² encapsulated space-grade gallium-arsenide solar panel.

The greatest sources of uncertainty in the solar model are atmospheric conditions due to the variability in cloud cover. In practice, the Flying Fish can measure and respond to the instantaneous solar conditions measured by the maximum power point tracker. Our planner incorporates ideal solar insolation conditions, leaving real-time estimation of cloud cover as future work. The onboard planner uses a data-driven steady wind estimator based on a weighted running average of air-data system measurements with newer data weighted more heavily. Future winds are estimated from recent wind data. Although Web-enabled forecasts might better inform the system, such data may not be available to a deployed seaplane.

B. Vehicle Energy Dynamics

The multiflight planner must estimate vehicle energy collection, storage, and expenditure during flight and drift operations. Energy collection is a function of solar conditions, solar-collection efficiency, and battery-charge status. Power collection is determined by the solar-incidence angle $\angle \mathcal{S}_A$, incident light spectral power density $\mathcal{P}_{\text{spec}}$, solar array area \mathcal{A}_{sol} , and solar collection efficiency η_{sol} . Energy collected by the solar array \mathcal{E}_{sol} is the time integral of instantaneous power available from the array \mathcal{P}_{sol} :

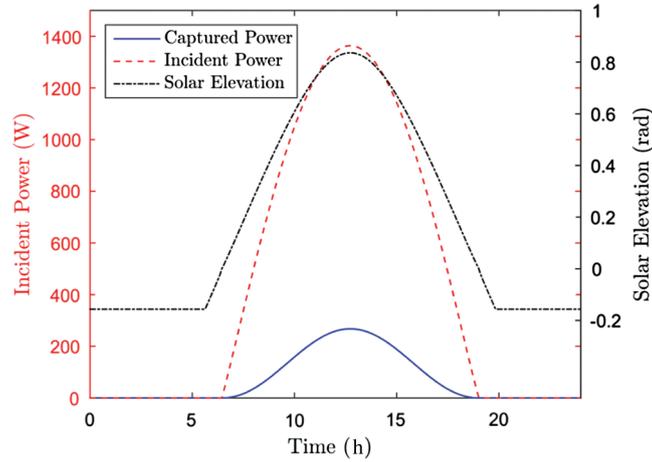


Fig. 1 Daily solar-energy model.

$$\begin{aligned}\mathcal{E}_{\text{sol}}(t, \cdot) &= \int_{t_0}^{t_f} \mathcal{P}_{\text{sol}} dt \\ \mathcal{P}_{\text{sol}}(t, \cdot) &= \eta_{\text{sol}} A_{\text{sol}} \mathcal{P}_{\text{spec}}(t, \cdot) \cos(\angle s_A(t, \cdot))\end{aligned}\quad (2)$$

In Eq. (2), the solar incidence is denoted as the angle of the solar-incidence vector in the solar array coordinate frame s_A . To compute the array-frame incidence vector s_A , the inertial-frame incidence vector s_I is computed using solar motion models rotated by the vehicle's Euler angles plus a pitch-axis rotation of the solar array in the vehicle frame θ_{array} using elementary rotations $R_x(\phi)$, $R_y(\theta)$, and $R_z(\psi)$:

$$s_A(t, \cdot) = R_x(\phi)R_y(\theta + \theta_{\text{array}})R_z(\psi)s_I(t, \cdot)\quad (3)$$

The solar-incidence angle $\angle s_A$ can then be determined based on the inner product of the solar-incidence vector in the array frame s_A and the z axis of the array-fixed frame ($\hat{k} = [0, 0, 1]$). The solar power [Eq. (2)] can then be computed using the incidence vector magnitude $\|s_A\|$ and z component $s_{A,z}$:

$$\cos(\angle s_A) = \frac{s_{A,z}}{\|s_A\|}\quad (4)$$

The solar array area, rotation angles, and solar collection efficiency are known a priori, whereas the vehicle attitude and position are estimated from expected wind and sea state conditions, or based on in situ measurements.

Energy input to the system is subject to the system's ability to accept power; a fully charged battery, for example, cannot accept additional energy, regardless of harvesting capacity. Our initial battery model fit polynomial curves to laboratory-measured charge/discharge trends [1].

Although this approach was accurate for moderate loading conditions, the resulting battery model did not accurately represent the dynamic response during heavy loading, such as during takeoff. A variety of lithium polymer battery models have been developed [59–61], given the growing interest in electric-powered transportation [62,63]. The Flying Fish battery model used in this paper is adapted from an NREL lithium battery model [63]. This model was attributed to the lithium battery manufacturer Saft as a variation on previously known models [59,60]. The NREL-Saft model represents the battery as a parallel resistor-capacitor network with input/output impedance (Fig. 2) and is given by the following:

$$\begin{bmatrix} \dot{V}_{Cb} \\ \dot{V}_{Cc} \end{bmatrix} = \begin{bmatrix} \frac{-1}{C_b(R_e + R_c)} & \frac{1}{C_b(R_e + R_c)} \\ \frac{1}{C_c(R_e + R_c)} & \frac{-1}{C_c(R_e + R_c)} \end{bmatrix} \begin{bmatrix} V_{Cb} \\ V_{Cc} \end{bmatrix} + \begin{bmatrix} \frac{-R_c}{C_b(R_e + R_c)} \\ \frac{-1}{C_c} + \frac{R_c}{C_c(R_e + R_c)} \end{bmatrix} I_s\quad (5)$$

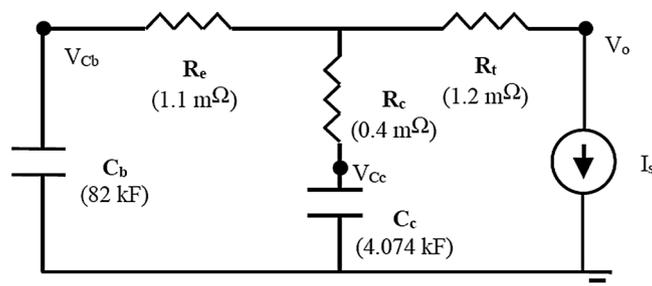


Fig. 2 NREL-Saft battery model [63].

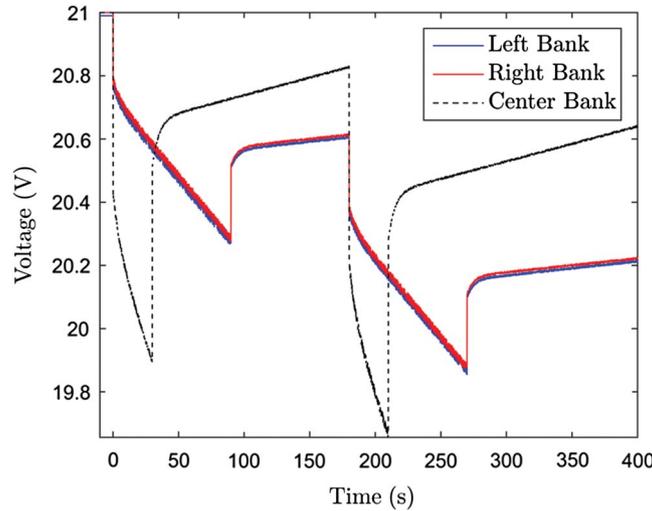


Fig. 3 Battery charge/discharge model: two simulated liftoffs with solar charge.

$$\begin{bmatrix} V_o \end{bmatrix} = \begin{bmatrix} R_c & R_e \\ (R_e + R_c) & (R_e + R_c) \end{bmatrix} \begin{bmatrix} V_{Cb} \\ V_{Cc} \end{bmatrix} - \left[R_t + \frac{R_c R_e}{(R_e + R_c)} \right] \begin{bmatrix} I_s \end{bmatrix} \quad (6)$$

This battery model charge/discharge response is governed by the output impedance, and a capacitor/resistor combination reproduces the nonlinear depletion region under load. The effective battery capacity is represented by the charge stored at a given voltage. Subject to the usable lithium battery voltage range (V_{\min} , V_{\max}), the maximum available energy and the energy remaining at an intermediate voltage $V(t)$ can be written as follows:

$$\mathcal{E}_{\text{batt,max}} = \frac{1}{2} C \cdot V_{\max}^2 - \frac{1}{2} C \cdot V_{\min}^2 \quad (7)$$

$$\mathcal{E}_{\text{batt}}(t) = \frac{1}{2} C \cdot V_{\max}^2 - \frac{1}{2} C \cdot V(t)^2 \quad (8)$$

Each Flying Fish battery pack has five lithium-polymer cells in series, each with a nominal voltage of 3.7 V, an operating range of $(V_{\min}, V_{\max}) = (3.1 \text{ V}, 4.2 \text{ V})$, and an energy capacity of approximately 72 kJ. Battery model parameters [Eq. (6)] have been tuned to deliver the voltage and capacity of a single cell by tuning the impedance and capacitance ($R_e = 1.1 \text{ m}\Omega$, $R_c = 0.4 \text{ m}\Omega$, $R_t = 2.2 \text{ m}\Omega$, $C_b = 18.45 \text{ kF}$, $C_c = 4.0 \text{ kF}$). This model changes the sign of I_s in Eq. (6), which is a suspected error in the original NREL report, as the published model gives increasing voltage under heavy loads. Series battery voltage is simulated by scaling the linear model to the cell count of each battery pack. Battery capacity is simulated by dividing system loads by the number of batteries in each bank. The Flying Fish power system response is shown over two sequential 1.5 min flight cycles with solar charging in Fig. 3. In this simulation, the drift time is brief to illustrate recharge; in practice, a significantly extended drift time would allow batteries to more fully recharge between flights.

Energy expenditures are characterized by two distinct loading processes: avionics (hotel) loads and flight. Avionics loads are assumed to also be present during flight, but the system can “shed” most power loading during drift. Assuming the vehicle must maintain situational awareness on the water, the fixed loads include an avionics computer [avionics central processing unit (ACPU)], an inertial navigation system (INS), wireless communications that can be idled but must monitor command channels, and miscellaneous regulator/interface overhead. The routinely sheddable loads on the water include the control actuation mechanisms (motors, servos, and motor controllers) and the ultrasonic altimeter. Auxiliary payloads are assumed to be “anytime” sheddable for the purposes of survival, although none were modeled in this work.

Table 1 provides an estimate of fixed and sheddable loads. For our purposes, the best case for field-test load shedding is motor shutdown, servo idling, and ultrasonic-altimeter deactivation. This 6 W average avionics power draw requirement over the duration of the mission allows the system to compute maintenance energy expenditures, which are especially critical for forecasting overnight survival. Energy expenditures due to

Table 1 Flying fish fixed and sheddable loads

	$P_{\text{on}}, \text{ W}$	$P_{\text{idle}}, \text{ W}$	$P_{\text{sleep}}, \text{ W}$
<i>Fixed hotel loads</i>			
ACPU	1.0	—	—
Modem	4.83	0.7	0.133
INS	1.2	—	—
Miscellaneous	0.9	—	—
<i>Sheddable loads</i>			
Ultrasonic	1.47	—	0.0
Servo controller	0.5	—	0.0
Control servos (each)	3.15	0.264	0.0
Servo receiver	0.075	—	0.0

flight (propulsive) loads are modeled from flight- and laboratory-derived tests. Second-order polynomial fits ($0.75 < R^2 < 0.8$) relate throttle settings to power requirements for main and boost motors. The curves are applied to the throttle vector over a flight profile to determine system loads and then, through the battery model, to compute cumulative energy expenditures. Throttle settings for each segment of flight, takeoff, climb, cruise, and descent have been extracted from flight data and are used to estimate the power required for each segment. The first-generation model assumed that the straight-line cruise segment of each Dubins path trajectory served as a reasonable average flight direction for that segment [1]. A solution of the wind-heading velocity triangle was used to determine the slipping-flight speed along the flight path, which divided the entire turn-fly-turn segment length to produce the segment flight time. A wind-aware bank-to-turn unicycle model was used to build the turning flight-path segments. Once an estimate of flight time in each steady flight segment is known, energy required per segment can be computed and then summed over the entire flight. The difference in the Dubins trajectory length between turning at a waypoint and turning before a waypoint was assumed negligible. We also assumed the transitions between steady flight states (e.g., to enter/exit a banked turn) were negligible.

C. Flying-Fish Model Characterization

A series of laboratory, drift, and flight tests were conducted to develop vehicle energy use and performance models. At deployment, the battery capacity was 3240 kJ, and the solar collection area was 1.3 m² using 352 28%-efficient gallium-arsenide solar cells. Although accurate characterization of the three-dimensional sea surface motion was beyond the scope of this project, inertial and air-data system data collection over drift/flight sequences indicated that drift direction was primarily in the direction of wind; drift speed was, on average, 3.5% of wind speed; and wind-induced drift dominated current-induced drift. These observations provided a simple wind-based drift model for the Flying Fish planner.

Figure 4 shows propeller efficiency versus airspeed (freestream velocity) based on benchtop primary left/right and boost motor tests. The measured power P is scaled by the square of the propeller diameter d to eliminate the length dimension from the presented trends. The figure illustrates that the lower-power boost motor has higher low-air-speed efficiency, whereas the main motors have higher power output with comparable efficiency at cruise speeds for which the boost motor will be off. These curves plus hydrodynamic drag provide a modeling basis for the high motor power draws repeatedly observed during takeoff. Figure 5 shows GPS data from an example watch-circle crossing flight test. For each POS flight, the Flying Fish accelerates as quickly as possible to liftoff speed, flies upwind at low altitude across the watch circle, and lands near the upwind boundary. The baseline POS mission calls for repeated upwind flight/downwind drift cycles across a fixed watch circle until energy is depleted or the mission ends.

Figure 6 shows the autonomous flight of the Flying Fish across a small-scale POS watch circle at Douglas Lake in Michigan. Figure 6a illustrates the pitch dynamics denoted by the phase of flight: takeoff initiation (TI), on step, liftoff, cruise, descent, and completed landing. The slow-period pitch oscillation before TI represents free drift of the vehicle induced by the water surface.

The high pitch rates observed just before liftoff represent the vehicle accelerating through surface chop, with similar behavior observed just after touchdown. The airspeed time history in Fig. 6b illustrates full-thrust acceleration to liftoff, an accelerated climb to cruise, and rapid deceleration due to hydrodynamic drag encountered after touchdown. Motor thrust levels (Fig. 6b) are set based on the phase of flight. Flight mode switches are triggered by a combination of airspeed, altitude, and GPS position relative to the watch-circle center and upwind landing waypoints.

IV. Mission Planner

The multiflight planner constructs sequences of flight and drift segments that visit surveillance waypoints in the air and on the ground, and that satisfy energy and performance constraints given current and expected environmental conditions. This section defines the planner domain including mission goals and constraints, utility, and cost metrics. An example watch-circle planning environment is shown in Fig. 7.

The planner uses a relative Cartesian frame (+y North, +x East) with the origin at the latitude and longitude of the watch-circle center. Planning goals and constraints are generated randomly, and the same randomly generated environment is used for consistency in compared results. Planner goals and constraints (obstacles) are represented as augmented waypoints with attributes of type (TYP), time, position, velocity, attitude, dimensions, execution priority, initial goal value, and description (DES), respectively:

$$\mathbb{P}_i = \{\text{TYP}, t, (x, y, z), (\dot{x}, \dot{y}, \dot{z}), (\phi, \theta, \psi)(r, h), \bar{p}, v_0, \text{DES}\} \quad (9)$$

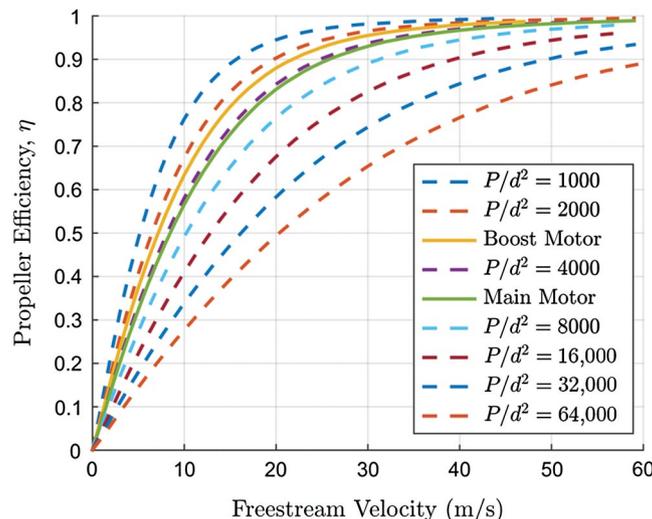


Fig. 4 Propeller efficiency vs velocity.

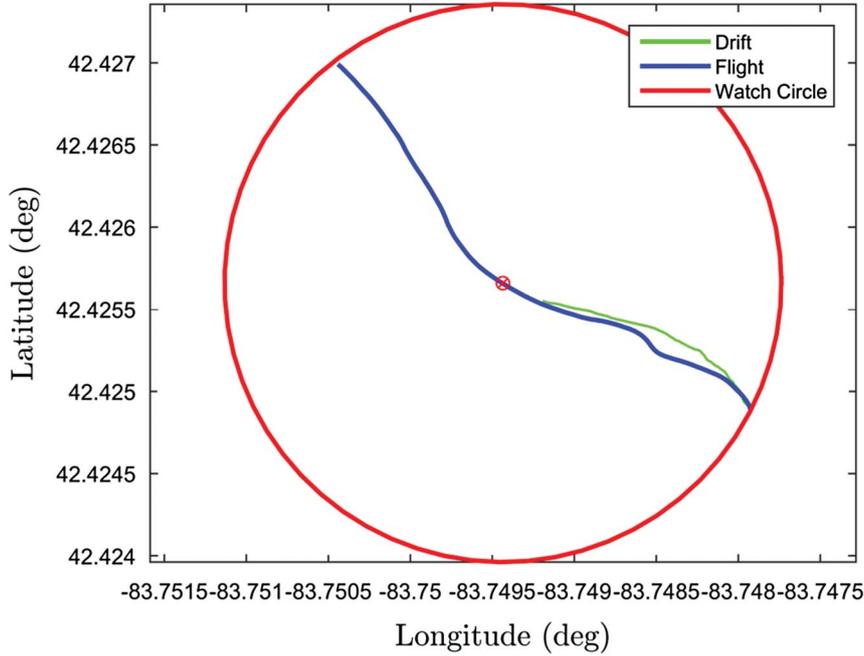


Fig. 5 Example watch-circle crossing.

Four environmental constraint types are defined: hard obstacles \mathcal{C}_{OH} , soft obstacles \mathcal{C}_{OS} , hard operational boundaries \mathcal{C}_{BH} , and soft operational boundaries \mathcal{C}_{BS} . Hard and soft constraints are inviolable and advisory barriers, respectively. This specification is included for completeness, but only hard constraints are used in our case studies. The velocity attribute supports moving constraints, e.g., a boat or aircraft. All obstacle and boundary constraints are modeled by finite vertical cylinders, allowing the Flying Fish to laterally or vertically circumvent obstacles. An example constraint environment is presented in Table 2 and shown in Fig. 7. Two types of mission goals are defined: surface ω_S , and aerial ω_A . Goals are considered satisfied when the vehicle passes within a specified tolerance of the waypoint. A single goal is assigned value and priority. The total value v_i of the i th goal is the sum of an initial goal value $v_{0,i}$ and a weighted time since last visit $v_{t,i}$ to encourage revisits:

$$v_i = v_{0,i} + v_{t,i} \cdot \Delta t \quad (10)$$

where $v_{0,i}$ are zeroed when goal satisfaction occurs. For cases where goals must be visited exactly once, $v_{0,i}$ can be set to unity and $v_{t,i}$ set to zero. Priorities are assigned values over $[0,1]$, with higher values indicating higher priority. An example goal set is given in Table 3.

A Flying Fish mission is planned over zero or more surface goals $\omega_{S,j}$ and zero or more airborne goals $\omega_{A,i}$ subject to vehicle and environmental constraints \mathcal{C} . A watch-circle boundary is defined to prevent unbounded drift once goals are satisfied. The planning problem for a single mission is given by

$$\mathbb{P} = \{\mathbb{P}_1, \dots, \mathbb{P}_n\} \quad \text{given } \{\mathcal{C}_{OH}, \mathcal{C}_{OS}, \mathcal{C}_{BH}, \mathcal{C}_{BS}, \omega_A, \omega_S\} \quad \text{such that } \{\mathcal{C}_{BS} \cup \mathcal{C}_{BH} \cup \omega_A \cup \omega_S\} \neq \emptyset \quad (11)$$

Figure 8 presents two variations of an example mission to explore a surface phenomenon (e.g., an algae bloom). In Fig. 8a, the system is directed to goals around the perimeter on successive flights before landing upwind and drifting back through the surface region of interest (ROI). In Fig. 8b, the ROI is a hard obstacle that requires in-flight exploration but does not safely support return drift through the region, e.g., due to debris.

We use the models presented previously for flight and environment simulations. Dubins paths are constructed between any two airborne waypoints with specified headings. Drift simulations indicate low-energy expenditure with positive overall energy balance during periods of modest to high solar insolation (Fig. 1). The greatest impact of drift on solar-energy collection is that high drift speeds will reduce total energy-harvesting time.

Planner actions include flight to an airborne goal, flight to a surface goal, and drift. The drift or no-operation (no-op) action is feasible for every surface location not in violation of a constraint. All airborne actions require propulsion system use that expends more energy than is collected. Each planned action initiates flight to a goal aerial waypoint, landing at a surface waypoint, or flight to a waypoint at the upwind/updrift boundary. This “fly-to-boundary” action repositions the aircraft at the location maximizing free drift time within the operating region. Available flight actions support transit to 1) another aerial goal, 2) a surface goal, or 3) the updrift boundary (surface) goal. Similarly, available actions from any surface state include 1) flight to an aerial goal, 2) water surface taxi to a surface goal, 3) flight to the updrift boundary location (direct repositioning), and 4) free drift (no-op). An example action sequence is presented in Fig. 9.

The actual cost Ξ for each action reflects energy expenditure and action completion time. Energy expenditure $\mathcal{E}_{\text{cost},i}$ for the i th action of the linear mission plan action sequence is given by

$$\Xi_i = \mathcal{E}_{\text{cost},i} \quad (12)$$

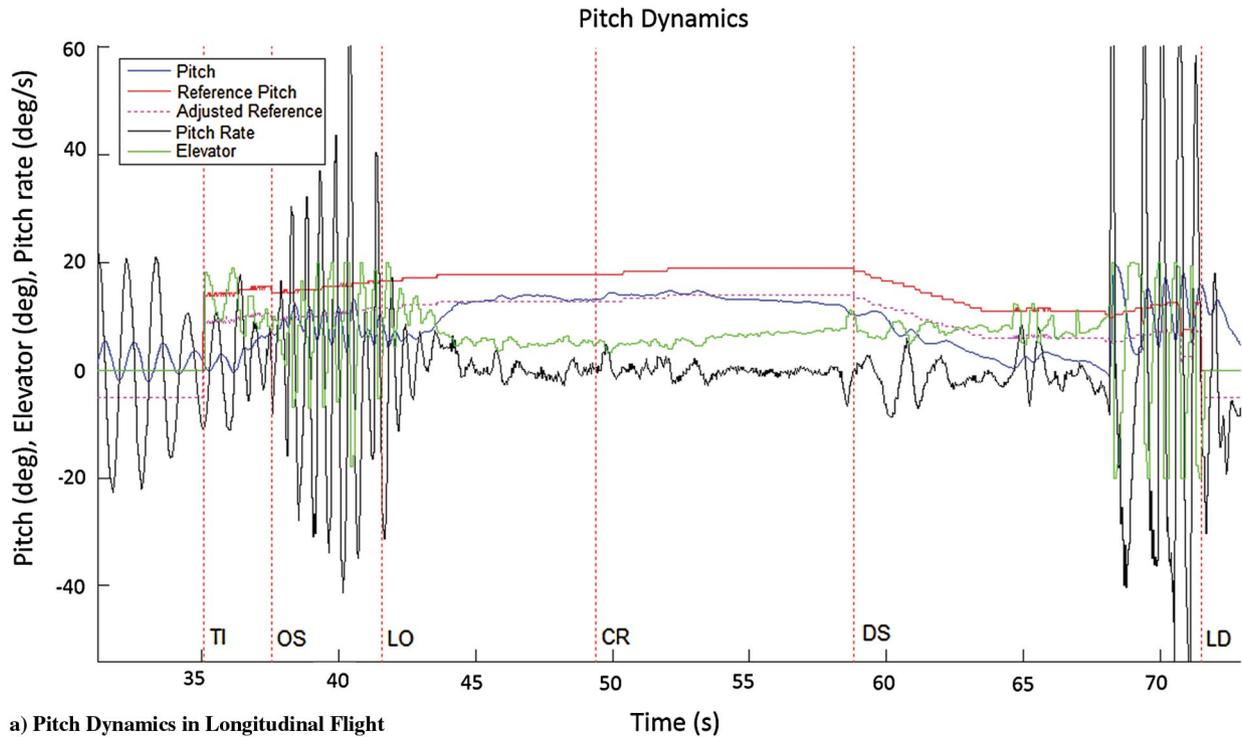
A positive benefit Γ term is also included to reward energy harvesting or recovery $\mathcal{E}_{\text{rcvr},i}$ over the i th action, which is nonzero only for free drift (no-op):

$$\Gamma_i = \min(\mathcal{E}_{rcvr,i}, (\mathcal{E}_{\max} - \mathcal{E}_{sys,i-1})) \quad (13)$$

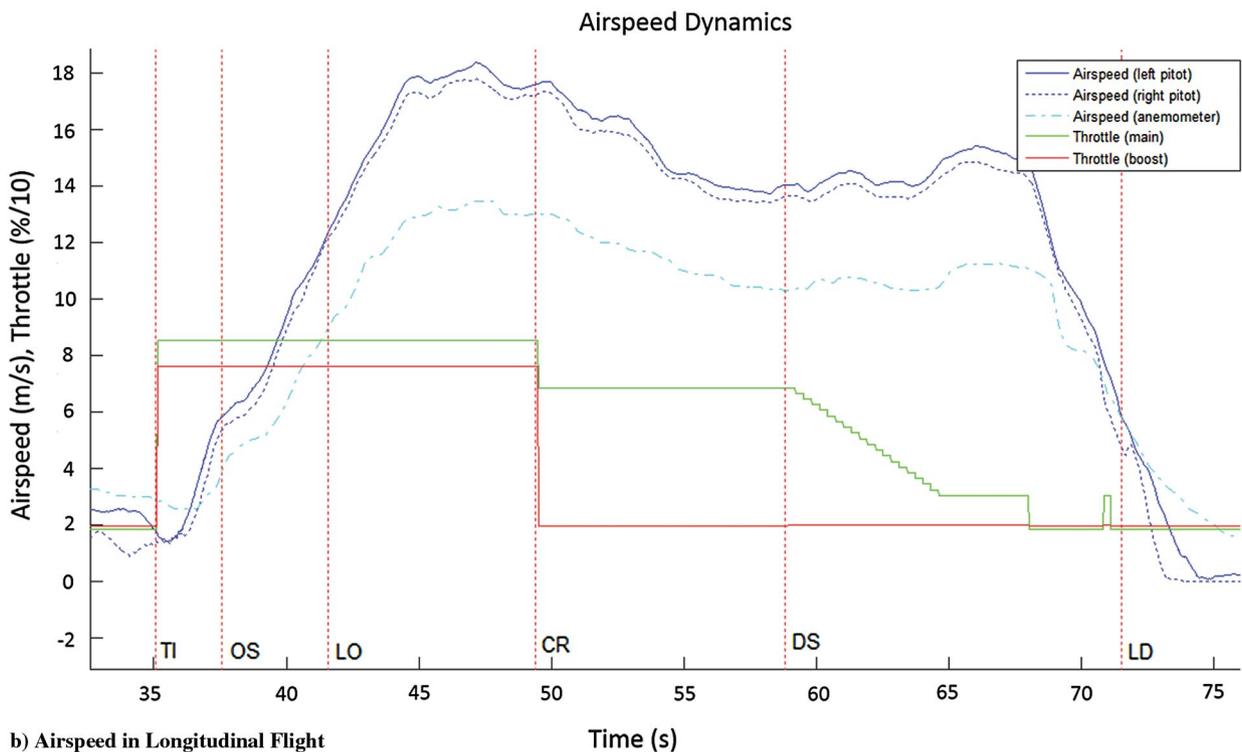
where \mathcal{E}_{\max} is the maximum battery energy storage capacity, and $\mathcal{E}_{sys,i-1}$ is the battery energy stored at the end of the $(i-1)$ th action or, equivalently, the beginning of the i th action. This formulation captures the constraint that batteries can only charge to capacity \mathcal{E}_{\max} . The third metric is the goal achievement value Φ . The value of the i th goal-seeking action is defined as $\Phi_i = v_i$ from Eq. (10). Free drift and updrift boundary repositioning actions are assigned a value of $\Phi = 0$. Finally, Utility Y represents the weighted sum of the cost, benefit, and value:

$$Y_i = \tau_{\Gamma} \cdot \Gamma_i + \tau_{\Phi} \cdot \Phi_i + \tau_{\Xi} \cdot \Xi_i \quad (14)$$

Weighting terms $\{\tau_{\Phi}, \tau_{\Gamma}, \tau_{\Xi}\}$ are given in each case study.



a) Pitch Dynamics in Longitudinal Flight



b) Airspeed in Longitudinal Flight

Fig. 6 Watch-circle crossing flight data.

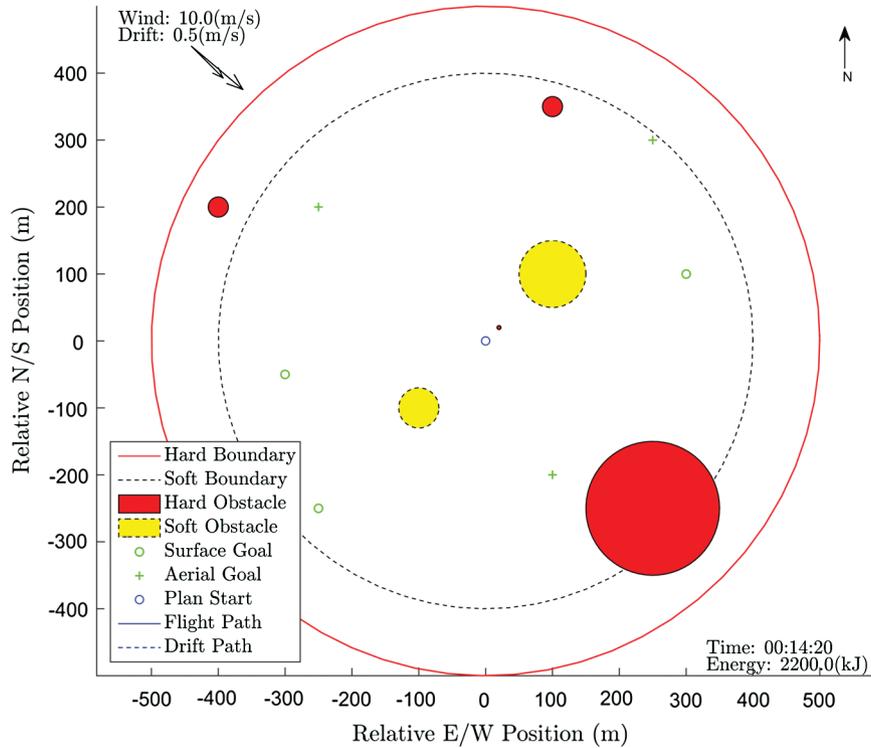


Fig. 7 Planning environment.

A. Multiflight Planning as a TSP Extension

A POS mission is successful if the platform visits every goal waypoint at least once and remains in the designated watch circle during the planning horizon without violating energy or collision constraints. For the seaplane, multiple flights are required in cases where waypoints cannot all be visited during a single flight and in cases where at least two repositioning flights are required to correct for drift motion over the planning horizon. This multiflight planning problem can be loosely mapped to the NP-hard traveling salesman problem [64]. The TSP finds an optimal (shortest distance) “tour” to visit every goal waypoint exactly once [64–66]. The Flying Fish must visit every waypoint from a given starting position, which is a form of TSP known as the traveling salesman path problem. An exact TSP solution can be found by exploration of all possible goal orderings; however, exhaustive search is impractical. Numerous TSP solvers have been developed, ranging from dynamic programming with complexities of $O(n^2 2^n)$ [67] or better [68] given n goals to more recent distributed algorithms such as ant colony optimization [69].

The standard TSP assumes transition graph costs are symmetric and all action transitions have positive cost corresponding to distance traversed or energy consumed. TSP solvers rely on the triangle inequality, which states that the cost over two successive transitions to a specific terminal

Table 2 Planner environment description

Type	Position, m	Velocity, m/s	Dimension, m	Description	
$C_{BH,1}$	Hard boundary	(0,0,0)	(0,0,0)	(500,100)	“Shore line”
$C_{BS,1}$	Soft boundary	(0,0,0)	(0,0,0)	(400,500)	“Advised airspace”
$C_{OH,1}$	Hard obstacle	(-100,200,0)	(-10,20,0)	(15,15)	“Boat”
$C_{OH,2}$	Hard obstacle	(20,20,0)	(0,0,0)	(3,4)	“Buoy”
$C_{OH,3}$	Hard obstacle	(250,-300,0)	(0,0,0)	(150,5)	“Reef”
$C_{OS,1}$	Soft obstacle	(100,-200,0)	(1,-1,0)	(50,0)	“Oil slick”
$C_{OS,2}$	Soft obstacle	(-100,-100,0)	(0,-1,0)	(30,0)	“Algae bloom”

Table 3 Planner mission description

Type	Position, m	Priority	Value	Description	
$\omega_{S,1}$	Surface	(-250,-250,0)	8	20	“Water sample”
$\omega_{S,2}$	Surface	(-300,-50,0)	7	10	Water sample
$\omega_{S,3}$	Surface	(300,100,0)	8	20	“Remotely-Operated Vehicle (ROV) communications”
$\omega_{A,1}$	Airborne	(250,300,20)	8	10	“Algae bloom image”
$\omega_{A,2}$	Airborne	(-250,200,20)	7	10	“Unknown object”
$\omega_{A,3}$	Airborne	(100,-200,20)	8	10	“Oil slick image”

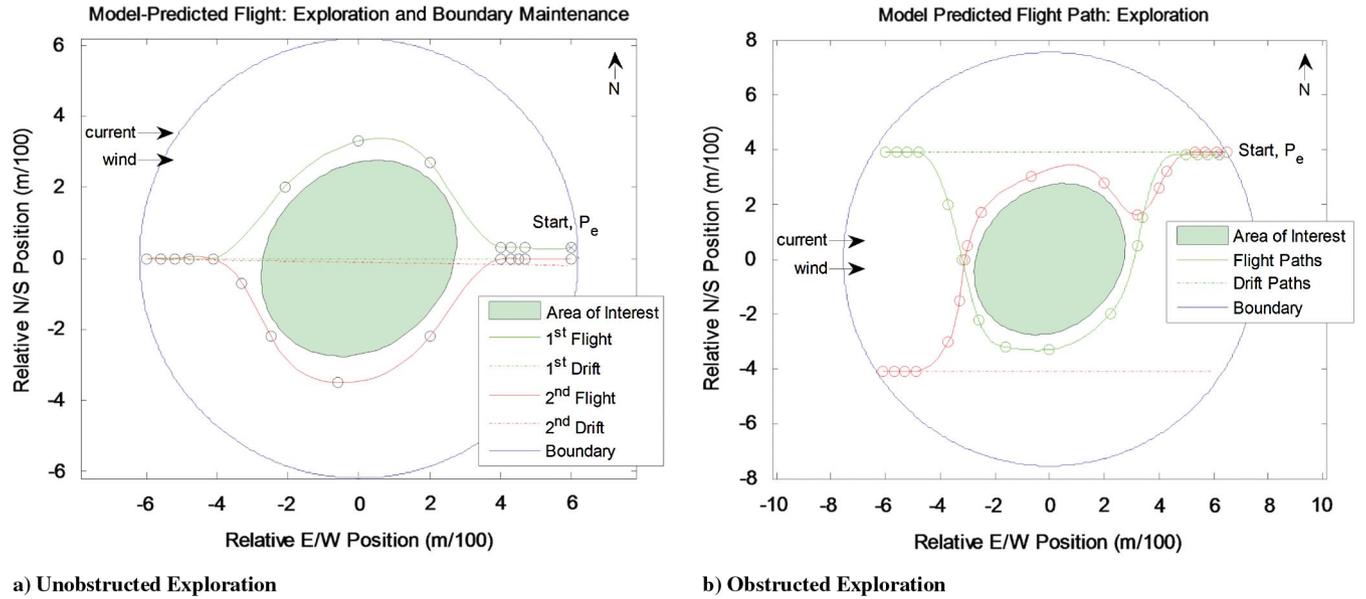


Fig. 8 Example missions.

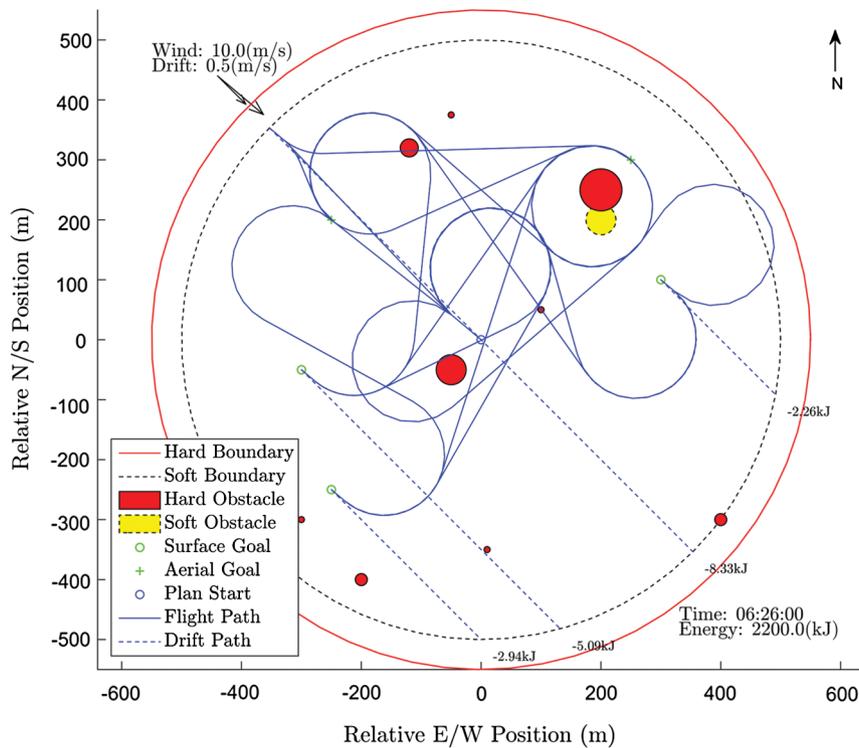


Fig. 9 Expansion of all possible mission trajectories between goal points.

node must be equal to or greater than the cost that would be incurred traveling directly to the terminal node. TSPs that obey the triangle inequality are denoted as metric TSPs.

Due to energy harvesting and wind, the Flying Fish planner cannot guarantee any of these TSP assumptions. First, the search problem is asymmetric because the cost of each flight segment and the requirement for takeoff and landing to occur at an upwind heading yield transitions with different traversal costs in one direction versus the other. Algorithms to convert asymmetric graphs to symmetric graphs may not be practical with large problem spaces [70]. The two non-goal-seeking actions of drift and flight to the upwind boundary can be beneficially selected in most states, which substantially grows search-space size. Consider the search space defined by n waypoint surveillance goals, no obstacle constraints, and actions to land at the updrift boundary (available from any flight state) or drift (available from any surface state). Because at least one of the non-goal-seeking actions is executable from every state, it is not possible to define a fixed search depth related to the number of waypoint goals. Instead, it may be beneficial to execute one or more (drift-fly) non-goal-seeking actions to maintain needed energy reserves. This attribute results

in the possibility of a search-space depth that well exceeds n , which theoretically could be infinite, given a plan requirement to achieve all mission goals and adverse wind conditions.

The Flying Fish is assumed to initially be drifting in the watch circle with user-specified energy reserves. The branching factor for this top node is $(n + 2)$, representing the option to visit any surface or airborne surveillance goal, continuing to drift, or flying directly to the updrift boundary point. Given value in repeat visits to surveillance goals (nonzero $v_{t,i}$), this worst-case branching factor $(n + 2)$ will persist at all search tree depths. Given no value in repeat goal visits ($v_{t,i} = 0$), the branching factor will decrease by one along a particular path each time a goal is visited.

Solar-energy harvesting presents a planning challenge. First, the triangle inequality may not hold in cases where extra actions (free drift segments) are able to actually reduce overall plan cost through energy recovery. In fact, the plan cost may be dominated by external factors such as solar-energy availability and wind rather than the physical relationship (distance) between goal waypoints. The Flying Fish mission planning problem is therefore a nonmetric TSP. Also, because drift segments typically harvest more energy than used, these graph edges have a negative overall cost. Negative edge costs can give rise to negative-cost (drift-fly) cycles within the search space, which are bounded only by maximum battery energy storage constraint \mathcal{E}_{\max} factored into benefit function Γ in Eq. (13).

Given the aforementioned extensions to the TSP, we refer to the solar-regenerative Flying Fish multiflight planning problem as the frequent flier salesman problem (FFSP). The FFSP is defined to have the same goal as the repeat-visit TSP but, in the FFSP, frequent flier “miles” are assumed to be bounded above by the search horizon depth or time constraints and from below by the salesman’s desire to not be stranded ($\mathcal{E}_{\text{sys},i} > 0$). Over a given subplan action sequence, the salesman must expend a relatively large number of miles to reach goals while accruing a sufficient number of new miles (energy) to maintain minimum reserves. Some energy is collected during flight segments, but more energy is expended during flight. Free drift enables energy accrual with almost no cost, at the cost of time delay in goal achievement and the need for updrift repositioning flights to remain in the desired watch circle. Suppose, at the beginning of a mission, that the salesman (Flying Fish) starts with a positive $0 < \mathcal{E}_{\text{sys},0} < \mathcal{E}_{\max}$ stored energy level. The total energy cost plus benefit ($\Xi + \Gamma$) of any path through the search tree is bounded by $(\mathcal{E}_{\text{sys},0} - \mathcal{E}_{\max}) < \Xi + \Gamma < \mathcal{E}_{\text{sys},0}$. So long as the goal achievement is weighted nontrivially in Υ , the planner will favor paths for which goal-seeking actions are inserted as often as possible until finding a path for which all goals have been achieved (given $v_{t,i} = 0$) or after exploring all paths within the given time or depth search horizon.

B. Search Strategy and Heuristics

For this initial investigation of POS multiflight planning, we adopt a traditional forward-chaining tree search strategy. Case studies rely on actual cost and heuristics estimating the cost to go. As will be described in the following, a greedy search over a simple heuristic supports the longest planning horizon with manageable complexity but returns suboptimal results. The uniform-cost (Dijkstra’s [4]) search over the total weighted utility Υ guarantees an optimal solution at the “cost” of high memory and time search complexity. A^* search requires an informed heuristic to realize appreciable search efficiency gains relative to uniform cost. Because POS can extend indefinitely, we impose a search horizon based on a fixed search depth limit or total mission plan time horizon in all case studies.

Both the uniform-cost and A^* searches adopt Υ with user-specified weights as the actual cost from the initial state to the current node. The greedy search can support the anytime search by quickly generating a suboptimal baseline plan and provides a method for generating plans over long mission time horizons. We use a simple one-step-lookahead heuristic for greedy search case studies that selects at each node either a goal-achieving action that immediately achieves a visitation goal (increasing Φ) or offers energy harvesting Γ via drift. Relative weighting of these two heuristic terms enables the user to tune whether a plan aggressively pursues goals versus conservatively drifting when possible to maximize energy reserves.

As discussed previously, a straight-line distance heuristic is insufficient for multigoal planning. We define an initial A^* search heuristic based on analyzing the set of unvisited goals to find the single shortest path between any two p_{\min} and multiplying this by the number of unvisited goals. Given n unvisited goals, the power in kilowatts required for trimmed cruise, and the maximum airspeed V_{\max} , the heuristic value in kilojoules is given by the following:

$$\Theta_0 = \frac{(n-1)p_{\min}}{V_{\max}} \mathcal{P}_{\text{cruise}} \quad (15)$$

This heuristic represents the absolute minimum energy required to sequentially fly to remaining waypoint goals that could be surveyed from the air but ignoring takeoffs, landings, and surface goals. Unfortunately, the set of unvisited goals is similar for each early node, resulting in heuristic values that provide little distinction in cost-to-go values. To offer a more informative heuristic, we formulated an “inner-loop” approximate TSP that finds the minimum geometric distance between all remaining unvisited goals. Although we believe this TSP heuristic is novel, particularly for flight planning, others have explored nested search problems related to the TSP. For example, manipulator path planning [71] over multiple goals in a constrained environment requires that the goals be efficiently ordered (e.g., the classic TSP), yet traversal costs are dependent on which traversal paths and inverse kinematic solutions are selected. Negative, nonmetric costs are distinct to our multiflight planning application, whereas the dual complexity of multilink collision-free path planning and the TSP is distinct to the manipulator application.

Our inner-loop TSP heuristic represents a minimum-geometric-distance TSP: the search graph is symmetric, adheres to the triangle inequality, and has fixed edge lengths that can be computed and stored a priori. To solve this inner-loop TSP, a recursive exhaustive solver is implemented as a heuristic for the A^* search. This TSP solver scales, for n goals, by the number of available paths p through all goals. For a goal action, the worst-case time complexity is $O((n-1)!)$ versus $O(n!)$ for a no-op (drift) action. The recursive search is structured to minimize the number of computations required over all paths by proceeding to maximum depth first and working backward over permutations of each path. It may be possible to obtain additional computational efficiency, at the cost of memory efficiency, by storing the first calculation of each graph edge cost to eliminate duplication. This heuristic $\Theta_{k,i}$ represents the shortest three-dimensional (3-D) path connecting every remaining unvisited goal p_{\min}^* . $\Theta_{k,i}$ requires cruise airspeed V_{\max} , cruise power, and steady wind speed $\|\mathbf{w}_I\|$:

$$\Theta_{k,i} = \frac{p_{\min}^*}{V_{\text{cruise}} + \|\mathbf{w}_I\|} \mathcal{P}_{\text{cruise}} \quad (16)$$

V. Case Study Results

This section presents a series of FFSP case studies using greedy, uniform-cost, and A^* search strategies. Short-term and long-term mission plan time horizons are considered, including an overnight mission where no energy harvesting is available, a midday maximum solar insolation mission, and a mission from dawn to dusk. In each case, the proposed search strategy, cost/heuristic metrics, and constraints are evaluated.

A. Greedy Search Case Study

The greedy FFSP algorithm ordered the search space based on one-step lookahead as discussed previously. The greedy search was applied to planning problems of increasing complexity, including boundary maintenance, boundary maintenance plus surface goal exploration, and boundary maintenance plus surface and aerial goals. The greedy search planner was able to find solutions with multiple-day planning horizon problems due to its straightforward search-space ordering favoring mission completion. The greedy search was used to find overnight survival plans based on wind and solar insolation data for March 2011 collected from a monitoring buoy in Douglas Lake near Pellston, Michigan.

The greedy search finds overnight-survivable plans for the baseline watch-circle POS mission and extended goal exploration missions but fails to find a valid plan after sunset, given sufficiently high goal revisit weightings. Greedy search cases support two conclusions. First, valuing overnight goal exploration without a hard constraint on long-term energy use is dangerous from the perspective of survival because, in some cases, overnight energy was depleted to dangerously low levels to achieve goals overnight. Second, it is difficult to develop a utility weighting choice to assure the greedy search planner will make appropriate mission vs survival trades; some daytime flight cases with high goal values resulted in energy depletion, given low solar insolation due to high incidence angles or cloud cover.

B. Midnight Missions: Optimal Planning Baseline

Given the nonmonotonic total energy/cost sequence in the FFSP with energy harvesting, we first consider the non-energy-harvesting case that occurs for the Flying Fish between sunset and sunrise. When no solar energy is collected, negative action costs are eliminated. In this case, the FFSP reduces to the nonmetric TSP. For this case study, a set of single-visit goals are specified but they are not given any value in overall utility to recognize the difficulty in trading off large sets of goals with survival. The action benefit Γ is also zero because no solar energy can be harvested overnight. Under these conditions, the planner will find energy-optimal solutions that visit each goal once with the total utility function Υ at level $k + 1$ defined by the following

$$\Upsilon_i = \Xi_i = \mathcal{E}_{\text{cost},i}, \quad i = 1, 2, 3, \dots \quad \Upsilon = \sum_1^k \Upsilon_i \quad (17)$$

A uniform cost search or Dykstra's search expands the action with the lowest cumulative cost (highest cumulative utility Υ) from the root node. The uniform cost is guaranteed to converge to a globally optimal solution if every action cost (utility) is bounded from below (above). In the nonregenerative case, the Flying Fish planner is always subject to a "hotel" load. A resulting plan and energy budget from a "midnight" simulation is presented in Fig. 10.

The three cases where the flight path crosses surface constraints are waypoint goals the system flies over with sufficient altitude separation (see Sec. IV). In this case, no solar energy is available but because drift periods with on hotel load enabled identification of more direct routes. Drift actions have a cost based on their minimal (hotel) power load and can provide a means of low-cost repositioning to a downwind watch circle station.

With a 2 h search horizon, the uniform cost search converges to the optimal solution in ~ 315 s on a dedicated desktop server (Intel Xeon X3450, quadcore CPU at 2.67 GHz and 8 GB of RAM), which is more powerful than the phase 2 Flying Fish embedded computer [TI open multimedia applications platform (OMAP) single-core CPU at 600MHz, 256 MB of RAM], but that is certainly feasible for future unmanned seaplanes. A uniform-cost search is an optimal strategy but is also computationally complex, in that it can explore a substantial part of the exhaustive search space. Given that mission planning horizons will typically be longer in practice, focus shifted to developing an efficient cost-to-go heuristic for A^* search with a total utility of

$$\Upsilon_k = \sum_{i=0}^n \Xi_{k,i} + \Theta_{k,i} \quad (18)$$

where the heuristic $\Theta_{k,i}$ was defined in Eq. (16). Application of this heuristic is effective. The results of the A^* search applied to the "midnight flight" case duplicate the optimal uniform-cost search solution (Fig. 10), but the A^* search solution is found in approximately one-third of the computational time. The inner-loop TSP heuristic effectively calculates the amount of energy required to fly the minimum 3-D distance through

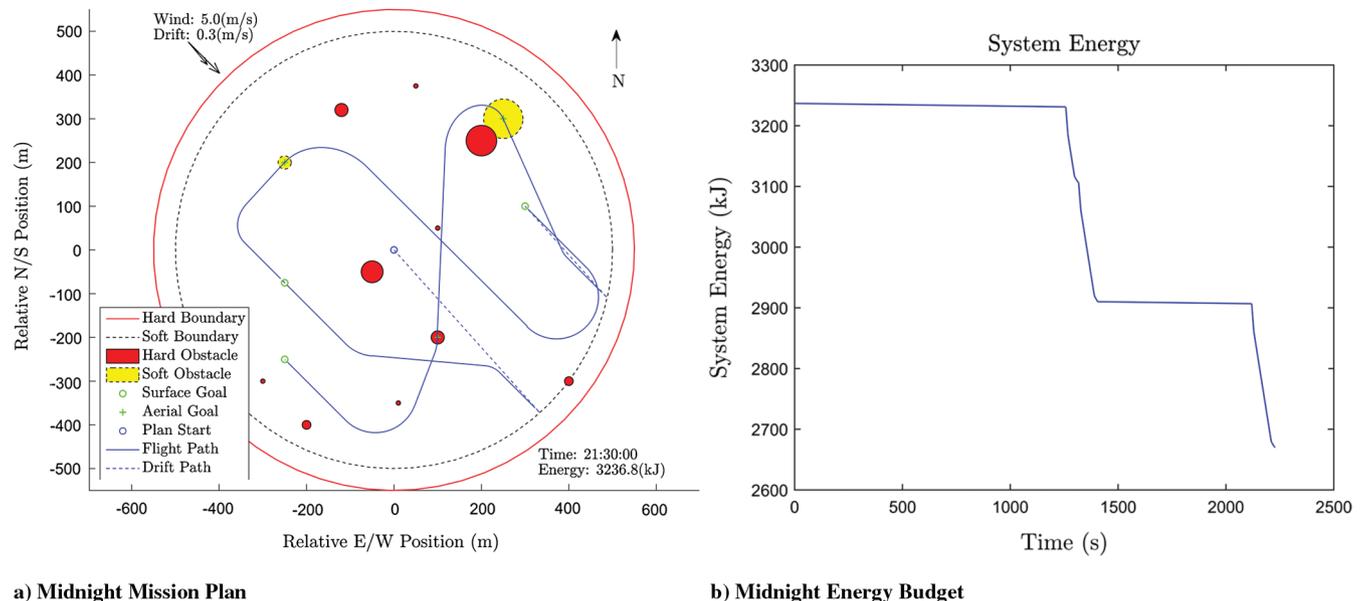
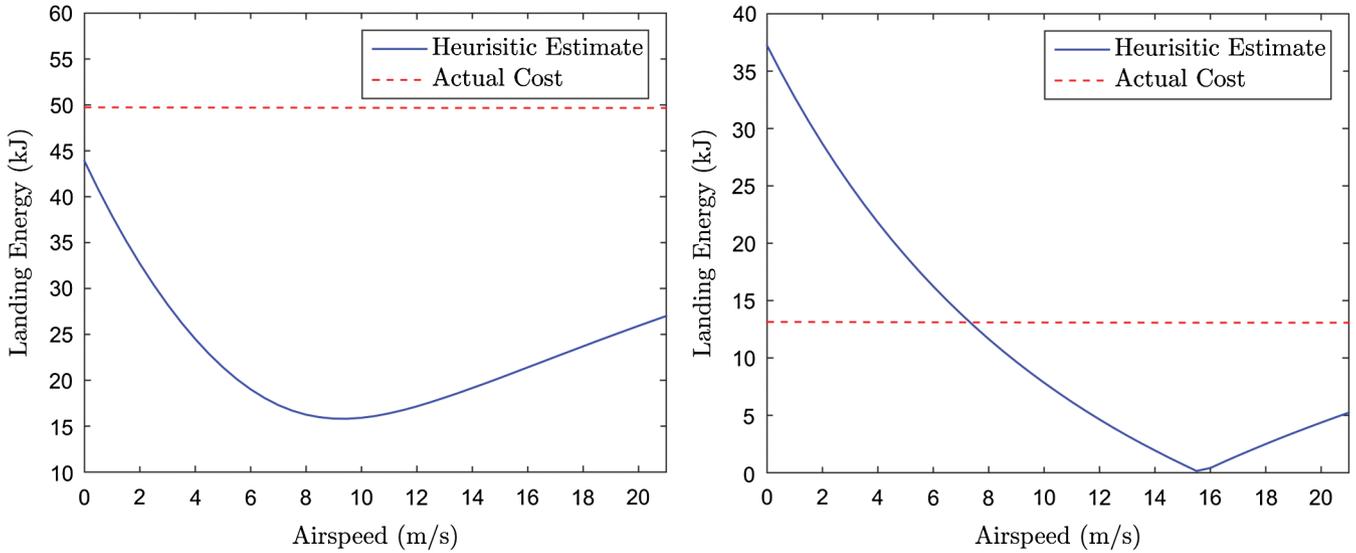


Fig. 10 Midnight uniform-cost search planning.



a) Landing from Downwind Trajectory

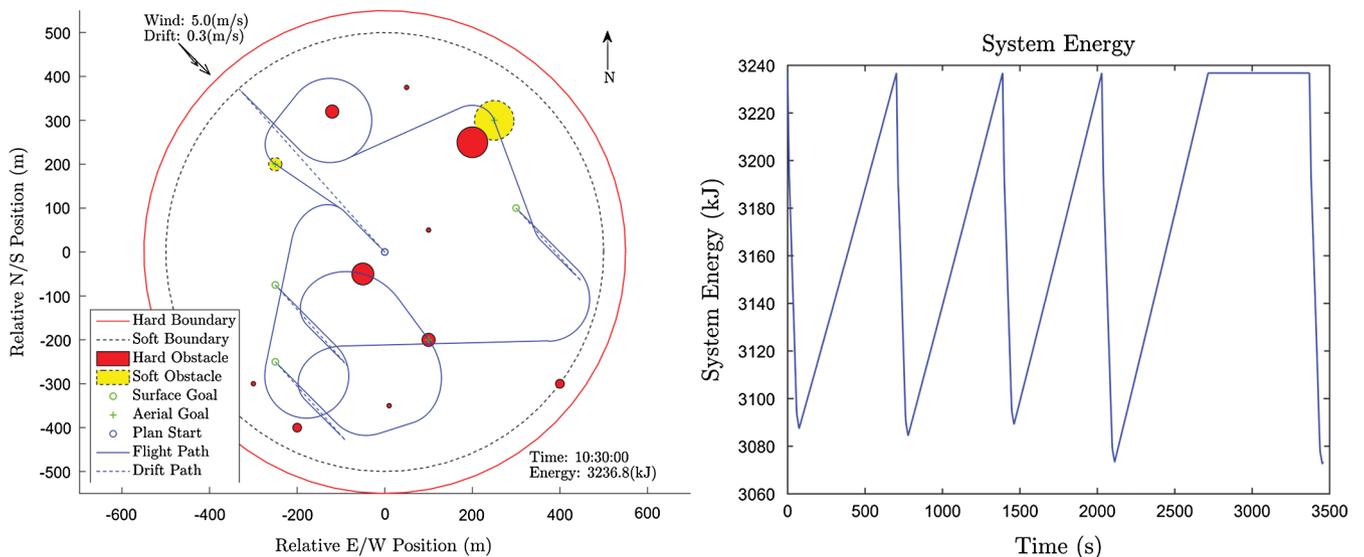
b) Landing from Upwind Trajectory

Fig. 11 Comparison of actual cost and heuristic estimates of landing.

every remaining goal as if the goals were in a straight downwind line. Given that crosswind and upwind flights are less efficient, the heuristic will always underestimate the actual cost of flight through any arbitrary set of goals that are not aligned in a straight downwind line and will exactly estimate the cost of goals that are in a straight downwind line. The heuristic will estimate the exact cost for flight in any straight-line sequence in the absence of wind. Furthermore, the heuristic will underestimate takeoff by a larger margin because surface acceleration, liftoff, and climb require significantly more power than cruise over any given distance. The only transition that is not underestimated is landing, for which large reserves of kinetic and potential energy, rather than battery energy, are traded for glide distance. In the limiting case, estimating the cost over a set of goals in a straight downwind line, the fact that our final-approach constraints require the vehicle to turn into the wind before descending results in higher-energy requirements for landing than are estimated by the heuristic, regardless of wind speed (Fig. 11a). However, a new limiting case must now be considered; although every nonlanding upwind transition will be underestimated, the same is not necessarily true for landing transitions over every possible wind speed. Also note that, in this work, our turn radius is sufficiently conservative, so we presume the plane can bank less or more as needed to track the planned constant-radius path segments in the wind.

Figure 11b shows that, for landings with wind speeds less than ~ 7.3 m/s, the heuristic overestimates the cost to reach the landing site due to the reduced thrust during descent. If landing is not the final action, a suboptimal solution might be found in this case. If there is a comparable number of takeoffs and direct-downwind landings with low wind, the underestimated takeoff cost may be sufficient to cancel the overestimated landing cost. In summary, the proposed heuristic has been demonstrated to match or underestimate the actual path cost, except in the following cases given winds less than ~ 7.3 m/s:

- 1) Exactly one transition remains, and it is a direct-downwind landing.
- 2) The last transition is a direct-downwind landing not preceded by a sufficient combination of takeoffs and up/crosswind cruise. Furthermore, the costs of direct-downwind landings executed in the middle of a plan will also be overestimated, but cost is underestimated for any subsequent



a) Midday Mission Plan

b) Midday Energy Budget

Fig. 12 Midday uniform-cost planning.

takeoff. Collected empirical evidence suggests that these inadmissible cases are possible but rare; solution costs with our heuristic were comparable to uniform-cost search counterparts in all simulated cases.

C. Midday Missions: The Energy-Harvesting Challenge

Using the uniform-cost and A^* search algorithms with the cost and heuristic functions proposed previously, we now describe a midday case with the potential for negative action costs when harvested energy exceeds expended energy. The same utility function [Eqs. (17) and (18)] is applied in this case, with the goal achievement value motivating the planner to select energy-consuming actions. Applying a uniform-cost search over a 2 h search horizon exceeds the recursive memory allocation limits of the search code, failing to converge due to energy-collection incentives encouraging exploration of all paths with negative-cost branches. Reducing the search horizon to 1 h adequately manages the exploration space, producing the plan in Fig. 12. The resulting plan extends to within ~ 100 s of the search horizon. The results again show there are cases where the planner elects drift rather than selecting strictly goal-seeking actions. The final drift clearly extends beyond the point at which the batteries are charged, suggesting some benefit is gained by allowing the system to drift downwind at 100% charge before flight. However, the sawtooth energy budget trend suggests drift segments are nominally selected to recharge the batteries and drive the effective path cost to zero. The ability to drive an arbitrary path cost to zero at any given step is problematic for the planner. At each step where the cost is driven to zero, the search can be thought to

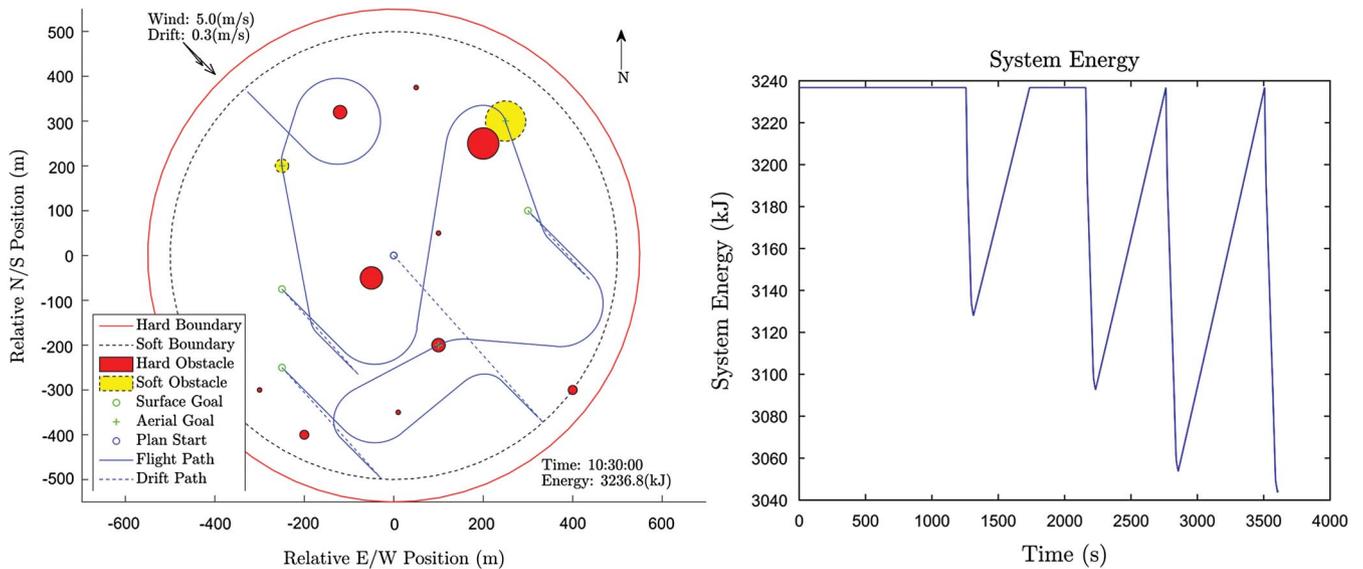


Fig. 13 Midday A^* search planning.

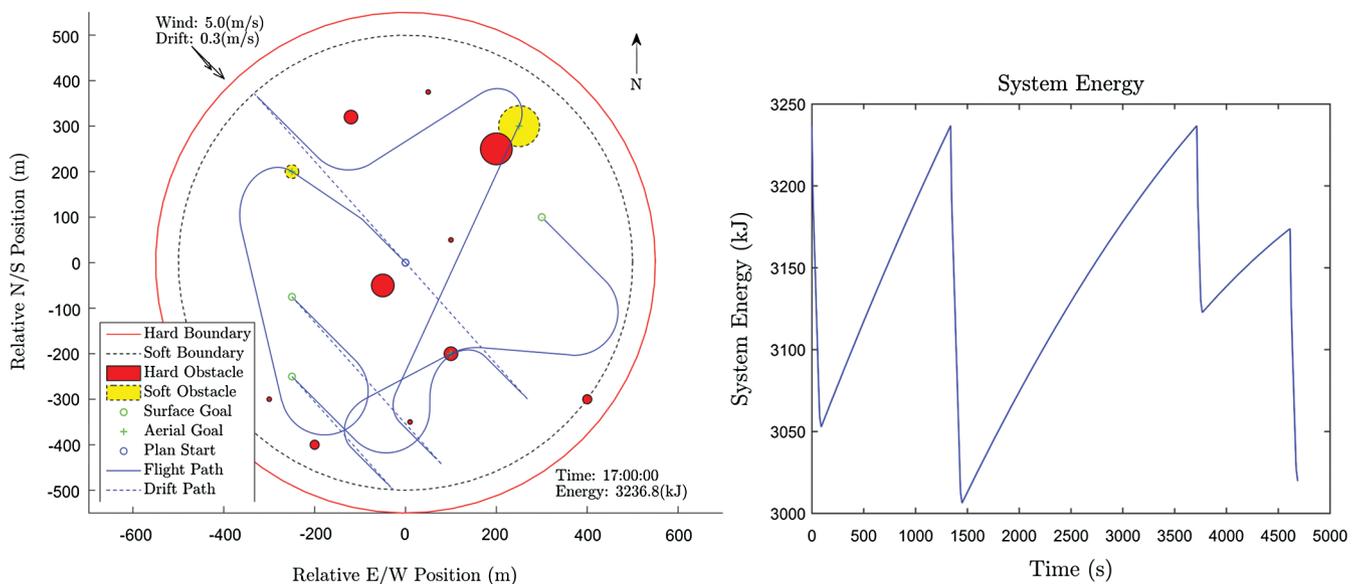


Fig. 14 Early dusk uniform-cost search planning.

restart with fewer goals; any previously reached goal will be on the visited list, and there will be no mathematically compelling reason to backtrack beyond the threshold of the zero path cost (unless all transitions in the branch following the zero-cost threshold are inadmissible or do not reach all goals). As such, it is the utility function and search-space geometry, and not a sense of global optimality, that determines the sequence of “canceled” goals that precede the final flight. Nevertheless, the search produces a viable plan provided the horizon is sufficiently close.

We subsequently applied the A^* search heuristic developed previously. Given that the heuristic estimates the positive cost through all goals, it may not be admissible in the solar-regenerative case where costs can be zero or negative. We subtracted the maximum system energy from the heuristic to ensure it stays negative, but this constant bias does not impact search results. Figure 13 presents the plan that results from applying the simplified TSP heuristic. Notably, sufficient separation from the boundary is available to allow drift for recovery of the terminal cost of both the uniform-cost and A^* search results and the difference between the terminal cost of the two searches is only ~ 30 kJ. However, although the uniform-cost search exceeds available memory with a 2 h search horizon, the A^* search with the TSP heuristic converges (nearly regardless of the horizon), in less than 10 s, the speed of the best short-horizon uniform-cost search.

D. Dawn/Dusk Missions: Low Solar Insolation

At this point, a set of clear FFSPs has been presented for the extreme cases of zero solar energy and high-intensity midday solar-regenerative planning. What remains is to consider the transition regions at dawn and dusk. Early cases to compare and contrast results in this region revealed that, as the sun neared the horizon, but distinctly before nightfall, the A^* search and uniform-cost search results became indistinguishable. If searches were conducted either earlier in the afternoon or later in the morning (late afternoon in this case), the results of the uniform-cost (Fig. 14) and A^* (Fig. 15) search strategies began to diverge steadily from one another.

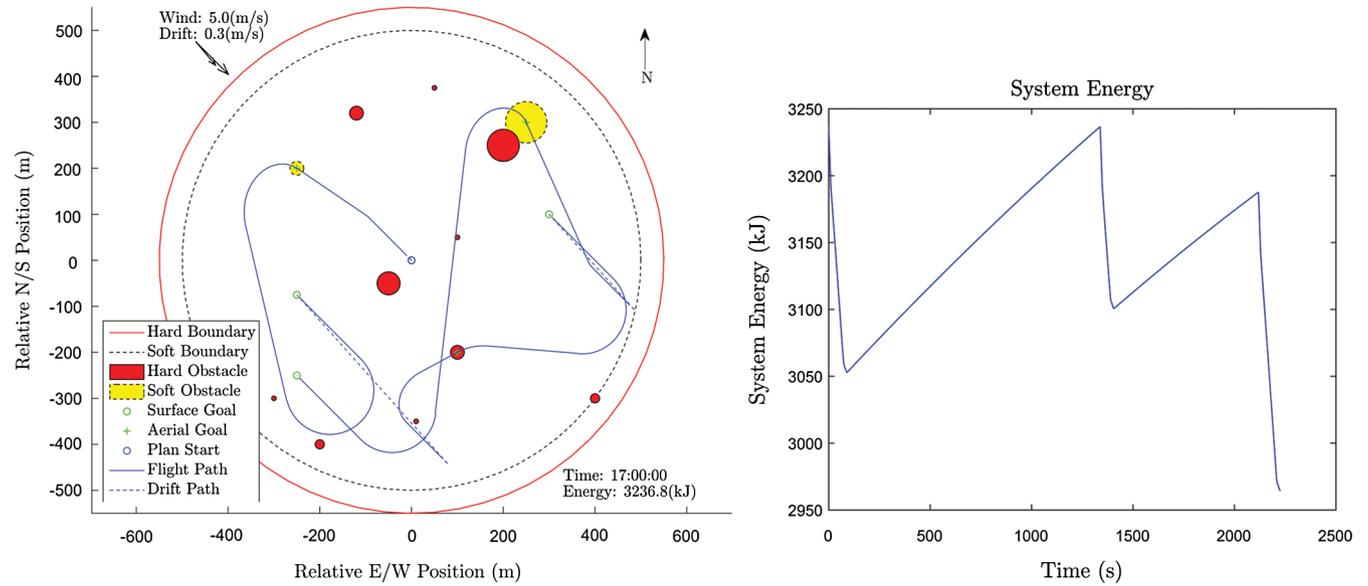


Fig. 15 Early dusk A^* search planning.

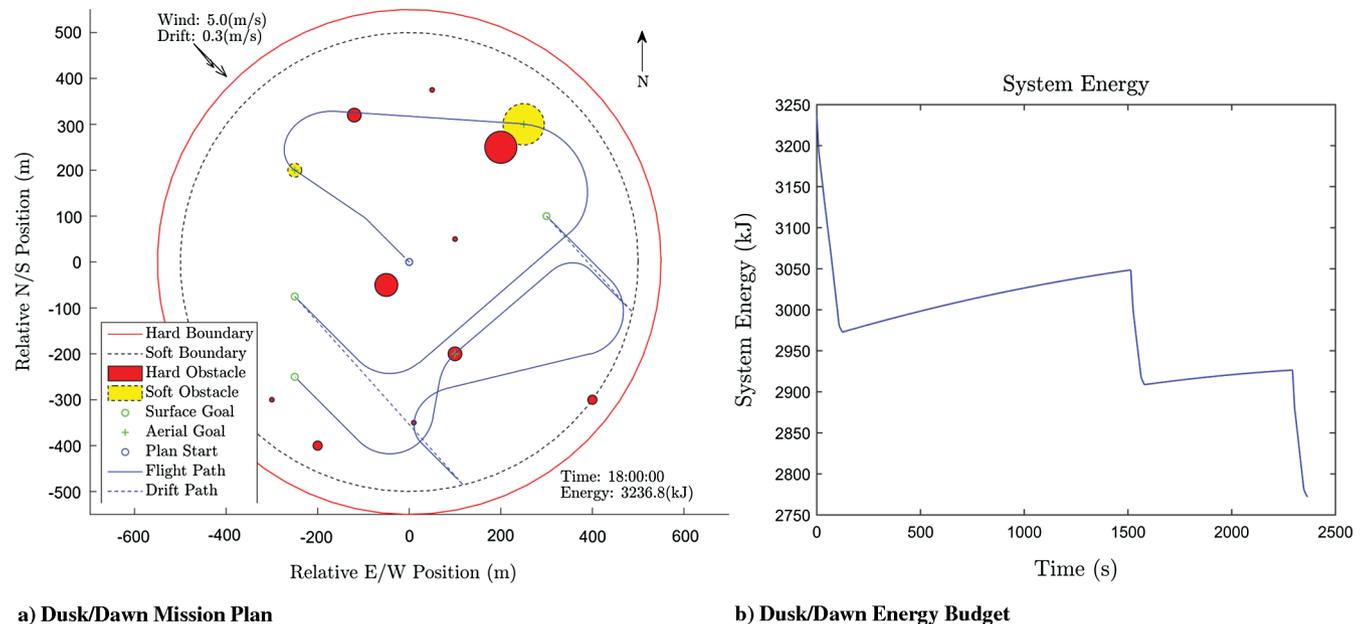


Fig. 16 Dusk/dawn mission planning.

Figure 16 presents an example of identical late-day/early-morning results. Solution convergence over the course of sunset (or divergence over sunrise) suggests the existence of a range of low solar insolation conditions for which the negative path cost may not impact the optimal solution because the energy cost can never be driven to zero through harvesting. It is beyond the scope of this paper to further support this hypothesis.

E. Discussion of Results

The FFSP algorithm has proven to be resilient to different configurations of goals, obstacles, constraints, and under various wind and solar insolation conditions. We conducted over more than 1000 randomly generated simulation scenarios (spanning more than three months of effort) and, in all cases, a solution was successfully found. Furthermore, we calculated a computational speed improvement of at least three times in all test cases; in most scenarios, a savings of up to 30 to 40 times was typical. Although we cannot yet prove our heuristic is admissible under all scenarios, we are confident in the algorithm's ability to solve the posed planning problem for all our use cases.

VI. Conclusions

In this paper was presented a multflight energy-aware planner to support long-duration autonomy. Environment and vehicle models for the Flying Fish solar-energy-harvesting seaplane provided the basis for flight drift plans that managed onboard energy and achieved surveillance mission goals. The planning problem was mapped to a modified traveling salesman problem. Physics-based trajectory planning, vehicle performance, and models of energy harvesting, usage, and storage were integrated into a discrete search tool for the determination of energy-optimal paths subject to vehicle performance and solar-energy recovery dynamics. Computational complexity was improved by the development of a novel heuristic, which was itself a low-complexity TSP problem, that decreased the search time by at least a factor of three in all test cases, with typical speed improvements of 30 to 40 times. Solution characteristics were examined over night, day, transitional dusk/dawn, and multiday planning horizons. In all cases, application of the current TSP heuristic was shown, despite the lack of admissibility with daylight, to converge to solutions comparable to exhaustive search with significantly lower search times.

This research is distinctive in its autonomous seaplane application, motivation by challenges observed during actual flight tests, and multflight planner formulation. Although individual models are based on the literature, vehicle performance, environment, and solar-energy-harvesting and usage models have never before been integrated into a multflight planner. Further, this is the first planner capable of optimizing both flight and surface drift, requiring knowledge of surface and airborne surveillance targets and obstacles.

Applications of an autonomous unmanned seaplane are far ranging, but the ultimate utility of the platform will not be realized until it can achieve truly unattended operation. Long-term milestones to this goal include further improving performance and robustness, more simulations under varying conditions, and field validating the system. The mission planner must be endowed with information to formulate not just situation-aware plans but to also adapt the mission plan to accommodate targets of opportunity, unexpected environment hazards, or vehicle system failures.

Appendix: Flying-Fish Platform

The Flying Fish airframe is a twin-tail twin-boom fixed-wing dual-pontoon floatplane (Fig. A1, Table A1). Structures are made unusually strong for a UAS to maximize resilience to the harsh ocean environment, so the airframe is relatively heavy. The twin-pontoon flotation system provides hydrodynamic stability but adds appreciable aerodynamic drag. The propulsion system includes a third "boost" pusher motor used only for takeoff. The set of three motors can deliver over 5 kW of propulsive power. The central avionics are housed in a fuselage below the main wing. Batteries and power electronics are housed in the vertical leg stanchions and pontoons, and solar cells are distributed over the upper wing surface. All avionics and power system housings and connectors are waterproofed.



Fig. A1 Flying fish solar energy-harvesting autonomous seaplane (phase 2).

Table A1 Phase 2 vehicle specifications

Attribute	Value
Wingspan b , wing area S , mean chord c	$b = 3.76$ m, $S = 1.88$ m, $c = 53.7$ cm
Airfoil, dihedral	NACA 2414, 3 deg
Pontoon length	1.15 m
Flight weight W , cruise speed V	30 kg, 17 m/s
Primary voltage, battery capacity	21 V, 48.6 Ah

Acknowledgments

This work was supported in part by the Office of Naval Research under award N000141410596. We would like to thank Dennis Bernstein for his feedback and the Flying Fish project team led by Guy Meadows, who is now Director of the Great Lakes Research Center at Michigan Technological University.

References

- [1] Eubank, R., and Atkins, E., "Energy-Aware Flight Management and Control of the Flying Fish Solar-Powered Autonomous Seaplane," *AUVSI Unmanned Systems North America Conference*, Denver, CO, 2010, pp. 910–950, <https://experts.umich.edu/en/publications/energy-aware-flight-management-and-control-of-the-flying-fish-sol>.
- [2] Eubank, R., Atkins, E., and Meadows, G., "Unattended Operation of an Autonomous Seaplane for Persistent Surface and Airborne Ocean Monitoring," *MTS/IEEE OCEANS*, IEEE, Piscataway, NJ, 2010, pp. 1–8.
- [3] Lawler, E. L., Lenstra, J. K., Kan, A. H. G. R., and Shmoys, D. B., *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, Vol. 3, Wiley, New York, 1985.
- [4] Russell, S. J., and Norvig, P., *Artificial Intelligence: A Modern Approach*, 3rd ed., Prentice-Hall, Upper Saddle River, NJ, 2010, pp. 366–473.
- [5] Barrett, A., and Weld, D. S., "Partial-Order Planning: Evaluating Possible Efficiency Gains," *Artificial Intelligence*, Vol. 67, No. 1, 1994, pp. 71–112. doi:10.1016/0004-3702(94)90012-4
- [6] Puterman, M. L., *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, Wiley, Hoboken, NJ, 2014, pp. 1–16.
- [7] Giunchiglia, F., "Using Abstrips Abstractions—Where Do We Stand?" *Artificial Intelligence Review*, Vol. 13, No. 3, 1999, pp. 201–213. doi:10.1023/A:1006507609248
- [8] Botea, A., Enzenberger, M., Müller, M., and Schaeffer, J., "Macro-FF: Improving AI Planning with Automatically Learned Macro-Operators," *Journal of Artificial Intelligence Research*, Vol. 24, 2005, pp. 581–621. doi:10.1613/jair.1696
- [9] Coles, A., and Smith, A., "Marvin: A Heuristic Search Planner with Online Macro-Action Learning," *Journal of Artificial Intelligence Research*, Vol. 28, Jan.–April 2007, pp. 119–156.
- [10] Davidov, D., et al., "Multiple-Goal Heuristic Search," *Journal of Artificial Intelligence Research*, Vol. 26, May–Aug. 2006, pp. 417–451.
- [11] Garvey, A. J., and Lesser, V. R., "Design-to-Time Real-Time Scheduling," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 23, No. 6, 1993, pp. 1491–1502. doi:10.1109/21.257749
- [12] Benton, J., Do, M., and Kambhampati, S., "Anytime Heuristic Search for Partial Satisfaction Planning," *Artificial Intelligence*, Vol. 173, No. 5, 2009, pp. 562–592. doi:10.1016/j.artint.2008.11.010
- [13] Keviczky, T., and Balas, G. J., "Software-Enabled Receding Horizon Control for Autonomous Unmanned Aerial Vehicle Guidance," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 3, 2006, pp. 680–694. doi:10.2514/1.15562
- [14] Bellingham, J., Richards, A., and How, J. P., "Receding Horizon Control of Autonomous Aerial Vehicles," *Proceedings of the 2002 American Control Conference, 2002*, Vol. 5, IEEE, Piscataway, NJ, 2002, pp. 3741–3746.
- [15] Hansen, E. A., and Zhou, R., "Anytime Heuristic Search," *Journal of Artificial Intelligence Research*, Vol. 28, Jan.–April 2007, pp. 267–297.
- [16] Bulitko, V., Sturtevant, N. R., Lu, J., and Yau, T., "Graph Abstraction in Real-Time Heuristic Search," *Journal of Artificial Intelligence Research*, Vol. 30, Sept.–Dec. 2007, pp. 51–100.
- [17] Sturtevant, N., and Buro, M., "Partial Pathfinding Using Map Abstraction and Refinement," *AAAI*, Vol. 5, Pittsburgh, PA, July 2005, pp. 1392–1397.
- [18] Musliner, D. J., Durfee, E. H., and Shin, K. G., "CIRCA: A Cooperative Intelligent Real-Time Control Architecture," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 23, No. 6, 1993, pp. 1561–1574. doi:10.1109/21.257754
- [19] Atkins, E. M., Abdelzaher, T. F., Shin, K. G., and Durfee, E. H., "Planning and Resource Allocation for Hard Real-Time, Fault-Tolerant Plan Execution," *Autonomous Agents and Multi-Agent Systems*, Vol. 4, Nos. 1–2, 2001, pp. 57–78. doi:10.1023/A:1010066729351
- [20] Do, M. B., and Kambhampati, S., "Sapa: A Multi-Objective Metric Temporal Planner," *Journal of Artificial Intelligence Research*, Vol. 20, Dec. 2003, pp. 155–194.
- [21] Beard, R. W., and McLain, T. W., *Small Unmanned Aircraft: Theory and Practice*, Princeton Univ. Press, Princeton, NJ, 2012, pp. 209–228.
- [22] McLain, T. W., and Beard, R. W., "Trajectory Planning for Coordinated Rendezvous of Unmanned Air Vehicles," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2000-4369, 2000.
- [23] Chandler, P., Rasmussen, S., and Pachter, M., "UAV Cooperative Path Planning," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2000-4370, 2000, pp. 1255–1265.
- [24] Eppstein, D., "Finding the k Shortest Paths," *SIAM Journal on Computing*, Vol. 28, No. 2, 1998, pp. 652–673. doi:10.1137/S0097539795290477
- [25] Latombe, J.-C., *Robot Motion Planning*, Kluwer Academic, Norwell, MA, 1991, pp. 153–199.
- [26] LaValle, S. M., and Kuffner, J. J., Jr., "Rapidly-Exploring Random Trees: Progress and Prospects," 2000, <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.38.1387>.
- [27] Bry, A., and Roy, N., "Rapidly-Exploring Random Belief Trees for Motion Planning Under Uncertainty," *IEEE International Conference on Robotics and Automation*, IEEE, Piscataway, NJ, 2011, pp. 723–730.
- [28] Karaman, S., and Frazzoli, E., "Incremental Sampling-Based Algorithms for Optimal Motion Planning," *Robotics Science and Systems VI*, Vol. 104, 2010.
- [29] Hargraves, C. R., and Paris, S. W., "Direct Trajectory Optimization Using Nonlinear Programming and Collocation," *Journal of Guidance, Control, and Dynamics*, Vol. 10, No. 4, 1987, pp. 338–342. doi:10.2514/3.20223
- [30] Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Journal on Optimization*, Vol. 12, No. 4, 2002, pp. 979–1006. doi:10.1137/S1052623499350013
- [31] Kamal, W. A., Gu, D.-W., and Postlethwaite, I., "Real Time Trajectory Planning for UAVs Using MILP," *44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05*, IEEE, Piscataway, NJ, 2005, pp. 3381–3386.
- [32] Dubins, L. E., "On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents," *American Journal of Mathematics*, Vol. 79, No. 3, 1957, pp. 497–516. doi:10.2307/2372560
- [33] Chitsaz, H., and LaValle, S. M., "Time-Optimal Paths for a Dubins Airplane," *2007 46th IEEE Conference on Decision and Control*, IEEE, Piscataway, NJ, 2007, pp. 2379–2384.
- [34] Betts, J. T., "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998, pp. 193–207. doi:10.2514/2.4231

- [35] Schultz, R. L., "Three-Dimensional Trajectory Optimization for Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 6, 1990, pp. 936–943. doi:10.2514/3.20564
- [36] Morse, B. S., Eng, C. H., and Goodrich, M. A., "UAV Video Coverage Quality Maps and Prioritized Indexing for Wilderness Search and Rescue," *5th ACM/IEEE International Conference on Human-Robot Interaction*, ACM/IEEE, New York, 2010, pp. 227–234.
- [37] Maza, I., and Ollero, A., "Multiple UAV Cooperative Searching Operation Using Polygon Area Decomposition and Efficient Coverage Algorithms," *Distributed Autonomous Robotic Systems 6*, Springer, New York, 2007, pp. 221–230.
- [38] Isaacs, J. T., Klein, D. J., and Hespanha, J. P., "Algorithms for the Traveling Salesman Problem with Neighborhoods Involving a {Dubins} Vehicle," *2011 American Control Conference*, IEEE, New York, 2011, pp. 1704–1709.
- [39] Carnie, R., Walker, R., and Corke, P., "Image Processing Algorithms for UAV 'Sense and Avoid'," *2006 IEEE International Conference on Robotics and Automation*, IEEE, Piscataway, NJ, May 2006, pp. 2848–2853.
- [40] Tomlin, C. J., Lygeros, J., and Shankar Sastry, S., "A Game Theoretic Approach to Controller Design for Hybrid Systems," *Proceedings of the IEEE*, Vol. 88, No. 7, 2000, pp. 949–970. doi:10.1109/5.871303
- [41] Richards, A., and How, J. P., "Aircraft Trajectory Planning with Collision Avoidance Using Mixed Integer Linear Programming," *2002 American Control Conference*, Vol. 3, IEEE, New York, 2002, pp. 1936–1941.
- [42] Campo, F. J. M., "The Collision Avoidance Problem: Methods and Algorithms," Ph.D. Thesis, Univ. Rey Juan Carlos, Madrid, 2010.
- [43] Williamson, T., and Spencer, N. A., "Development and Operation of the Traffic Alert and Collision Avoidance System (TCAS)," *Proceedings of the IEEE*, Vol. 77, No. 11, 1989, pp. 1735–1744. doi:10.1109/5.47735
- [44] Boisvert, R. E., and Orlando, V. A., "ADS-Mode S System Overview," *AIAA/IEEE 12th Digital Avionics Systems Conference*, AIAA, Reston, VA, 1993, pp. 104–109.
- [45] Orlando, V. A., and Drouilhet, P. R., *Mode S Beacon System: Functional Description*, Massachusetts Inst. of Technology Lincoln Lab. ATC-42-REV-B, 1982.
- [46] Chakrabarty, A., and Langelaan, J. W., "Energy-Based Long-Range Path Planning for Soaring-Capable Unmanned Aerial Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 4, 2011, pp. 1002–1015. doi:10.2514/1.52738
- [47] Klesh, A. T., and Kabamba, P. T., "Solar-Powered Aircraft: Energy-Optimal Path Planning and Perpetual Endurance," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 4, 2009, pp. 1320–1329. doi:10.2514/1.40139
- [48] Spangelo, S. C., Gilbert, E. G., Klesh, A. T., Kabamba, P. T., and Girard, A. R., "Periodic Energy-Optimal Path Planning for Solar-Powered Aircraft," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2009-6016, 2009.
- [49] Rabideau, G., Knight, R., Chien, S., Fukunaga, A., and Govindjee, A., "Iterative Repair Planning for Spacecraft Operations Using the ASPEN System," *Artificial Intelligence, Robotics and Automation in Space*, Vol. 440, 1999, pp. 99–106, <http://adsabs.harvard.edu/abs/1999ESASP.440...99R>.
- [50] Ceballos, A., et al., "A Goal-Oriented Autonomous Controller for Space Exploration," *ASTRA*, Vol. 11, 2011.
- [51] Roberts, G. N., and Sutton, R. (eds.), *Further Advances in Unmanned Marine Vehicles*, Control, Robotics & Sensors, Inst. of Engineering and Technology, England, U.K., 2012, <http://digital-library.theiet.org/content/books/ce/pbce077e>.
- [52] Wilson, M., Auslander, B., Johnson, B., Apker, T., McMahan, J., and Aha, D. W., "Towards Applying Goal Autonomy for Vehicle Control," *Goal Reasoning: Papers from the ACS Workshop*, Univ. of Maryland, Maryland, 2013, p. 127.
- [53] Michalsky, J. J., "The Astronomical Almanac's Algorithm for Approximate Solar Position (1950–2050)," *Solar Energy*, Vol. 40, No. 3, 1988, pp. 227–235. doi:10.1016/0038-092X(88)90045-X
- [54] Reda, I., and Andreas, A., "Solar Position Algorithm for Solar Radiation Applications," National Renewable Energy Lab. TR 5, Golden, CO, 2008.
- [55] Perez, R., Ineichen, P., Seals, R., Michalsky, J., and Stewart, R., "Modeling Daylight Availability and Irradiance Components from Direct and Global Irradiance," *Solar Energy*, Vol. 44, No. 5, 1990, pp. 271–289. doi:10.1016/0038-092X(90)90055-H
- [56] Spencer, J. W., "Fourier Series Representation of the Position of the Sun," *Search*, Vol. 2, No. 5, 1971, p. 172, <https://www.mail-archive.com/sundial@uni-koeln.de/msg01050.html>.
- [57] Iqbal, M., *An Introduction to Solar Radiation*, Academic Press, Orlando, FL, 1983, pp. 303–334.
- [58] Zimmerman, J. C., "Sun-Pointing Programs and Their Accuracy," NASA Scientific and Technical Information TR SAND-81-0761, Sandia National Labs., Albuquerque, NM, 1981, p. 30643, <https://www.sti.nasa.gov/>.
- [59] Chen, M., and Rincon-Mora, G. A., "Accurate Electrical Battery Model Capable of Predicting Runtime and I-V Performance," *IEEE Transactions on Energy Conversion*, Vol. 21, No. 2, June 2006, pp. 504–511. doi:10.1109/TEC.2006.874229
- [60] Gao, L., Liu, S., and Dougal, R. A., "Dynamic Lithium-Ion Battery Model for System Simulation," *IEEE Transactions on Components and Packaging Technologies*, Vol. 25, No. 3, 2002, pp. 495–505. doi:10.1109/TCAPT.2002.803653
- [61] Rao, V., Singhal, G., Kumar, A., and Navet, N., "Battery Model for Embedded Systems," *2005 IEEE International Conference on VLSI Design*, IEEE, Piscataway, NJ, 2005, pp. 105–110.
- [62] Kroeze, R. C., and Krein, P. T., "Electrical Battery Model for Use in Dynamic Electric Vehicle Simulations," *2008 IEEE Power Electronics Specialists Conference*, IEEE, Piscataway, NJ, 2008, pp. 1336–1342.
- [63] Johnson, V. H., Pesaran, A. A., and Sack, T., *Temperature-Dependent Battery Models for High-Power Lithium-Ion Batteries*, National Renewable Energy Lab., Golden, CO, 2001.
- [64] Johnson, D. S., and McGeoch, L. A., "The Traveling Salesman Problem: A Case Study in Local Optimization," *Local Search in Combinatorial Optimization*, Vol. 1, Nov. 1995, pp. 215–310, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.70.7639&rep=rep1&type=pdf>.
- [65] Kruskal, J. B., "On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem," *Proceedings of the American Mathematical Society*, Vol. 7, No. 1, 1956, pp. 48–50. doi:10.1090/S0002-9939-1956-0078686-7
- [66] Dorigo, M., and Gambardella, L. M., "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem," *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, 1997, pp. 53–66. doi:10.1109/4235.585892
- [67] Held, M., and Karp, R. M., "A Dynamic Programming Approach to Sequencing Problems," *Journal of the Society for Industrial and Applied Mathematics*, Vol. 10, No. 1, 1962, pp. 196–210. doi:10.1137/0110015
- [68] Karp, R. M., "Dynamic Programming Meets the Principle of Inclusion and Exclusion," *Operations Research Letters*, Vol. 1, No. 2, 1982, pp. 49–51. doi:10.1016/0167-6377(82)90044-X
- [69] Dutta, M., Pragy, A., and Pratyush, , "TSP Solution Using Dimensional Ant Colony Optimization," *2015 Fifth International Conference on Advanced Computing and Communication Technologies*, IEEE, Piscataway, NJ, 2015, pp. 506–512.

- [70] Jonker, R., and Volgenant, T., "Transforming Asymmetric into Symmetric Traveling Salesman Problems," *Operations Research Letters*, Vol. 2, No. 4, 1983, pp. 161–163.
doi:10.1016/0167-6377(83)90048-2
- [71] Saha, M., Roughgarden, T., Latombe, J.-c., and Sánchez-Ante, G., "Planning Tours of Robotic Arms Among Partitioned Goals," *International Journal of Robotics Research*, Vol. 25, No. 3, 2006, pp. 207–223.
doi:10.1177/0278364906061705

M. J. Kochenderfer
Associate Editor