Trajectory Optimization for High-Altitude Long-Endurance UAV Maritime Radar Surveillance

ANGUS BROWN
DAVID ANDERSON
University of Glasgow, Glasgow, U.K.

For an unmanned aerial vehicle (UAV) carrying out a maritime radar surveillance mission, there is a tradeoff between maximizing information obtained from the search area and minimizing fuel consumption. This article presents an approach for the optimization of a UAV's trajectory for maritime radar wide area persistent surveillance to simultaneously minimize fuel consumption, maximize mean probability of detection, and minimize mean revisit time. Quintic polynomials are used to generate UAV trajectories due to their ability to provide complete and complex solutions while requiring few inputs. Furthermore, the UAV dynamics and surveillance mission requirements are used to ensure that a trajectory is realistic and mission compatible. A wide area search radar model is used within this article in conjunction with a discretized grid in order to determine the search area's mean probability of detection and mean revisit time. The trajectory generation method is then used in conjunction with a multiobjective particle swarm optimization algorithm to obtain a global optimum in terms of path, airspeed (and thus time), and altitude. The performance of the approach is then tested over two common maritime surveillance scenarios and compared to an industry recommended baseline.

Manuscript received July 10, 2018; revised March 1, 2019 and July 24, 2019; released for publication September 30, 2019. Date of publication November 1, 2019; date of current version June 9, 2020.

DOI. No. 10.1109/TAES.2019.2949384

Refereeing of this contribution was handled by L. Pollini.

This work was supported in part by Leonardo MW Ltd under Grant SELEX/GU/2015/SOW01 and in part by the ESPRC under Grant EP/N509176/1.

Authors' addresses: The authors are with the School of Engineering, University of Glasgow, G12 8QQ Glasgow, U.K., E-mail: (a.brown.6@research.gla.ac.uk; dave.anderson@glasgow.ac.uk). *Corresponding author: Angus Brown.)*

I. INTRODUCTION

Airborne surveillance is of significant interest for both military and civil applications. Radar, in comparison to other remote sensors, provides large area surveillance in both adverse weather conditions (e.g., rain, fog) and varying light conditions (i.e., day and night). An airborne platform is often required for large area maritime radar surveillance. As such, the length of surveillance time is dependent on the fuel consumption of said platform, which in turn is dependent on the platform trajectory. However, the platform trajectory also affects the surveillance of the search area. In particular, it affects the visibility of the radar and where the main beam intersects the surface. Specifically, the platform airspeed, altitude, and path will affect the fuel consumption and how long the radar's beam stays within a given area, thus affecting both the probability of detection and revisit time.

For persistent surveillance missions, the aim is not necessarily to search for a specific target but to patrol a region of interest where complete coverage is required (i.e., every point in the search area is visited to some degree at least once). Furthermore, the platform trajectory is required to be as continuous as possible with little to no downtime to allow for constant surveillance. Therefore, minimizing the fuel consumption of the platform allows for a given surveillance trajectory to be flown on for longer. High-altitude UAVs are typically designed for long endurance, and as such are able to perform surveillance missions for significant amount of time before a refuel is required [1]. Additionally, UAVs have a significant advantage over other airborne platforms, in that they are able to fly any time of day with minimal human input. For both these reasons, a high-altitude UAV was selected as the platform for this problem. Note that high altitude is generally defined as above commercial altitudes (taken as 11 000 m).

There is therefore a tradeoff between minimizing the UAV fuel consumption and maximizing search area coverage, where the coverage is defined in terms of both the probability of detecting a given target and the revisit time (i.e., the time between covering the same point in the search area). Increasing the coverage of the surveillance area requires a larger trajectory, resulting in more fuel consumption. However, there is also a tradeoff between the revisit time and the probability of detection. If the trajectory covers a large area at a low speed, a lot of time is spent at each point in the search area, which increases the overall probability of detection. However, this trajectory type results in a greater time between revisiting a given point in the search area. Thus, the aim is to obtain a trajectory that simultaneously minimizes fuel consumption, maximizes the average probability of detection, and minimizes average revisit time for a given search area. These three criteria were quantified with cost functions for which a multiobjective global optimization algorithm was used to obtain trajectories that minimize the costs. Currently, within industry [2], only specific trajectories are considered, which may not be optimal for a given search area and mission. The optimized trajectories

were compared with industry recommended trajectories to highlight the performance of this approach.

Note that this problem deals with finding an optimal trajectory for the surveillance of a prespecified area (e.g., a fishing zone). As such, the solution for a given area does not need to be solved in real time [3] and an offline solution is adopted.

A. Related Work

A simple way to define a UAV trajectory is to use polynomials in time [4]. In particular, quintic polynomials provide a simple method for the representation of velocities and accelerations for a complete trajectory of a given platform. The main advantage of polynomials, relative to other trajectory methods, is the ability to specify the start and end conditions, which is a requirement in this case. These polynomials have been used for modeling both fixed-wing UAV [5] and multirotor [6], [7] trajectories. For optimizing a UAV's trajectory for surveillance, it would be typical to consider the four-dimensional (4-D) problem (space and time). However, for maritime surveillance, the surface is level, thus there would be no need to change altitude, which is therefore assumed constant throughout a given trajectory. There have been numerous implementations of polynomials for the use of trajectory optimization. The majority of the work focuses on minimizing path risk [8], minimizing path length in the face of objects [9], maximizing threat avoidance while minimizing fuel consumption [10], or a combination of fuel minimization, threat avoidance, and reconnaissance [11]. In addition, multiobjective optimization has been implemented for polynomial UAV trajectory optimization in order to maximize threat avoidance while also incorporating terrain constraints [12] and also fuel consumption [13].

Optimizing a full UAV trajectory involves a large search space, which often requires a global optimization algorithm. Evolutionary algorithms provide a method for global optimization while also maintaining relative efficiency. Notably, both genetic algorithms (GAs) [14] and particle swarm optimization (PSO) [15] have been used for trajectory optimization and path planning. In [16], PSO appears to outperform GA in terms of computational efficiency when applied to constrained nonlinear problems with continuous design variables, as is the case with the proposed problem. For these problems, PSO also achieves higher quality solutions in general. While GA generally outperforms PSO [17] for discrete trajectory optimization cases, according to Besada-Portas [18], PSO outperforms the GA for continuous trajectory scenarios, which involve a series of waypoints and areas of significance. Additionally, when a polynomial trajectory generation method has been used, PSO has been found to outperform GA [19], [20]. Similarly, Foo *et al.* [15] used PSO for B-spline trajectory optimization. For these reasons, PSO was chosen.

Current literature has not considered sensor trajectory optimization for a large area under persistent surveillance. For sensor trajectory optimization (with sensor modeling), up to now, the work has focused on either a downward-looking sensor [21]–[24] or synthetic aperture radar [25]. In the aforementioned studies, the input to the UAV is some form of discrete series of heading changes or waypoints. This method generates continuous polynomials for smooth and efficient circular turns. Additionally, in the abovementioned papers, the trajectory is optimized for searching certain locations of interest within the search area. This problem differs from the problem presented in this article, where trajectories are obtained that maximize the coverage for a whole search area. Additionally, due to the cyclic nature of the requirement that the UAV returns to its starting pose, the type of search algorithm that can be used is limited.

For the papers mentioned above, both the search area and sensor are modeled at a much lower resolution. For large area problems, higher resolution is required, which is a source of great computational cost. This article outlines the trajectory optimization method in a way that makes obtaining optimal solutions practicable.

Specifically, in both [21] and [25], there are multiple small regions with the start point, end point, and visiting order of the regions prespecified. In the case mentioned in [25], an A* search algorithm is employed, which cannot be applied to this problem for a multitude or reasons, primarily due to the problems cyclic nature. In the case mentioned in [21], the optimizer is preseeded which, while increasing computational efficiency, does not perform a truly global search. This article presents the surveillance trajectory problem as a global optimization problem by allowing a large degree of freedom in terms of path, altitude, and airspeed.

In [24], a receding horizon approach is used, which does not provide a global search. Additionally, the UAV height, commanded airspeed, initial position, and initial heading are fixed. These constraints sacrifice potential solutions in favor of computational efficiency. In their case, the computational efficiency is more advantageous; however, in the cases presented within this article, optimality is a priority over computational efficiency.

B. Main Contributions

The contribution of this article is the optimization of trajectories for large area radar surveillance using the following.

1) First, a polynomial trajectory generation method is derived that provides complex trajectories while requiring few inputs. This method accounts for the fixed-wing platform dynamics and propulsion while also accounting for the requirements of a persistent surveillance mission (i.e., the start pose is equal to the end pose).
2) A radar mathematical model is derived for maritime wide area surveillance. This model includes the earth's curvature and sector scan. Furthermore, a method for determining the search area coverage in

terms of probability of detection and revisit time is outlined.

3) The considerations of the required platform for this mission type are accounted for. These considerations include the high-altitude region of operation, long mission time, and high fuel-to-weight ratio of the platform.
4) Finally, the surveillance trajectory is used with a multiobjective global optimization algorithm. Results are compared to an industry recommended baseline, which shows the performance of this method. Additionally, the geometric similarity of trajectories within the Pareto front are highlighted.

Section II outlines the trajectory generation method and formulates the fuel consumption for a given trajectory. This section also derives the maritime radar wide area surveillance model, and how it is used to obtain the probability of detection and revisit time at a given point in the search area. Section III describes the cost function and the optimization method. Section IV presents simulation results, and Section V concludes this article.

II. UAV SURVEILLANCE TRAJECTORY GENERATION

A. Polynomial Trajectory

The polynomials used to define the UAV's trajectory can be a function of either position on a constant altitude plane with Cartesian coordinates (x,y) or velocity, which is defined by airspeed and heading, with each method having their advantages and disadvantages. The advantage of using a polynomial in x and y is that specific coordinates can be defined such that the UAV passes through them. In the case of a surveillance trajectory, a polynomial in x and y allows for the UAV to return to its initial coordinate, thus allowing for a continuous trajectory. However, one of the most common UAV maneuvers is the banked turn where a polynomial defined in x and y essentially approximates sinusoidal functions. For banked turns that result in a large change in heading, the polynomial cannot approximate the large trigonometric segment as effectively as it can be for a shorter segment. This insufficient approximation can result in unrealistic trajectories.

For a polynomial in heading and velocity, the banked turns are easily represented. However, the major drawback of these polynomials is the inability to control specific coordinates. This drawback prevents a repeated surveillance trajectory where the start and end positions and heading are equal. Thus, both polynomials will be used here with the majority of the trajectory defined by a series of n_{h} heading polynomials with an x and y polynomial used to return the UAV to the starting coordinate. In this case, the velocity polynomial is ignored to simplify the problem given that there would be little reason for the UAV to change airspeed during a surveillance mission.

Starting with the return polynomial defined in x and y (where x and y are in north, east, down (NED) coordinates with x parallel to lines of constant latitude and y parallel to lines of constant longitude), the position $\boldsymbol{p}$, velocity $\boldsymbol{v}$, and acceleration $\boldsymbol{a}$ can be obtained as follows:

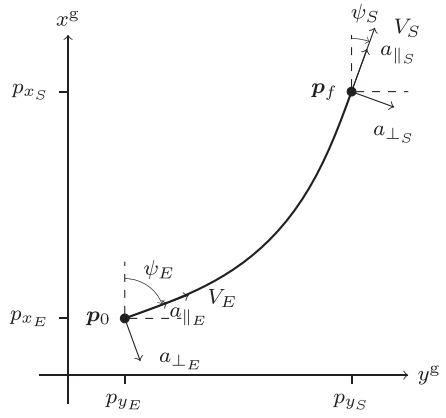

Fig. 1. Illustration of the return polynomial components. p_{x_E} and p_{y_E} indicate the last x and y position of the combined heading polynomials, whereas p_{x_S} and p_{y_S} indicate the initial starting position of the whole trajectory.

$$\boldsymbol{p}(\tau) = \boldsymbol{c}\tau^5 + \boldsymbol{d}\tau^4 + \boldsymbol{e}\tau^3 + \boldsymbol{f}\tau^2 + \boldsymbol{g}\tau + \boldsymbol{h} \tag{1}$$

$$\boldsymbol{v}(\tau) = \frac{\mathrm{d}\boldsymbol{p}(\tau)}{\mathrm{d}\tau} = 5\boldsymbol{c}\tau^4 + 4\boldsymbol{d}\tau^3 + 3\boldsymbol{e}\tau^2 + 2\boldsymbol{f}\tau + \boldsymbol{g} \tag{2}$$

$$\boldsymbol{a}(\tau) = \frac{\mathrm{d}\boldsymbol{v}(\tau)}{\mathrm{d}\tau} = 20\boldsymbol{c}\tau^3 + 12\boldsymbol{d}\tau^2 + 6\boldsymbol{e}\tau + 2\boldsymbol{f} \tag{3}$$

where $\boldsymbol{p}(\tau) = [x(\tau), y(\tau)]$, $\boldsymbol{v}(\tau) = [\dot{x}(\tau), \dot{y}(\tau)]$, and $\boldsymbol{a}(\tau) = [\ddot{x}(\tau), \ddot{y}(\tau)]$. In addition, $\tau \in [0, \tau_f]$, where $\tau_f = t_f - t_0$ with t_0 and t_f being the respective start and end times of the polynomial. The 1×2 vectors $\boldsymbol{c}$, $\boldsymbol{d}$, $\boldsymbol{e}$, $\boldsymbol{f}$, $\boldsymbol{g}$, and $\boldsymbol{h}$ define the polynomial with the first column defining the x-axis and the second column defining the y-axis.

The start point of the polynomial consists of a position, velocity, and acceleration denoted by $\boldsymbol{p}_0$, $\boldsymbol{v}_0$, and $\boldsymbol{a}_0$. Similarly, the end point consists of $\boldsymbol{p}_f$, $\boldsymbol{v}_f$, and $\boldsymbol{a}_f$. These values can be obtained by recalling that the return polynomial connects the last coordinate from the heading polynomials to the initial starting coordinate. As a result, the following can be stated: $\boldsymbol{p}_0 = \boldsymbol{p}_E$, where $\boldsymbol{p}_E$ indicates the last coordinate reached by the heading polynomials; $\boldsymbol{p}_f = \boldsymbol{p}_S$ where $\boldsymbol{p}_S$ is the initial starting coordinate of the whole trajectory; $\boldsymbol{v}_0 = [V_E \cos(\psi_E),\ V_E \sin(\psi_E)]$, where V_E and ψ_E are the airspeed and heading of the UAV at the last coordinate reached; $\boldsymbol{v}_f = [V_E \cos(\psi_S),\ V_E \sin(\psi_S)]$, where ψ_S is the initial heading of the UAV for the whole trajectory; $\boldsymbol{a}_0 = [a_{\parallel_E} \cos(\psi_E) - a_{\perp_E} \sin(\psi_E),\ \ a_{\parallel_E} \sin(\psi_E) + a_{\perp_E} \cos(\psi_E)]$, where $a_{\parallel_E}$ and $a_{\perp_E}$ are the respective parallel and normal accelerations to the UAV's forward direction at the last coordinate reached; and $\boldsymbol{a}_f = [a_{\parallel_S} \cos(\psi_S) - a_{\perp_S} \sin(\psi_S),\ a_{\parallel_S} \sin(\psi_S) + a_{\perp_S} \cos(\psi_S)]$, where $a_{\parallel_S}$ and $a_{\perp_S}$ are the initial UAV's respective parallel and normal accelerations (where $a_{\parallel_S}$ is taken to be 0). Fig. 1 outlines the components of the return polynomial trajectory.

Using the three polynomial start conditions in conjunction with (1)–(3), the following can be obtained:

$$\boldsymbol{p}(0) = \boldsymbol{p}_0 = \boldsymbol{h} \tag{4}$$

$$\boldsymbol{v}(0) = \boldsymbol{v}_0 = \boldsymbol{g} \tag{5}$$

$$\boldsymbol{a}(0) = \boldsymbol{a}_0 = \boldsymbol{f}\,. \tag{6}$$

The above equations can then be used with the three end conditions to form three simultaneous equations. Solving which gives the following:

$$\boldsymbol{e} = \big(-{\tau_f}^2(\boldsymbol{a}_0 - 3\boldsymbol{a}_f) - 4\tau_f(3\boldsymbol{v}_0 + 2\boldsymbol{v}_f) - 20(\boldsymbol{p}_0 - \boldsymbol{p}_f)\big)/\left(2{\tau_f}^3\right) \tag{7}$$

$$\boldsymbol{d} = \big({\tau_f}^2(3\boldsymbol{a}_0 - 2\boldsymbol{a}_f) + 2\tau_f(8\boldsymbol{v}_0 + 7\boldsymbol{v}_f) + 30(\boldsymbol{p}_0 - \boldsymbol{p}_f)\big)/\left(2{\tau_f}^4\right) \tag{8}$$

$$\boldsymbol{c} = \big(-{\tau_f}^2(\boldsymbol{a}_0 + \boldsymbol{a}_f) - 6\tau_f(\boldsymbol{v}_0 + \boldsymbol{v}_f) - 12(\boldsymbol{p}_0 - \boldsymbol{p}_f)\big)/\left(2{\tau_f}^5\right)\,. \tag{9}$$

For the heading polynomials, similar methods are used. The polynomials obtained are for $\psi_i(\tau_i)$, $\dot{\psi}_i(\tau_i)$, and $\ddot{\psi}_i(\tau_i)$ for $i = 1, \ldots, n_\mathrm{h}$. Note that for each segment, $\ddot{\psi}_0$ and $\ddot{\psi}_f$ are taken to be 0. Also note that the end conditions of a given polynomial i are equal to the start conditions of subsequent polynomial $i+1$. This continuity allows for each polynomial to be defined in the time domain by τ given that the initial polynomial starts at time 0. Furthermore, given continuity at the polynomial boundaries, the n_h heading polynomials can be combined to form one continuous entity. Additionally, including the return polynomial (for a total of n_w polynomials) forms another continuous entity. It is necessary to differentiate between the two as certain properties are evaluated differently for the return polynomial (in x and y), but the full trajectory is still required.

In order to obtain various properties (such as the aerodynamic and propulsive components) along the trajectory in a computationally efficient manner, both the combined heading polynomials and the return polynomial were discretized with a spacing of $\Delta\tau_\mathrm{f}$ (taken to be 1 s). The total number of discrete points along the combined heading polynomials is given by $k_h = \lfloor t_{n_{\mathrm{h}f}}/\Delta\tau_\mathrm{f} \rfloor + 1$, whereas the total number of discrete points along the full trajectory is given by $k_w = \lfloor t_{n_{\mathrm{w}f}}/\Delta\tau_\mathrm{f} \rfloor + 1$. The evaluated position and velocity vectors for the combined heading polynomials at discrete point h (for $h = 1, \ldots, k_h$) in x and y are then given by $\boldsymbol{p}_h$ and $\boldsymbol{v}_h$, respectively.

For the heading polynomials, the velocity in x and y can be obtained for a given heading ψ_h and airspeed V_h (12) at a given step h with the following:

$$\boldsymbol{v}_h = [V_h \cos(\psi_h), V_h \sin(\psi_h)]\,. \tag{10}$$

In order to determine the return polynomial, the final coordinate in x and y needs to be calculated. With the use of the trapezoidal rule for numerical integration applied to (10), the position can be updated as follows:

$$\boldsymbol{p}_h = \boldsymbol{p}_{h-1} + \Delta\tau\,\frac{\boldsymbol{v}_h + \boldsymbol{v}_{h-1}}{2}\,. \tag{11}$$

As previously stated, each polynomial is defined by its start and end conditions as well as the value of τ. For the polynomial in x and y, the start and end conditions are already known, and as such the polynomial is simply defined by the value of τ, which for this polynomial is denoted by τ_R. A drawback of heading polynomials not previously mentioned is that the nature of a polynomial makes transitions overly smooth and slow (i.e., a change in heading lasting the whole segment). These transitions are not representative of UAV flight, and so for this reason, it is assumed that each segment has a constant heading rate such that a singular segment is defined by τ and $\Delta\psi$ (which is equivalent to $\psi_f - \psi_0$). The heading rate is then taken as $\Delta\psi/\tau$. However, this assumption results in discontinuous segments. In order to remedy this discontinuity, transitional segments were introduced to blend the heading rates. These segments are defined as having a fixed value of 20 s for τ_tr with the start and end heading rates for the transitional segments defined as $\dot{\psi}_{\mathrm{tr}_0} = \dot{\psi}_{i_f}$ and $\dot{\psi}_{\mathrm{tr}_f} = \dot{\psi}_{i+1_0}$. The average heading rate between polynomial i and polynomial $i+1$ can then be used to obtain the change in heading for the transitional segment: $\Delta\psi_\mathrm{tr} = \tau_\mathrm{tr}(\dot{\psi}_{i_f} + \dot{\psi}_{i+1_0})/2$.

For the sake of computational time, ways of reducing the search space were explored. One such way was to replace the setting of the speed with a normalized value μ_V using the stall speed V_stall and the minimum drag speed V_md. The commonly used formulations for these speeds were used. The normalization was done on a linear scale such that for $\mu_V = 0$, the speed would equal the stall speed, and for $\mu_V = k_V$, the speed would equal the minimum drag speed. The value of k_V was set to 0.288 and was chosen as such because it allows the UAV to reach the maximum airspeed V_max at any altitude. The equation to obtain the UAV airspeed from the normalized value at a given step k along the combined heading polynomials is given as follows:

$$V_h = V_{\mathrm{stall}_h} + \mu_V\,\frac{V_{\mathrm{md}_h} - V_{\mathrm{stall}_h}}{k_V}\,, \quad \{\mu_V \in \mathbb{R} | \mu_V \in [0, 1]\}\,. \tag{12}$$

This equation would allow for the optimizer to change altitude without offsetting the UAV's speed while also being used to update the UAV's airspeed throughout the heading polynomials. Given that the return polynomial is defined in x and y, it is not simple to adjust the velocity throughout its trajectory. Therefore, for the return segment, it is assumed that both the start and end velocities will be equal to the value of the velocity at the end of the last heading polynomial.

To further heuristically reduce the search space, the value of τ_R for the return segment can be replaced with a τ modifier value k_τ. Since the start and end velocities are equal, the straight line distance between the start and end coordinates of the return segment can be used to estimate a straight line time t_sl. The value of k_τ is then used to modify the straight line distance such that the value of τ_R is obtained as

$$\tau_\mathrm{R} = k_\tau t_\mathrm{sl}\,, \quad \{k_\tau \in \mathbb{R} | k_\tau \in [1, 1.1579]\} \tag{13}$$

TABLE I
UAV Specifications

Parameter	Unit	Value
Operational ceiling, $h^{\mathrm{p}}_{\max}$	m	19 800
Minimum altitude, $h^{\mathrm{p}}_{\min}$	m	11 000
Maximum airspeed, $V_{\max}$	$\mathrm{m\,s^{-1}}$	186
Maximum load factor, $n_{\max}$		1.035
Maximum coefficient of lift, $C_{l_{\max}}$		1.65
Wing reference area, S	$\mathrm{m^2}$	50
Wing aspect ratio, AR		25
Zero-lift drag coefficient, C_{d_0}		0.0135
Oswald efficiency, e_0		0.75
Take-off thrust, T_0	N	34 000
Take-off mass, m_{TOW}	kg	11 600
Fuel capacity, m_{fuel}	kg	6500

where the upper value of k_τ is set such that the UAV can perform a 45° turn with constant speed.

In conjunction with the value of k_τ for the return polynomial and the values of τ_i and $\Delta\psi_i$ for the heading polynomials, additional parameters are used to define the trajectory. These consist of the initial coordinate $\boldsymbol{p}_S$, normalized airspeed μ_V, initial heading ψ_S, and altitude of platform operation h^{p}. The whole polynomial trajectory can then be defined by vector $\boldsymbol{x}$ such that

$$\boldsymbol{x} = [\boldsymbol{\tau}, \boldsymbol{\Delta\psi}, \boldsymbol{p}_S, \mu_V, \psi_S, k_\tau, h^{\mathrm{p}}] \tag{14}$$

where $\boldsymbol{\tau} = [\tau_1, \ldots, \tau_{n_{\mathrm{h}}}]$, $\boldsymbol{\Delta\psi} = [\Delta\psi_1, \ldots, \Delta\psi_{n_{\mathrm{h}}}]$, and $\boldsymbol{p}_S = [x_{1_0}, y_{1_0}]$.

B. Fixed-Wing UAV Fuel Consumption

For surveillance missions, the UAV will fly for as long as possible while carrying a radar unit. For this reason, the UAV specifications were loosely based on the Northrop Grumman RQ-4 Global Hawk [1], [26]. Further to these specifications, it is also necessary to consider some mission-specific limits. Primarily, the UAV needs to fly above commercial aircraft altitudes, and so the minimum altitude was set to 11 000 m. It is also assumed that there is no wind acting on the UAV. The UAV specifications are then outlined in Table I.

From take-off to landing, the flight is assumed to be broken into three stages. The first stage encompasses the process of take-off, climb, and cruising to the start of the surveillance mission. The second stage involves carrying out the surveillance mission and the third and final stage encompasses the process of returning to base and landing. It is assumed that the fuel fraction (i.e., the fraction of fuel consumed during a given stage of flight) for the first and third stages is 0.9. These fractions also account for any excess fuel required for safety. Therefore, the initial weight of the UAV at the start of the trajectory is denoted by m_{I}, which is equal to $\frac{m_2}{m_1} m_{\mathrm{TOW}}$, whereas the minimum required weight of the UAV at the end of the trajectory is denoted by m_{F} and equals $m_{\mathrm{I}}(m_{\mathrm{TOW}} - m_{\mathrm{fuel}})/(\frac{m_2}{m_1}\frac{m_4}{m_3} m_{\mathrm{TOW}})$.

The rate of fuel consumption of the UAV can be calculated (15) by the product of the thrust-specific fuel consumption TSFC and the required thrust T_{required}, both of which are dependent on the setting of the UAV. The rate of fuel consumption can then be integrated to obtain the total fuel consumed for a given trajectory. Given the large fuel-to-weight ratio along with the potentially long flight times, it is important to consider the effects of the change in UAV mass along the trajectory. This effect is achieved by incorporating the current UAV mass m_{C} into the required thrust equation. Since the path is discretized, the trapezoidal method for numerical integration is then applied to update the current UAV mass at each point in the discretized trajectory using the rate of fuel consumption as shown in the following equation:

$$\dot{m}_{\mathrm{f}_w} = T_{\mathrm{required}_w} \mathrm{TSFC}_w \tag{15}$$

$$m_{\mathrm{C}_w} = m_{\mathrm{C}_{w-1}} - \Delta\tau_{\mathrm{f}} \frac{\dot{m}_{\mathrm{f}_w} + \dot{m}_{\mathrm{f}_{w-1}}}{2}\,. \tag{16}$$

The required thrust depends on the setting of the UAV, which includes the altitude, airspeed, current mass, bank angle, and acceleration. For the heading polynomials, the acceleration is obtained with respect to the local coordinates. In other words, the tangential component is tangent to the path curvature defined by the heading polynomial and is simply given by $a_{\parallel_h} = (v_h - v_{h-1})/\Delta\tau_f$. The centripetal component is defined along the outward normal to the curve and is obtained with the radius of curvature ($\rho = v/\dot{\psi}$) as follows: $a_{\perp_h} = -{v_h}^2/\rho_h$.

For the return segment, the polynomial is defined in x and y coordinates. However, the UAV will fly with forward velocity along the longitudinal axis. It is then necessary to resolve the return polynomial in the direction of the velocity vector [27]. By denoting a given velocity vector by $\boldsymbol{v}$ and a given acceleration vector by $\boldsymbol{a}$, the airspeed V for the return polynomial is then simply calculated by $V = \|\boldsymbol{v}\|$.

The x and y acceleration of the UAV can be resolved along the longitudinal axis (parallel to the airspeed) to obtain the forward acceleration for the return polynomial as follows:

$$a_\parallel = \frac{\boldsymbol{a}^T \boldsymbol{v}}{\|\boldsymbol{v}\|}\,. \tag{17}$$

Using Pythagoras' theorem, the centripetal acceleration for the return polynomial can then be obtained by

$$a_\perp = \sqrt{\|\boldsymbol{a}\|^2 - a_\parallel^2} \times sgn(a_x v_y - a_y v_x) \tag{18}$$

where the term $\mathrm{sgn}(a_x v_y - a_y v_x)$ is used to determine the direction of the centripetal acceleration with the x and y subscripts indicating the x and y components of the respective vector.

For obtaining the radar pointing direction, it is necessary to obtain the UAV heading. For the return polynomial, the heading is obtained as: $\psi^{\mathrm{p}} = \arctan2(p_y, p_x)$, where p_y and p_x refer to the respective x and y elements of $\boldsymbol{p}$ (the position of the UAV).

It is then necessary to determine the required thrust for a given setting of the UAV. For a fixed-wing UAV, a banked turn is used to change heading. By banking, a horizontal component of the lift force is generated, which forms the centripetal acceleration. The UAV's drag during a steady

level flight banked turn is then defined as

$$D = c_1 V^2 + \frac{c_2 n^2}{V^2} \tag{19}$$

where

$$c_1 = \frac{1}{2}\rho S C_{d_0},\ c_2 = \frac{2W^2}{\pi e_0 A R \rho S},\ n = \sqrt{1 + a_\perp^2/g^2} \tag{20}$$

where $W = m_C g$ and g is the gravitational acceleration with value 9.80665 $\mathrm{m \cdot s^{-2}}$. The required thrust is then obtained as follows:

$$T_{\text{required}} = D + m_C a_\parallel . \tag{21}$$

Equation (15) stated that the rate of fuel consumption is the product of the required thrust and the thrust specific fuel consumption. The next steps are to determine the available thrust in order to determine that the required thrust does not exceed it, and also to obtain the thrust specific fuel consumption.

The engine is assumed to be a high-bypass turbofan with an available thrust that decreases with altitude and Mach number M. The thrust available is assumed to vary linearly with Mach number given that the UAV operates within a fairly small band of Mach numbers (roughly between 0.3 and 0.6). It is assumed that at $M = 0.3$, there is 75% of the take-off thrust T_0, and at $M = 0.6$, there is 60% of T_0. These values were selected in a similar manner to the values of b_1 and b_2 while also being chosen such that they fitted within reasonable variations of thrust [28]. The variation in available thrust as a function of altitude and Mach number [29] is given by:

$$T_{\text{available}} = T_0(\mu(M - 0.3) + 0.6)\sigma_1^{b_1}\sigma_2^{b_2} \tag{22}$$

where $\sigma_1 = \rho_t/\rho_0$ with ρ_0 being the air density at sea level (with value 1.225 $\mathrm{kg \cdot m^{-2}}$) and ρ_t being the air density at the end of the troposphere (with value 0.3639 $\mathrm{kg \cdot m^{-2}}$). Similarly, $\sigma_2 = \rho/\rho_t$. Significant lack of engine data resulted in the values for b_1 and b_2 assumed to be 0.5 and 0.8, respectively. These values were selected to provide sufficient thrust at maximum altitude to maintain the maximum speed. The gradient of the thrust–Mach line is denoted by μ with value $(0.6 - 0.75)/(0.6 - 0.3)$, and $M = V/a$, where a is the speed of sound, which is assumed constant given the UAV's region of altitude operation.

The thrust-specific fuel consumption can be obtained as a function of altitude and Mach number [29]. However, the effect of altitude on TSFC is a function of temperature, and in this case (within the region of 11 000 and 20 000 m), the temperature is taken as constant. Thus, the thrust-specific fuel consumption can be calculated by

$$\text{TSFC} = \text{TSFC}_0 M^\alpha \tag{23}$$

where TSFC_0 was taken to be $2.55 \times 10^{-5}\ \mathrm{kg \cdot N^{-1} \cdot s^{-1}}$ and α was taken to be 0.6, which is typical for a high-bypass turbofan.

The value of TSFC and T_{required} can now be evaluated at any point on the whole trajectory, and thus the equations to obtain fuel consumed and change in UAV mass ((15) and (16), respectively) can be used to obtain the total fuel consumed: $m_{\text{consumed}} = m_I - m_{C_{k_w}}$, where $m_{C_{k_w}}$ is the UAV weight at the end of the trajectory.

C. Radar Coverage

To obtain the coverage of the radar, a method of representing the sensor and how it covers a given area needs to be implemented. Two aspects will be considered with regard to coverage: first, the probability of detection at a point within the area, and second, the revisit time to a point within the area. The probability of detection, as the name suggests, mathematically encapsulates the probability of detecting a specific target. The revisit time is the time between repeated coverages of a given point in the search area. As the UAV moves around the search area, the main beam of the radar will intersect the surface, thus increasing the probability of detection and resetting the current revisit time to zero, for those points within the beam. Ideally, the probability of detection is to be maximized, whereas the revisit time is to be minimized. There is thus an inherit tradeoff between the two.

In order to determine the global coverage for a given trajectory in an efficient manner, the search area was discretized into an evenly spaced x–y search grid consisting of n^g nodes (where x is along north, y is along east, and z is assumed sea level (i.e., 0 m)). The search grid is defined in terms of the node spacing ΔG (which sets in meters how far apart each node is along each axis), and the number of nodes n_x^g and n_y^g in the x and y axes, respectively. Both node numbers are taken to be odd such that there is a definite center node (note that the center of the search grid is always taken to be 0 rad in latitude and longitude and (0, 0, 0) in NED coordinates). The total number of nodes in the search grid is therefore equal to $n_x^g n_y^g$.

At each node in the search grid, the probability of detection and revisit time needs to be evaluated for the full trajectory. These components can be evaluated by first representing the probabilities of detection grid and the revisit time grid by matrices $\boldsymbol{D}$ and $\boldsymbol{T}$, respectively, both of which have dimensions $n_y^g \times n_x^g$.

For efficiency, the radar was similarly discretized into an x–y radar grid (and as before z is taken to be 0) with n^r nodes. Each node represents the probability of detecting a specific target on the ground at a given relative NED coordinate to the UAV. Furthermore, due to the relatively small size of search grids, it is assumed that the radar grid is calculated at a latitude and longitude of 0 rad. To account for the curvature of the earth, the nodes in NED coordinates are converted to earth centered, earth fixed (ECEF) coordinates to obtain the absolute range to the UAV as outlined in (24) and (25).

For both the search grid and the radar grid, it is accepted that there will be some warping effects (i.e., the spacing between nodes will not be equal), particularly near the poles of the earth. However, there is warping effects for all methods of mapping a sphere to a 2-D grid, and this effect is only of concern if the search area (or radar range) is significantly large. In this case, the diagonal distance between the cells

TABLE II
Radar Specifications

Parameter	Unit	Value
Power, P_{tx}	W	3000
Wavelength, λ	m	0.031
Antenna height, A_h	m	0.5
Antenna length, A_l	m	1
Pulse compression factor, η		185.2
Coherent integration factor, N_p		32
Noise figure, F_n	dB	7
System losses, L_{sys}	dB	7.5
Probability of false alarm, PFA		1×10^{-6}
Maximum range, $r_{\max}$	m	185 200
Scan rate, $\dot{\omega}$	$\text{rad}\,\text{s}^{-1}$	$\pi/6$
Noise Bandwidth, B_n	Hz	1.5×10^6

of the grid deviated less than 1 m when wrapped around the earth. This error was deemed well within reasonable limits.

As is often the case, the full 360° scan of the radar may not be needed. For example, if a thin coastal stretch is to be monitored, the maximum angular sector of the radar need only cover the length of the coastal strip. Accordingly, sector scan is introduced to allow for this scenario and is defined by the sector scan angular position δ and the sector scan angular width γ.

The radar grid represents a full scan of the radar (for any γ). Given the radial nature of a radar scan, the number of nodes in the x-axis is equal to the number of the nodes in the y-axis, and is given by $n^{\text{r}}_{xy} = 2\lfloor r_{\max}/\Delta G \rfloor + 1$. Thus, $n^{\text{r}} = n^{\text{r}}_{xy} n^{\text{r}}_{xy}$. The matrices containing the range, elevation from boresight, elevation from UAV, signal-to-noise ratio, and probability of detection are denoted by $\boldsymbol{r}$, $\boldsymbol{\Theta}$, $\boldsymbol{\theta}^{\text{r}}$, $\boldsymbol{SNR}$, and $\boldsymbol{P_{\text{d}}}$, respectively.

The earth is assumed to be spherical with radius $r_{\text{e}} = 6318137\,\text{m}$. For the radar modeling, the commonly used "four-thirds earth model" is used to account for atmospheric refraction, and thus, the value of the radius of the earth is given by $r_{\text{a}} = (4/3)r_{\text{e}}$. The ECEF coordinates (denoted by X, Y, and Z) of each node in the radar grid, converted from NED coordinates (denoted by x^{r} and y^{r}), are obtained as follows:

$$\begin{aligned} X_{uv} &= r_{\text{a}} \cos\left(x^{\text{r}}_{uv}/r_{\text{a}}\right) \cos\left(y^{\text{r}}_{uv}/r_{\text{a}}\right) \\ Y_{uv} &= r_{\text{a}} \cos\left(x^{\text{r}}_{uv}/r_{\text{a}}\right) \sin\left(y^{\text{r}}_{uv}/r_{\text{a}}\right) \\ Z_{uv} &= r_{\text{a}} \sin\left(x^{\text{r}}_{uv}/r_{\text{a}}\right) \\ \text{for } u &= 1, \ldots, n^{\text{r}}_{xy} \quad \text{and} \quad \text{for } v = 1, \ldots, n^{\text{r}}_{xy} \end{aligned} \tag{24}$$

where subscripts u and v indicate the row and column indices of a given node in the radar grid.

The range from the UAV to an individual radar grid node in matrix $\boldsymbol{r}$ is obtained as follows (recalling that the radar is calculated at a latitude and longitude of 0rad):

$$r_{uv} = \sqrt{(X_{uv} - h^{\text{p}})^2 + {Y_{uv}}^2 + {Z_{uv}}^2}\,. \tag{25}$$

The radar assumed here was based on a typical airborne maritime surveillance radar suitable for a long-range UAV mission. The values for this radar are outlined in Table II with the radar assumed to have a rectangular antenna, and thus the beamwidth in elevation is approximated with $\theta_{3\,\text{dB}} = 0.88(\lambda/A_h)$. The radar sensor is represented using the radar range equation, and consequently, the signal-to-noise ratio for each element in matrix $\boldsymbol{SNR}$ is given by

$$\text{SNR}_{uv} = \frac{\eta N_p P_{\text{tx}} G^2(\Theta_{uv}) \sigma_{\text{tgt}} \lambda^2}{(4\pi)^3 k_{\text{B}} T B_n F_n L_{\text{sys}} {r_{uv}}^4} \tag{26}$$

where k_{B} is Boltzmann's constant with value $1.38064852 \times 10^{-23}\,\text{J} \cdot \text{K}^{-1}$, and T is the temperature taken to be a constant 213 K. For this scenario, a typical target is assumed to be a fishing boat with radar cross section σ_{tgt} of $100\,\text{m}^2$. Note that the values quoted in Table II for F_n and L_{sys} are converted to linear values. The radar gain G is a function of elevation from radar boresight given in matrix form as

$$G(\boldsymbol{\Theta}) = \frac{4\pi A_l A_h}{\lambda^2} \operatorname{sinc}^2\left(\pi \frac{A_h}{\lambda} \sin(\boldsymbol{\Theta})\right) \tag{27}$$

where $\boldsymbol{\Theta} = \boldsymbol{\theta}^{\text{r}} - \theta_{\text{B}}$, $\boldsymbol{\theta}^{\text{r}}$ is the elevation angle to the node from the UAV, and θ_{B} is the radar boresight elevation angle.

Obtaining the elevation angle from the UAV to a node in the radar grid requires the use of the geometry outlined in Fig. 2. First, the slant range to the horizon r_{h} is simply obtained using Pythagoras' theorem by $r_{\text{h}} = \sqrt{(r_{\text{a}} + h^{\text{p}})^2 - r_{\text{a}}^2}$.

If the slant range to a point on the ground r_{c} is known, then the elevation angle from the UAV to that point can be obtained by the following:

$$\theta_{\text{c}} = \arcsin\left(\frac{r_{\text{c}}^2 + (r_{\text{a}} + h^{\text{p}})^2 - r_{\text{a}}^2}{2 r_{\text{c}} (r_{\text{a}} + h^{\text{p}})}\right). \tag{28}$$

Alternatively, for a given elevation angle θ_{c}, the slant range to a point on the ground can be calculated as follows:

$$r_{\text{c}} = (r_{\text{a}} + h^{\text{p}}) \sin\theta_{\text{c}} - \sqrt{(r_{\text{a}} + h^{\text{p}})^2 \sin^2\theta_{\text{c}} - r_{\text{h}}^2}\,. \tag{29}$$

For surveillance operations, the boresight of the radar beam is set such that it maximizes the area of coverage. This setting is achieved by pointing the lower edge of the beam at the maximum range. By setting $r_{\text{c}} = r_{\max}$ (and accounting for the beamwidth), (28) can be used to set the boresight for maximum area. Additionally, using the range to each node in the radar grid r_{uv} in conjunction with (28), the elevation from boresight for each node can also be obtained.

The $\boldsymbol{SNR}$ can now be obtained for any node, and thus the probability of detection $\boldsymbol{P_{\text{d}}}$ can be obtained (an example of which is shown in Fig. 3). Given that the target can be

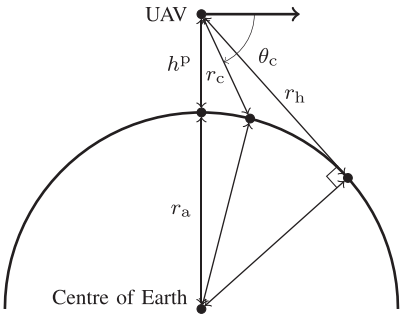

Fig. 2. Geometry of UAV radar boresight with respect to earth.

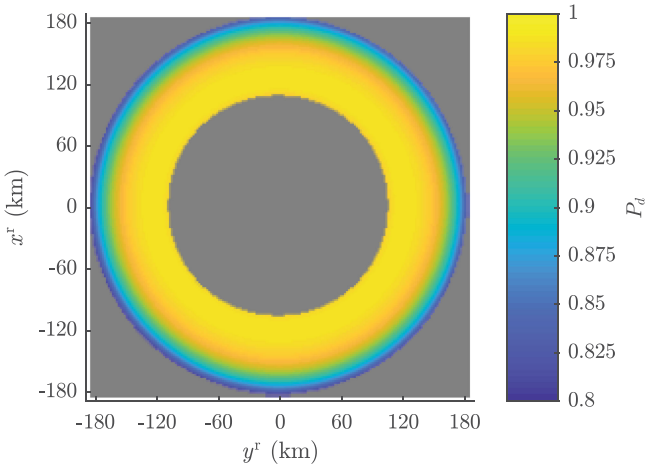

Fig. 3. Probability of detection radar grid at altitude $h^{\mathrm{p}} = 15\,000$ m. The gray area indicates excluded nodes.

approximated by Swerling 1 models, each element in the matrix can then be approximated [30] to

$$P_{\mathrm{d}uv} = \mathrm{PFA}^{(1+\mathrm{SNR}_{uv})^{-1}} . \tag{30}$$

It should be noted that in order to improve computational time, several conditions were applied during the creation of matrix $\boldsymbol{P_d}$. The first of which was to disregard the radar side lobes and ensure that only the main beam of the radar was considered. This consideration was implemented by removing any nodes that had a range r_{uv} less than the upper and lower slant ranges of the radar main beam such that $r^{-}_{3\,\mathrm{dB}} \leq r_{uv} \leq r^{+}_{3\,\mathrm{dB}}$. In order to obtain the slant ranges, (29) was used where $\theta^{-}_{3\,\mathrm{dB}} = \theta_{3\,\mathrm{dB}} + \theta_{3\,\mathrm{dB}}/2$ and $\theta^{+}_{3\,\mathrm{dB}} = \theta_{3\,\mathrm{dB}} - \theta_{3\,\mathrm{dB}}/2$. The probability of detection for the ignored nodes was 0.

When sector scan is applied, only the nodes within the sector need to be considered. The following condition can then be used on each node in the radar grid to determine if it lies within the sector: $|\psi^{\mathrm{r}}_{uv} - \psi^{\mathrm{p}} - \delta| \leq \gamma/2$.

The matrix $\boldsymbol{P_{\mathbf{d}}}$ provides the probability of detection for the radar grid at a given point in the trajectory. However, it is necessary to obtain the accumulated probability of detection at each node in the global search grid for a given trajectory. In order to calculate the accumulated probability of detection, it is first necessary to obtain the current search grid nodes that are aligned with the current radar grid nodes.

Given the long surveillance range, the maximum speed of the UAV, and the fast scan rate, it is assumed that a full sector scan (for any γ) is applied at a time step equal to the equivalent scan time. Note that the distance covered by the UAV during the time taken to scan is not significant relative to the range covered by the radar. This assumption greatly improves computational efficiency by allowing the radar to be represented by one matrix. Similar to obtaining the fuel consumption, the trajectory is discretized in τ. In this case, however, the spacing $\Delta\tau_{\mathrm{r}}$ was set to match the scan rate of the radar and was thus equal to $\gamma/\dot{\omega}$.

A given discrete position ($\boldsymbol{p}$ and h^{p}) in NED coordinates along the trajectory can be projected to determine the NED position on the ground that is equivalent to the position of the center of the radar grid. This ground position, obtained with (31), is then rounded to the nearest node on the global search grid. At this point, the radar grid will be aligned with the global search grid, and thus, the current probability of detection of the radar grid can be used to update the global search grid probability of detection (or revisit time) as

$$\boldsymbol{p}^{\mathrm{g}} = \boldsymbol{p}\frac{r_{\mathrm{e}}}{r_{\mathrm{e}} + h^{\mathrm{p}}} . \tag{31}$$

For a given discrete position of the UAV projected to the ground $\boldsymbol{p}^{\mathrm{g}}$, the lower and upper matrix indices of the global search grid that are occupied by the radar grid are given by the 1×2 vectors $\boldsymbol{m}$ and $\boldsymbol{n}$, respectively. The indices of the center node for the search grid and the radar grid are denoted by $\boldsymbol{g}^{\mathrm{g}} = \lceil [n^{\mathrm{g}}_x/2, n^{\mathrm{g}}_y/2] \rceil$ and $g^{\mathrm{r}} = \lceil n^{\mathrm{r}}_{xy}/2 \rceil$, respectively. $\boldsymbol{m}$ and $\boldsymbol{n}$ are then obtained as follows:

$$\boldsymbol{m} = \mathrm{nint}\left(\frac{\boldsymbol{p}^{\mathrm{g}}}{\Delta G}\right) + \boldsymbol{g}^{\mathrm{g}} - g^{\mathrm{r}} + 1 \tag{32}$$

$$\boldsymbol{n} = \mathrm{nint}\left(\frac{\boldsymbol{p}^{\mathrm{g}}}{\Delta G}\right) + \boldsymbol{g}^{\mathrm{g}} + g^{\mathrm{r}} - 1 \tag{33}$$

where, for example, m_1 and m_2 indicate the respective column and row index of the lower bounds of the global search grid that is within the radar grid. Note that the function nint obtains the nearest integer.

The probabilities of detection for the current radar grid can then be combined with the probabilities of detection for the global search grid. Noting that for events that are not mutually exclusive, the probability of event A or B occurring is given by $P(A \cup B) = P(A) + P(B) - P(A)P(B)$. The appropriate nodes in the probability of detection search grid can then be updated as

$$\boldsymbol{D}_{jk} \leftarrow \boldsymbol{D}_{jk} + \boldsymbol{P}_{\mathbf{d}w} - \boldsymbol{D}_{jk} \circ \boldsymbol{P}_{\mathbf{d}w} \tag{34}$$

where $j = m_1, \ldots, n_1$ and $k = m_2, \ldots, n_2$, which indicate the respective row and column indices of the search grid nodes that are within the radar grid. Note that the Hadamard product is used for elementwise multiplication. An additional check of j and k is applied to ensure that each value lies within the bounds of the search grid such that $1 \leq j \leq n^{\mathrm{g}}_y$ and $1 \leq k \leq n^{\mathrm{g}}_x$. Values that do not satisfy this condition are ignored. $\boldsymbol{P}_{\mathbf{d}w}$ indicates the radar grid probability of detection $\boldsymbol{P_{\mathbf{d}}}$ at discrete point w on the trajectory (with a time step of $\Delta\tau_{\mathrm{r}}$ used).

In order to obtain the average revisit time, three quantities are required for each node in the search area grid: the number of visits by the radar T^{n}, the total revisit time T^{t}, and the current revisit time T^{c}.

Due to the discretization of the search area, radar area, and scan time, it is possible for a node to enter coverage, then exit coverage, then re-enter coverage all within three simulation steps. While this event is rare, a smoothing technique was nevertheless employed. This technique ensures that nodes will only have their visit number incremented if there has been two steps since the last increment. The update rules for the number of visits and total revisit time for each node in the search area grid are given in the following

equations:

$$\boldsymbol{T}^{\mathrm{n}}_{pq} \leftarrow \boldsymbol{T}^{\mathrm{n}}_{pq} + 1 \tag{35}$$

$$\boldsymbol{T}^{\mathrm{t}}_{pq} \leftarrow \boldsymbol{T}^{\mathrm{t}}_{pq} + \boldsymbol{T}^{\mathrm{c}}_{pq}\,. \tag{36}$$

On the first step of the simulation, p and q are equal to j and k, respectively. These indices result in the visit count T^{n} of each search grid node within radar coverage being incremented by 1, in addition to the total visit time T^{t} being incremented by the current revisit time T^{c}. On the second step of the simulation, p and q indicate the respective row and column indices of the search grid nodes that have newly entered radar coverage. From Step 3 until the end of the simulation, p and q indicate the respective row and column indices of the search grid nodes that have newly entered coverage and have not been incremented in the previous two steps. To update the revisit time search grid, the whole grid is first incremented by the current scan time (i.e., the time that has passed since the previous scan) as follows:

$$\boldsymbol{T}^{\mathrm{c}} \leftarrow \boldsymbol{T}^{\mathrm{c}} + \Delta\tau_{\mathrm{r}}\,. \tag{37}$$

The current revisit time search nodes that are within the radar grid are then multiplied by either 0 or 1 depending on whether the probability of detection is 0 or greater than 0, respectively. The current revisit time for these nodes is therefore equated as follows:

$$\boldsymbol{T}^{\mathrm{c}}_{jk} \leftarrow \boldsymbol{T}^{\mathrm{c}}_{jk} \circ \lfloor (1 - \boldsymbol{P}_{\mathbf{d}w}) \rfloor\,. \tag{38}$$

In order to obtain the average revisit time for the whole search area, the remaining revisit time at the end of the trajectory needs to be accounted for. As such, the nodes that were covered at both the start and end of the trajectory have their visit counts reduced by 1. This criterion does not include those nodes that were within radar coverage for the full trajectory (i.e., having a visit count equal to 1). Additionally, the nodes within $\boldsymbol{T}^{\mathrm{t}}$ that were not covered at the start/end have their corresponding elements from $\boldsymbol{T}^{\mathrm{r}}$ added, with $\Delta\tau_r$ subtracted from these nodes to account for the start and end steps being equivalent. With the remaining revisit time accounted for, the average revisit time T for a given node is then simply the total revisit time divided by the number of visits: $T = T^{\mathrm{t}}/T^{\mathrm{n}}$.

III. OPTIMIZATION

A. Cost Function

The objective is to find trajectories that simultaneously minimize fuel consumption, maximize the probability of detection within the search grid, and minimize the revisit time within the search grid. In order to achieve this objective, a cost function must be defined for each of the three criteria.

The cost for probability of detection is 1 minus the mean probability of detection for a given search grid. This cost can be equated as follows:

$$J_D = 1 - \frac{1}{n^{\mathrm{g}}} \sum_{j=1}^{n^{\mathrm{g}}_y} \sum_{k=1}^{n^{\mathrm{g}}_x} \boldsymbol{D}_{jk}\,. \tag{39}$$

Similarly, the cost for revisit time is given by the mean revisit time for a given search grid

$$J_T = \frac{1}{n^{\mathrm{g}}} \sum_{j=1}^{n^{\mathrm{g}}_y} \sum_{k=1}^{n^{\mathrm{g}}_x} \boldsymbol{T}_{jk}\,. \tag{40}$$

For the fuel consumption cost, the cost is simply the mass of the fuel consumed

$$J_m = m_{\mathrm{consumed}} = m_{\mathrm{I}} - m_{\mathrm{C}_{n_\tau}}\,. \tag{41}$$

The input to the cost function was the UAV trajectory $\boldsymbol{x}$ defined in (14), where most of the upper and lower values were outlined in either Section II-A or in Table I. For a constrained optimization problem, there needs to be limits placed on all inputs. The limits for $\boldsymbol{\tau}$, $\boldsymbol{\Delta\psi}$, and $\boldsymbol{p}_S$ are heuristically chosen to allow for a singular segment to cover either a relatively short or relatively long distance.

However, certain constraints cannot be applied to the cost function input and instead must be applied within. For example, in order to ensure that the UAV does not stall, each value for the velocity along a given polynomial must be checked. The constraints that are within the cost function are outlined in (42). The first of these checks that the air speed of the UAV at each discrete point along the trajectory neither stalls nor exceeds the maximum airspeed. The second checks that the thrust is greater than or equal to zero (i.e., no reverse thrust) and less than or equal to the available thrust. The third checks that the load factor of the UAV is less than the maximum load factor. The final constraint checks that the UAV weight is not less than the minimum required UAV weight at the end of the trajectory. All of these combined are used to determine if a given polynomial is feasible

$$\begin{aligned} v_{\mathrm{stall}} \le v_w \le v_{\max}, &\quad w = 1, \ldots, k_w \\ 0 \le T_w \le T_{\mathrm{available}}, &\quad w = 1, \ldots, k_w \\ n_w \le n_{\max}, &\quad w = 1, \ldots, k_w \\ m_{\mathrm{F}} \le m_{\mathrm{C}_{k_w}}\,. & \end{aligned} \tag{42}$$

Rather than checking each discrete point, the maximum or minimum of each value (excluding the minimum load factor and maximum UAV weight) was used as the input to either (43) or (44) depending on whether the upper or lower bounds are being checked. These equations penalize the violation of a given constraint such that the further the value exceeds the bounds, the greater the violation cost. This penalty helps guide the optimization algorithm toward feasible solutions

$$c^{\mathrm{l}}(x) = \begin{cases} \epsilon\left(1 + \frac{x_{\min} - x}{x_{\max} - x_{\min}}\right)^2, & \text{if } x < x_{\min} \\ 0, & \text{otherwise} \end{cases} \tag{43}$$

$$c^{\mathrm{u}}(x) = \begin{cases} \epsilon\left(1 + \frac{x - x_{\max}}{x_{\max} - x_{\min}}\right)^2, & \text{if } x > x_{\max} \\ 0, & \text{otherwise} \end{cases} \tag{44}$$

where $c^{\mathrm{l}}(x)$ and $c^{\mathrm{u}}(x)$ are the violation costs for the lower and upper bounds, respectively. ϵ influences the magnitude of violation cost and was heuristically chosen to be 1×10^{-5}.

Recalling the issues that a polynomial in x and y has for changes in heading approximately greater than 45°, an additional violation cost is added. This cost is denoted by c_ψ and is obtained with (43) and (44), where the minimum and maximum values were set to $-45°$ and 45°, respectively.

To avoid a trajectory appearing within a trajectory, an additional violation cost was added. This cost ensured that the total absolute change in heading was less than 720°, which prevented two full 360° rotations occurring within the trajectory. The total change in heading was obtained by summing the absolute change in heading for each segment. The cost is denoted by $c_{\Delta\psi}$ and obtained with (44) with the condition changed to $x \geq x_{\max}$.

Finally, for the surveillance of a given area, it is required that the whole area is covered. Therefore, an additional violation cost is applied to ensure that every node is visited. This cost is simply the sum of the total unvisited nodes scaled by a factor of ϵ and is denoted by c_R.

Since these constraints are not applied on the input, there is no absolute guarantee that a trajectory can be obtained that will satisfy them. However, the cost for violating these constraints can be set to a degree that near guarantees that the solution will fall within the constraints (assuming such a trajectory is possible). The total violation cost J_c is then the sum of each violation cost for every polynomial in the trajectory. This cost is equated as follows:

$$J_{\mathrm{c}} = c_R + c_{\Delta\psi} + c_\psi + c_V^{\mathrm{l}} + c_V^{\mathrm{u}} + c_T^{\mathrm{l}} + c_T^{\mathrm{u}} + c_n^{\mathrm{u}} + c_{m_{\mathrm{C}}}^{\mathrm{l}} \tag{45}$$

where, for example, c_V^{l} is the violation cost for the lower bounds of the UAV airspeed V.

The vector of cost functions is given in (46). Only solutions that do not violate the constraints within the cost function are desired and thus the violation cost is added to the cost vector. This approach ensures that feasible solutions are considered more optimal than nonfeasible solutions

$$\boldsymbol{J} = [J_D, J_T, J_m] + J_{\mathrm{c}}. \tag{46}$$

The multiobjective optimization problem is then formulated as follows:

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & \boldsymbol{J}(\boldsymbol{x}) \\ \text{subject to} \quad & \tau_{\min} \leq \tau_i \leq \tau_{\max},\ i = 1, \ldots, n_{\mathrm{h}} \\ & \Delta\psi_{\min} \leq \Delta\psi_i \leq \Delta\psi_{\max},\ i = 1, \ldots, n_{\mathrm{h}} \\ & \boldsymbol{p}_{\min} \leq \boldsymbol{p}_S \leq \boldsymbol{p}_{\max} \\ & 0 \leq \mu_V \leq 1 \\ & \psi_{S_{\min}} \leq \psi_S \leq \psi_{S_{\max}} \\ & 1 \leq k_\tau \leq 1.1579 \\ & h_{\min}^{\mathrm{p}} \leq h^{\mathrm{p}} \leq h_{\max}^{\mathrm{p}}. \end{aligned} \tag{47}$$

B. Multiobjective PSO

Outlined by Kennedy and Eberhart [31], PSO is based on social behavior of swarms and can be used for global optimization. For a swarm of p particles, the position of particle d within the variable space is given by $\boldsymbol{x}^d$. At each iteration, the position is updated with a velocity based on its own best position as well as the best position of the swarm. The position update is equated as follows:

$$\boldsymbol{x}_{k+1}^d = \boldsymbol{x}_k^d + \boldsymbol{v}_{k+1}^d \tag{48}$$

where the velocity is updated as follows:

$$\boldsymbol{v}_{k+1}^d = \omega \boldsymbol{v}_k^d + c_1 \boldsymbol{r}_1 \circ \left(\boldsymbol{p}_k^d - \boldsymbol{x}_k^d\right) + c_2 \boldsymbol{r}_2 \circ \left(\boldsymbol{p}_k^g - \boldsymbol{x}_k^d\right) \tag{49}$$

where c_1 and c_2 represent the cognitive learning factor and the social learning factor, respectively. In essence, these represent the weighting of the attraction between particle d's best position ($\boldsymbol{p}_k^d$) and the swarm's best position ($\boldsymbol{p}_k^g$), both of which are the best positions in the variable space from all previous iterations. ω is the inertia value, which weights the effect of the previous velocity. $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ are vectors of the same length as x, containing random numbers with range [0,1] with uniform distribution.

The original PSO algorithm only optimizes for one variable, so given the nature of the problem, a multiobjective PSO (MOPSO) algorithm was required. The algorithm selected for in this instance is OMOPSO [32], which was shown to be the most salient [33] of several MOPSO algorithms. Given that more than one cost function is being optimized for, there is more than one value that could be considered optimal. The set of solutions that are in some sense optimal is known as the Pareto front. OMOPSO uses crowding distance to reduce the size of the Pareto front and also uses mutation to diversify the swarm search. For the mutation, the swarm is subdivided into three sets of equal size with each subset having a different mutation scheme applied (no mutation, uniform mutation, and nonuniform mutation). The global best used in (49) is then selected using the particle with the least particles dominated, and in the case that multiple particles meet this criteria, then the particle with the maximum crowding distance is chosen (failing a singular choice, a particle is then chosen randomly from those that meet both criteria).

For high-dimensional search spaces, the initialization of the particles can have a significant effect. A method to improve the initialization is outlined by Richards and Ventura [34], which uses centroidal Voronoi tessellations. The parameters used within the OMOPSO algorithm are outlined in Table III.

TABLE III
Multiobjective PSO parameters

Parameter	Value
c_1	1.5
c_2	1.5
ω	0.3
Maximum iterations	1000
Swarm size	3600
Maximum Pareto front size	250
Mutation probability	1/length($\boldsymbol{x}$)

IV. RESULTS

This section outlines the results for two commonly seen scenarios for airborne maritime radar surveillance. The first, Scenario A, is a large square area, whereas the

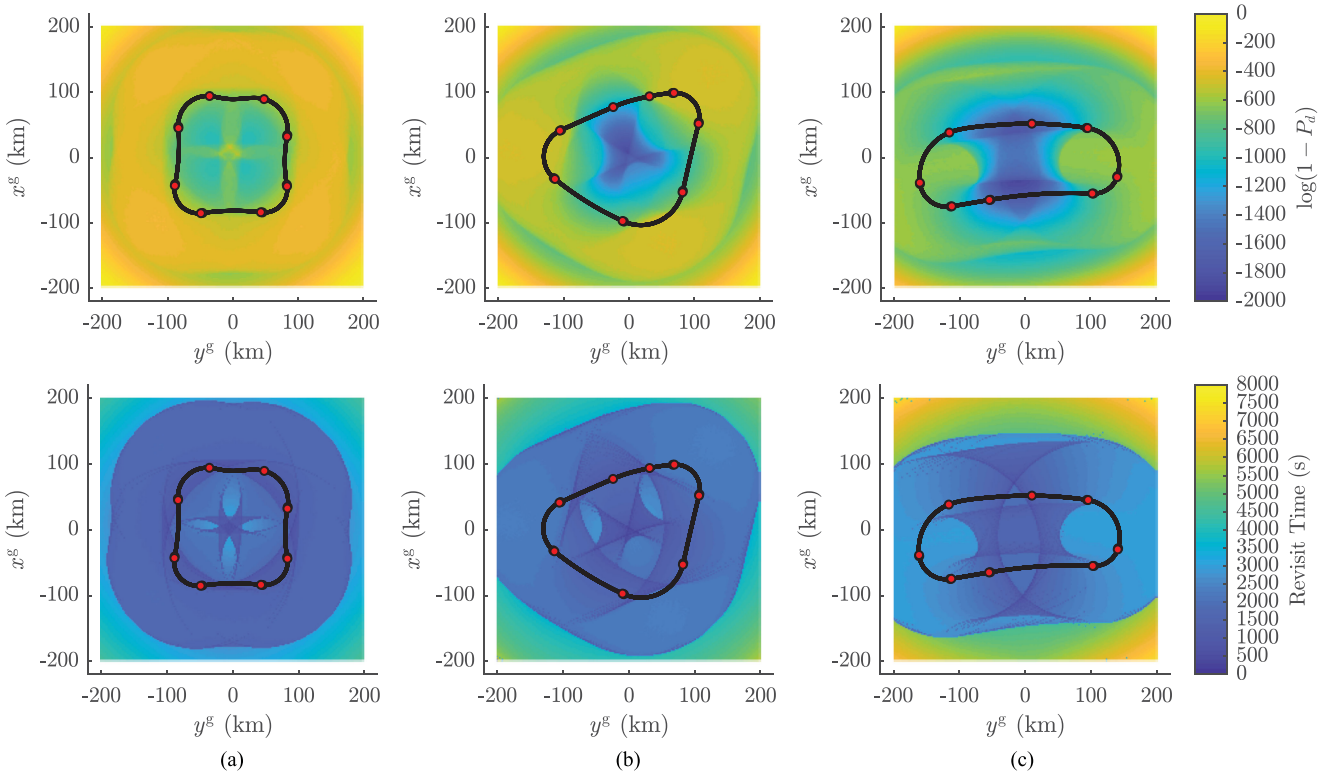

Fig. 4. Pareto front for Scenario A. Selected values shown in Fig. 5 are indicated by larger dots with the baseline trajectory indicated by the square.

second, Scenario B, is a thin curved coastal strip. For these scenarios, the number of heading polynomials n_h was taken to be six and seven, respectively with a grid spacing ΔG taken to be 2000 m. Furthermore, it is assumed for both scenarios that the radar was stabilized against platform motion.

For each of these scenarios, a Pareto front is obtained with several examples plotted. The Pareto front would allow a trajectory to be selected for a given mission based on further criteria (e.g., minimum revisit times, minimum probability of detection, and total surveillance time required).

The results were obtained with the use of MATLAB used in conjunction with an i7-6700 processor and 16 GB of RAM. For each scenario, the simulation was run several times in order to ensure repeatability amongst the solutions. The average execution time for Scenario A was 709 171 s, whereas for Scenario B, it was 611 845 s. Whilst these times show the significant computational cost, this method is intended to be used with predefined search areas where the optimization is performed once in advance. The execution time could be improved by using more powerful hardware, a compiled programming language, and/or reducing several of the scenario and optimization parameters.

A. Scenario A

The scenario being optimized for in this case involves a large rectangular grid, specifically 400 km × 400 km, such that the full radar beam ($\gamma = 360^\circ$) fits within the area. For this scenario, the constraints for τ_i were heuristically chosen as $\tau_{\min} = 300$ s and $\tau_{\max} = 2500$ s.

For this search area, constraining the start position to one quadrant of the area has zero effect on the results. As such, the upper and lower limits on $\boldsymbol{p}_S$ were simply set to 200000 m and 0 m, respectively. For the same reason, the upper and lower limits for the initial heading ψ_S were set to $\pi/4$ and $-3\pi/4$ (i.e., along the diagonal of the search area). Additionally, and since trajectories with a total absolute change in heading greater than 720° were undesired, the limits of $\Delta\psi$ were heuristically set to π and $-\pi/4$ to prevent the trajectory doubling back on itself.

The Pareto front obtained from the optimization is shown in Fig. 4. It was observed that the majority of the trajectories fell under three broad archetypes based on their geometric similarity. Each archetype consists of a series of segments, which alternate between a large turn and a small turn (or no turn). The archetypes are differentiated by the number of turning segments, which is 2, 3, and 4 for the racetrack, the rounded triangle, and the rounded square, respectively. The fact that every trajectory within each archetype shared a common geometric theme is an interesting result as opposed to their being a large variety of differing solutions. Furthermore, the geometrical simplicity of the archetypes would make them more likely to be accepted for practical use over intricate trajectories that are far removed from the current industry standards.

In general, each archetype occupied an area within the Pareto front. Specifically, for the probability of detection,

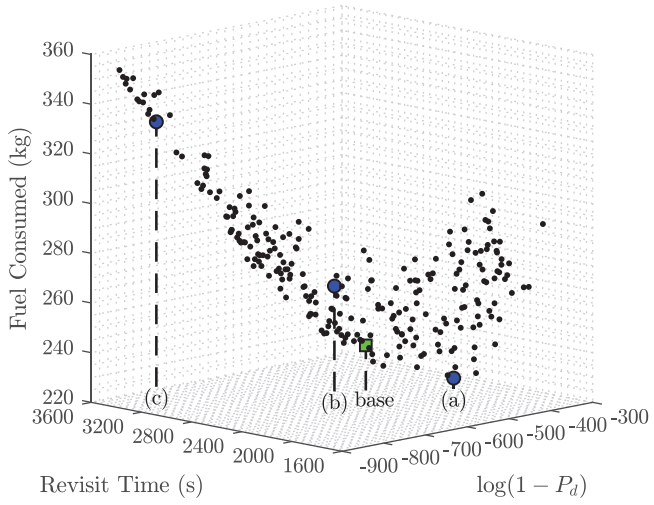

Fig. 5. Selected Pareto front probability of detection grids (top row) and revisit time grids (bottom row) for Scenario A. The black line indicates the UAV's path, whereas the dots indicate the transition points between polynomials. Note that a lower value in the probability of detection grid means higher probability of detection, whereas a lower value in the revisit time grid means more frequent revisit.

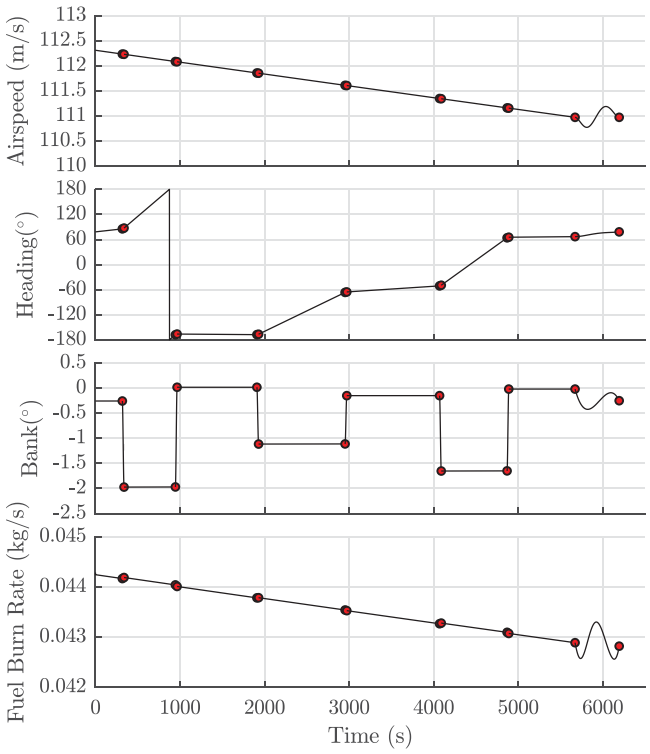

Fig. 6. Time history for trajectory (*b*), scenario A. The dots indicate the end points of each polynomial. Note that the timescale hides the continuous nature of the transition between polynomials.

revisit time, and fuel consumption, the rounded square offered low to medium values for each. Additionally, the rounded triangle provided medium to high values, whereas the racetrack only offered high values. Three trajectories were selected from the Pareto front to highlight these archetypes as shown in Fig. 5.

Trajectory (*a*) shows a rounded square path, which provides low fuel consumption and low revisit times at the cost of low probability of detection. By flying close to the center of the search area, the total distance traveled is minimized. For this trajectory, the UAV flies at an altitude of 12 715 m and an airspeed of 128.84 m·s^{-1}, which is slightly above than the airspeed for maximum endurance. This airspeed further reduces the fuel consumption while also improving the revisit time.

Trajectory (*b*) shows a rounded triangle path, which has both a high probability of detection and low revisit time at the cost of high fuel consumption. The path covers a large distance thus increasing the probability of detection while also maintaining a short distance to the center, which lowers the revisit time. With an altitude of 11 021 m, the area covered by the radar is nearly maximized at the cost of fuel consumption. However, the UAV flies at an airspeed of 112.31 m·s^{-1}, which is just below the maximum endurance speed. This speed helps minimize the fuel lost due to the low altitude and large path.

Trajectory (*c*) shows a large but narrow racetrack path. The narrowness allows for the radar to cover regions of the area to both sides at a given point in time. However, the path must travel nearly the length of the area in order to cover the central regions. This results in a high probability of detection at the cost of higher revisit time and higher fuel consumption. Similar to trajectory (*b*), the altitude is 11 007 m, which maximizes the area covered by the radar. Conversely, the UAV flies at an airspeed of 93.07 m·s^{-1}, which is 19.33 m·s^{-1} lower than the maximum endurance speed. As a result of this speed, the probability of detection is further increased with both the revisit time and fuel consumption increased.

To highlight a trajectory behavior, the time history of trajectory (*b*) is shown in Fig. 6. Most notably, there are fluctuations in the return segment polynomials, which is the main drawback of the x and y polynomials. However, this drawback is accepted due to the need for the UAV to return to the initial position with the same heading as the initial heading.

In order to highlight the performance of these trajectories within this scenario, a baseline trajectory was used for comparison. Based on industry recommendations, a circular trajectory was used as a baseline with the diameter equal to half the length of the square area. Additionally, the UAV was flown at the maximum endurance speed at an altitude of 11 000 m. It was found that three trajectories from the Pareto front were better than the baseline trajectory with regards to all three cost functions. This few number of trajectories shows that the baseline is not too far from the optimized trajectories, though there is still improvement that can be made. For example, one of these trajectories offered a 2.54% decrease in revisit time, a 2.40% decrease in fuel consumption, and a 0.16% increase in probability of detection. Additionally, the Pareto front offers trajectories in any direction in terms of cost (e.g., a trajectory that saves more fuel at the cost of probability of detection).

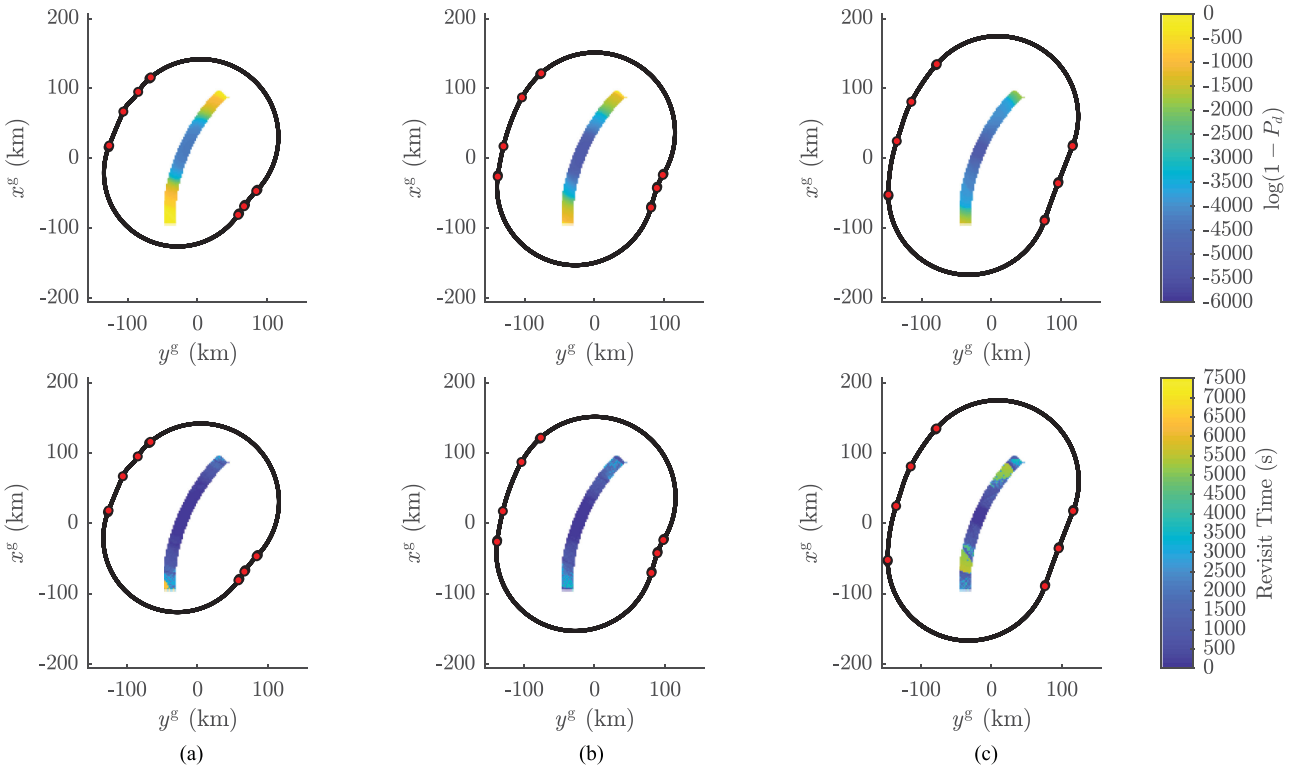

Fig. 7. Selected Pareto front probability of detection grids (top row) and revisit time grids (bottom row) for Scenario B. The black line indicates the UAV's path, whereas the dots indicate the transition points between polynomials. Note that a lower value in the probability of detection grid means higher probability of detection, whereas a lower value in the revisit time grid means more frequent revisit.

B. Scenario B

In this scenario, the trajectory is to be optimized for the surveillance of a coastal strip. A curved coastal strip was set up where the curvature was assumed to be circular. For this scenario, it is commonly required for the radar to operate with a sector scan pointing to one side of the UAV. In this case, the angular position δ was set to $\pi/2$ rad, the angular width γ set to $\pi/3$ rad, and the scan rate set to $\pi/18$ rad $\cdot$ s^{-1}. For this case, the trajectory will simply move around the coastal strip. As a result, several heuristics were employed to reduce the search space and thus execution time. The starting position $\boldsymbol{p}_S$ was constrained to start at some point on the northwest side of the coast with the initial heading ψ_S fixed in the direction of the coastal curvature. The upper and lower bounds of τ_i and $\Delta\psi_i$ were then set such that there would be two large segments (for turning around the coastal strip) and several smaller segments on either side. The turning segments were heuristically set with the following lower and upper bounds: $\tau_{\min} = 1500$ s, $\tau_{\max} = 3500$ s, $\Delta\psi_{\min} = -190°$, and $\Delta\psi_{\max} = -160°$. Similarly, for the smaller segments, the lower and upper bounds were set as $\tau_{\min} = 100$ s and $\tau_{\max} = 1000$ s, whereas the bounds for $\Delta\psi_i$ were set such that the value would be between $0°$ and $\pm 30°$.

The Pareto front for this scenario is shown in Fig. 8, with three selected trajectories shown in Fig. 7. Trajectory (a) provides low fuel consumption and low revisit times at the cost of lower probability of detection. The UAV maintains a close distance to the search area, which means that a smaller area of the radar beam intersects the area. This close proximity results in a higher revisit time as less search area is covered by the radar at each point along the trajectory. In this trajectory, the UAV flies at an altitude of 110 059 m,

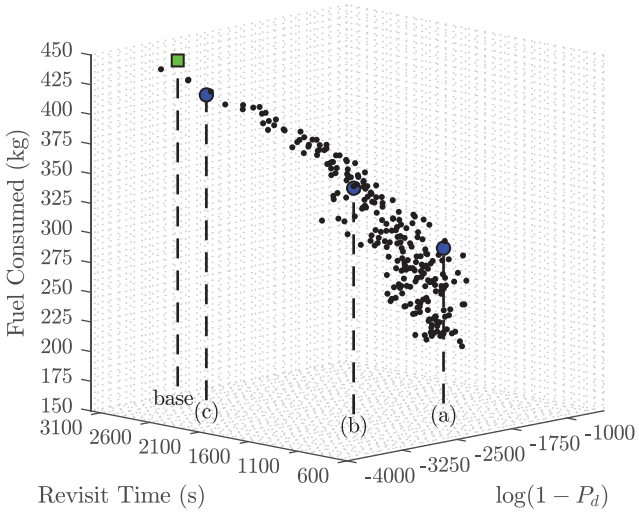

Fig. 8. Pareto front for Scenario B. Selected values shown in Fig. 7 are indicated by larger dots with the baseline trajectory indicated by the square.

which allows for the radar beam to intersect the surface at closer distances, thus allowing the UAV to maintain a close course. Additionally, the UAV flies at 125.23 m·s^{-1}, which is well above the maximum endurance speed. The result of this speed is a sacrifice of some fuel for a decrease in revisit time.

Trajectory (*b*) provides both high probability of detection and low revisit time at the cost of high fuel consumption. By maintaining a distance that allows for the center beam to intersect the search area, the probability of detection is maximized. Furthermore, the UAV flies at a low altitude (11006 m), which further increases the probability of detection by reducing both the distance to the beam center and the range. With an airspeed (100.61 m·s^{-1}) less than the airspeed for maximum endurance, there is more time on area for the radar at the cost of fuel consumption.

Trajectory (*c*) provides high probability of detection by sacrificing fuel consumed and revisit time. The UAV flies at an altitude of 11 000 m, which provides the maximum probability of detection for the radar. With an airspeed of 92.91 m·s^{-1} (well below the maximum endurance speed), more time is spend on each point within area, thus maximizing the probability of detection at the cost of fuel.

In terms of path, the biggest difference between trajectories (*a*), (*b*), and (*c*) is the distance between the turns at the tips. Trajectory (*a*) turns sharply around the tips, to the point where there is significantly less probability of detection at the tips. Conversely, trajectory (*c*) keeps a distance that allows the radar beam to intersect more of the outer edges of the search area. However, by keeping a far distance from the edges, the flight time is significantly increased, which increases the revisit time and the fuel consumed. Trajectory (*b*) provides a medium between (*a*) and (*c*).

The time history of trajectory (*b*) is shown in Fig. 9. Notably, there are larger fluctuations in the return polynomial relative to the fluctuations shown in Fig. 6. This difference is largely due to the return polynomial having more curvature relative to the return polynomial in trajectory (*b*) in Scenario A. However, the fluctuation in airspeed for this case is still less than 1 m·s^{-1}, which has little impact on performance.

The industry recommended trajectory for this scenario was such that the center of the beam intersects the center line of the curved strip. Additionally, circular turns are performed at the ends of the area with the beam still intersecting the center line. As with Scenario A, the UAV was flown at the maximum endurance speed at an altitude of 11 000 m. It was found that 97 trajectories from the Pareto front were better than the baseline in all three cost functions. This number suggests a great improvement in performance. For example, one of these trajectories offered a 74.33% decrease in revisit time, a 9.05% decrease in fuel consumption, and a 26.71% increase in probability of detection.

In both Scenarios A and B, none of the trajectories had an altitude above 14 500 m. This observation suggests that at higher altitudes, too much fuel is consumed in making up for the lost coverage than would otherwise be saved flying at these high altitudes.

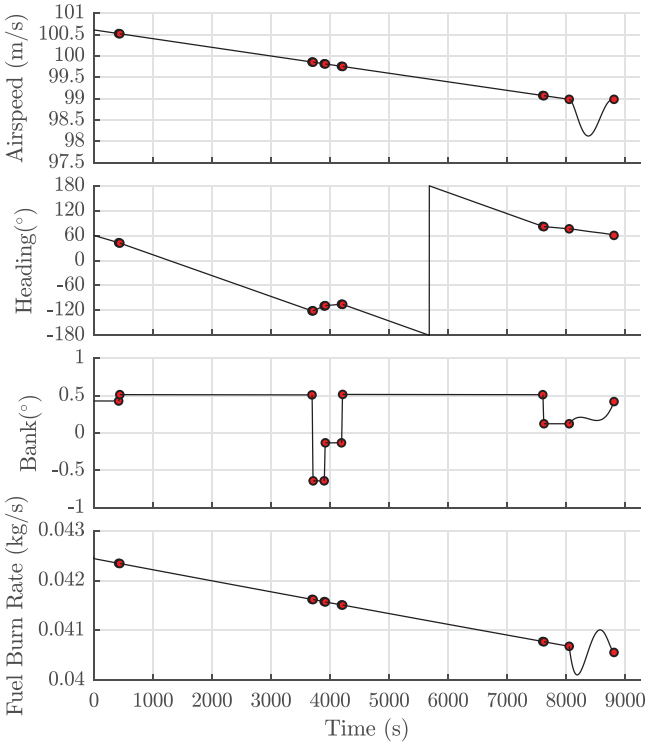

Fig. 9. Time history for trajectory (*b*), scenario B. The dots indicate the end points of each polynomial. Note that the timescale hides the continuous nature of the transition between polynomials.

V. CONCLUSION

This article outlines a method for trajectory optimization for airborne maritime radar wide area persistent surveillance using a polynomial trajectory generation method. The considerations of the dynamics, propulsion, and mission requirements of a fixed-wing UAV, as well as a maritime surveillance radar, provide a method to obtain the fuel consumption, probability of detection, and revisit time for a given trajectory. The polynomial trajectory generation method provides a simple method to produce complex trajectories necessary to obtain the UAV dynamics for the fuel consumption, dynamic limitations of the UAV, and path. By discretizing the search area and radar coverage area into grids, a computationally efficient way of obtaining the probability of detection and revisit time for each point in the grid is outlined.

Multiobjective PSO was used in conjunction with the cost function for probability of detection, revisit time, and fuel consumption, resulting in a Pareto front of trajectories that provide several suitable options for various UAV maritime radar surveillance mission requirements. The results are not just in terms of path, but also in terms of the altitude of operation and airspeed of the UAV. Results are obtained for two commonly seen scenarios. For Scenario A, the results showed repeated geometrically similar paths within the Pareto front as opposed to a variety of differing paths. Additionally, three trajectories were found that were better than the industry recommended baseline in all three cost

functions. For Scenario B, 97 trajectories were found to be better than the baseline, which suggests a significant improvement. For Scenario B, the main advantage of the trajectories on the Pareto front is their sharp turns around the corner, increasing the relative time spent covering the long sides of the search area.

ACKNOWLEDGMENT

The authors would like to thank both Leonardo and Douglas Thomson for their assistance in this work.

REFERENCES

[1] Z. Goraj *et al.*
High altitude long endurance unmanned aerial vehicle of a new generation—A design challenge for a low cost, reliable and high performance aircraft
Bull. Polish Acad. Sci., Tech. Sci., vol. 52, no. 3, pp. 173–194, 2004.

[2] Leonardo MW Ltd, Private Communication, 2018.

[3] J. Hall and D. Anderson
Reactive route selection from pre-calculated trajectories—Application to micro-UAV path planning
Aeronaut. J., vol. 115, no. 1172, pp. 635–640, 2011.

[4] D. Delahaye, S. Puechmorel, P. Tsiotras, and E. Feron
Mathematical Models for Aircraft Trajectory Design: A Survey (Lecture Notes in Electrical Engineering), vol. 290. New York, NY, USA: Springer, 2014, pp. 205–247.

[5] K. Bousson and T. A. Gamerio
A quintic spline approach to 4D trajectory generation for unmanned aerial vehicles
Int. Rev. Aerosp. Eng., vol. 8, no. 1, pp. 1–9, 2015.

[6] I. D. Cowling, O. A. Yakimenko, J. F. Whidborne, and A. K. Cooke
A prototype of an autonomous controller for a quadrotor UAV
In *Proc. Eur. Control Conf.*, 2007, pp. 4001–4008.

[7] A. Barrientos, P. Gutiérrez, and J. Colorado
Advanced UAV trajectory generation: Planning and guidance
In *Aerial Vehicles*, T. M. Lam Ed. Rijeka, Croatia: InTech, 2009, pp. 55–82.

[8] L. Babel
Three-dimensional route planning for unmanned aerial vehicles in a risk environment
J. Intell. Robot. Syst., vol. 71, no. 2, pp. 255–269, 2013.

[9] B. Salamat and A. M. Tonello
Stochastic trajectory generation using particle swarm optimization for quadrotor unmanned aerial vehicles (UAVs)
Aerospace, vol. 4, no. 2, 2017, Art. no. 27.

[10] J. L. Foo, J. S. Knutzon, J. H. Oliver, and E. H. Winer
Three dimensional path planning of unmanned aerial vehicles using particle swarm optimization
In *Proc. 11th AIAA/ISSMO Multidisciplinary Anal. Optim. Conf.*, Sep. 2006, pp. 1–10.

[11] J. Wang, L. Liu, T. Long, and Z. Wang
Three-dimensional constrained UAV path planning using modified particle swarm optimization with digital pheromones
In *Proc. 3rd Int. Conf. Eng. Optim.*, 2012.

[12] J. Karimi and S. H. Pourtakdoust
A real-time algorithm for variable-objective motion planning over terrain and threats
Proc. Inst. Mech. Eng. G, J. Aerosp. Eng., vol. 229, no. 6, pp. 1043–1056, 2015.

[13] E. Besada-Portas, L. De La Torre, J. M. De La Cruz, and B. De Andrés-Toro
Evolutionary trajectory planner for multiple UAVs in realistic scenarios
IEEE Trans. Robot., vol. 26, no. 4, pp. 619–634, Aug. 2010.

[14] M. B. Pellazar
Vehicle route planning with constraints using genetic algorithms
In *Proc. Nat. Aerosp. Electron. Conf.*, 1994, pp. 111–118.

[15] J. L. Foo, J. Knutzon, V. Kalivarapu, J. Oliver, and E. Winer
Path planning of unmanned aerial vehicles using B-splines and particle swarm optimization
J. Aerosp. Comput., Inf., Commun., vol. 6, no. 4, pp. 271–290, 2009.

[16] R. Hassan, B. Cohanim, O. de Weck, and G. Venter
A comparison of particle swarm optimization and the genetic algorithm
In *Proc. 46th AIAA/ASME/ASCE/AHS/ASC Struct., Struct. Dyn. Mater. Conf.*, 2005, Paper AIAA 2005-1897.

[17] M. Bagherian and A. Alos
3D UAV trajectory planning using evolutionary algorithms: A comparison study
Aeronaut. J., vol. 119, no. 1220, pp. 1271–1285, 2015.

[18] E. Besada-Portas, L. De La Torre, A. Moreno, and J. L. Risco-Martín
On the performance comparison of multi-objective evolutionary UAV path planners
Inf. Sci., vol. 238, pp. 111–125, 2013.

[19] A. A. Heidari, A. Afghan-Toloee, and R. A. Abbaspour
Path planning of an autonomous mobile multi-sensor platform in a 3D environment using newtonian imperialist competitive optimization method
Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci., vol. XL-1/W3, pp. 191–196, 2013.

[20] Y. Fu, M. Ding, C. Zhou, and H. Hu
Route planning for unmanned aerial vehicle (UAV) on the sea using hybrid differential evolution and quantum-behaved particle swarm optimization
IEEE Trans. Syst., Man, Cybern., Syst., vol. 43, no. 6, pp. 1451–1465, Nov. 2013.

[21] H. Ergezer and K. Leblebicioglu
3D path planning for multiple UAVs for maximum information collection
J. Intell. Robot. Syst., vol. 73, no. 1, pp. 737–762, 2014.

[22] H. Ergezer and K. Leblebicioglu
Path planning for UAVs for maximum information collection
IEEE Trans. Aerosp. Electron. Syst., vol. 49, no. 1, pp. 502–520, Jan. 2013.

[23] S. Perez-Carabaza, E. Besada-Portas, J. A. Lopez-Orozco, and J. M. de la Cruz
A real world multi-UAV evolutionary planner for minimum time target detection
In *Proc. Genetic Evol. Comput. Conf.*, 2016, pp. 981–988.

[24] S. Perez-Carabaza, E. Besada-Portas, J. A. Lopez-Orozco, and J. M. de la Cruz
Ant colony optimization for multi-UAV minimum time search in uncertain domains
Appl. Soft Comput. J., vol. 62, pp. 789–806, 2018.

[25] J. Li, J. Chen, P. Wang, and C. Li
Sensor-oriented path planning for multiregion surveillance with a single lightweight UAV SAR
Sensors, vol. 18, no. 2, 2018, Art. no. E548.

[26] Office of the Under Secretary of Defence, UAV Annual Report, FY 1997
Off. Under Secretary Def., Washington, DC, USA, Tech. Rep., 1997. [Online]. Available: http://www.dtic.mil/dtic/tr/fulltext/u2/a336710.pdf

[27] Y. Zeng and R. Zhang
Energy-efficient UAV communication with trajectory optimization
IEEE Trans. Wireless Commun., vol. 16, no. 6, pp. 3747–3760, Jun. 2017.

[28] J. B. Russell
Standard atmospheres
In *Performance & Stability of Aircraft*. London, U.K.: Butterworth–Heinemann, 1996, ch. 1.5, pp. 15–17.

[29] M. E. Eshelby
Aircraft Performance: Theory and Practice. London, U.K.: Butterworth–Heinemann, 2000.

[30] M. A. Richards, J. A. Scheer, and W. A. Holm
Swerling 1 Target Model
In *Principles of Modern Radar: Basic Principles*. Walnut, CA, USA: SciTech, 2010, ch. 3.3.8.2, pp. 107–108.

[31] J. Kennedy and R. Eberhart
Particle swarm optimization
In *Proc. IEEE Int. Conf. Neural Netw.*, 1995, vol. 4, pp. 1942–1948.

[32] M. R. Sierra and C. A. Coello Coello
Improving PSO-based multi-objective optimization using crowding, mutation and ϵ-dominance
In *Proc. Int. Conf. Evol. Multi-Criterion Optim.*, 2005, pp. 505–519.

[33] J. Durillo, J. García-Nieto, A. J. Nebro, C. A. Coello Coello, F. Luna, and E. Alba
Multi-objective particle swarm optimizers: An experimental comparison
In *Proc. Evol. Multi-Criterion Optim.*, 2009, pp. 495–509.

[34] M. Richards and D. Ventura
Choosing a starting configuration for particle swarm optimization
In *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2004, vol. 3, pp. 2309–2312.

Angus Brown received the M.Eng. degree in aerospace systems from the University of Glasgow, Glasgow, U.K., in 2016, where he is currently working toward the Ph.D. degree in aerospace engineering with the University of Glasgow.

His Ph.D. research involves unmanned aerial vehicle trajectory optimization and the applications of artificial intelligence and machine learning algorithms to improve autonomy for surveillance radar.

David Anderson received the B.Eng. and the Ph.D. degrees in aeronautical engineering from the University of Glasgow, Glasgow, UK, in 1994 and 1997 respectively. He is currently a Senior Lecturer in Aerospace Sciences with the University of Glasgow, Glasgow, U.K. He is the Director of Glasgow's Micro Air Systems Technologies Laboratory, a combined rapid prototyping and testing facility for unmanned aerial vehicles. He is also an Architect and the Principal Developer of MAVERIC, a multiagent, multiresolution simulation environment. His research centers on engineering simulation and control, with particular application to aerospace and remote-sensing systems.