

Research Article

An Efficient Energy Constraint Based UAV Path Planning for Search and Coverage

German Gramajo and Praveen Shankar

Department of Mechanical and Aerospace Engineering, California State University, Long Beach, 1250 N. Bellflower Blvd., Long Beach, CA 90840, USA

Correspondence should be addressed to Praveen Shankar; praveen.shankar@csulb.edu

Received 5 October 2016; Revised 10 January 2017; Accepted 22 March 2017; Published 30 April 2017

Academic Editor: Paul Williams

Copyright © 2017 German Gramajo and Praveen Shankar. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A path planning strategy for a search and coverage mission for a small UAV that maximizes the area covered based on stored energy and maneuverability constraints is presented. The proposed formulation has a high level of autonomy, without requiring an exact choice of optimization parameters, and is appropriate for real-time implementation. The computed trajectory maximizes spatial coverage while closely satisfying terminal constraints on the position of the vehicle and minimizing the time of flight. Comparisons of this formulation to a path planning algorithm based on those with time constraint show equivalent coverage performance but improvement in prediction of overall mission duration and accuracy of the terminal position of the vehicle.

1. Introduction

The increased interest in UAVs has seen their implementation in military and civilian operations. Small inexpensive autonomous aerial vehicles are of great interest in search and coverage, surveillance, border patrol, and mapping missions [1, 2]. These missions require repetitive aerial maneuvers in order to locate objects/targets as soon as possible in a region or to generate a collage of a specified region. Due to the repetitive nature of the mission, small autonomous aerial vehicles are the clear choice to perform these tedious missions. The repetitive nature allows for the automation of the mission that reduces cost of the mission and allows for a faster and more effective coverage of an area [1–3].

The primary challenge in implementing small autonomous aerial vehicles for a search and coverage mission is planning the path of the vehicle that will effectively cover the specified region. This requires the development of an algorithm that will always generate trajectories to maximize the spatial coverage for any specified conditions. Different approaches exist to deal with the search and coverage problem. These fall broadly into two categories: standard search patterns and nonstandard search patterns. Standard patterns include those such as spiral and serpentine/grid (boustrophedon motion) [4, 5]. Even though standard search

patterns have proven useful to search an area, they are not ideal since they do not suit situations where multiple vehicles are cooperating to accomplish the task. On the other hand, nonstandard search patterns that have a random trajectory are easier to program in cooperative search and coverage scenario. Some algorithms used to generate the nonstandard search patterns are A^* and traveling salesmen, which are heuristic techniques [6]. A heuristic technique optimizes the trajectory based on the cost to reach the current state and the cost to reach the goal from the current state. Other forms of search techniques used are probabilistic, which use the probability of the location of an object/target in a region, requiring some prior information of the bounded region [7, 8]. In this form of search the user must provide the number of targets in the bounded region and the probability of the distribution of the targets/objects in the bounded region from prior survey information. This time of search method is primarily used in military application in which satellite information can provide the likelihood of the location of objects/targets. The UAV would primarily focus on sections with the greatest probability of identifying object/targets and then searches the regions with the least probability of a target/object.

The foundation for the primary contribution in this paper is the algorithm defined in [9, 10] for generating a trajectory

that maximizes spatial-temporal coverage based on a preset of turning rates and preset mission duration. It is a heuristic algorithm and a modified version of A* that quantifies the percentage of area covered by each of the possible paths from the current state and the cost to reach the desired exit state from the end of the current path being calculated. The path selected is the one with the lowest value of the sum of the two costs. The algorithm utilizes a receding horizon control (RHC) formulation to generate the trajectory which includes a feedback to account for any disturbance that may deviate the vehicle from its predicted path. There are several shortcomings of the formulation presented in [9]. It uses set mission duration with an assumption of constant power consumption by the vehicle during the mission. The power consumption of the vehicle is not constant since it varies per the maneuver and turn rate. Secondly, the algorithm selects the path using a discrete set of turning rates that must be specified a priori. Even though the algorithm can potentially generate the most optimal trajectory, an exhaustive simulation analysis is required to obtain the appropriate set of turning rates resulting in multiple iterations. Moreover, the optimal discrete set of turning rates may only be optimal for a set of boundary conditions. This requires the user to iterate the computation of the best discrete set of turning rates for each possible specified condition. Finally, the algorithm requires the discretization of the search region to compute the amount of the area covered by the vehicle as it performs the mission. This can result in an overestimation of the actual area covered by the vehicle since it considers a discrete space as fully covered if the sensor footprint covers the center of the each discrete. This can be overcome by the varying amount and size of each discrete space. However, finding the appropriate space discretization is computationally expensive.

This paper proposes a path planning formulation of a single UAV that can be implemented in real-time which was first presented in [11]. The formulation allows for maximizing the spatial coverage of an area under the constraints of limited energy and power consumption that is dependent on the maneuver. There are three main features of the proposed path planning formulation. First, the total energy available is explicitly modeled as a constraint in the optimization and is updated every time step. Second, the method determines the path based on the vehicle's maneuverability capability. This maneuverability capability, defined by the maximum load factor for a sustained turn, allows the path planning algorithm to utilize a range of turning rates rather than a discrete set. Third, the formulation utilizes a novel cost function that prioritizes the movement of the vehicle towards uncovered areas within the region. Further, a Boolean operation-based polygon area calculation to update the region covered by the UAV is utilized to improve accuracy. Additionally, it overcomes the limitation of the user choosing an appropriate discretization of the space.

The rest of the paper is organized as follows: Section 2 provides a brief background of the path planning problem. Section 3 discusses the formulation of the path planning algorithm that is the primary contribution of the paper. Section 4 provides the simulation platform for the implementation of the path planning algorithm while Section 5 presents the

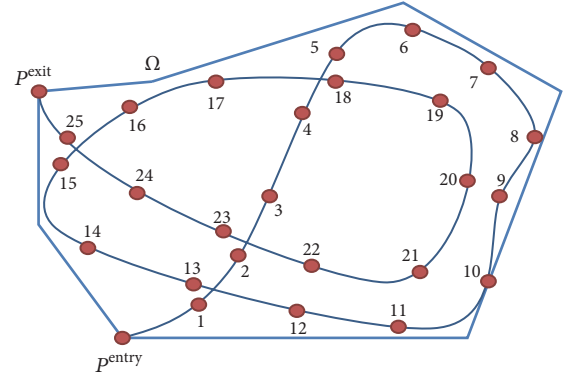


FIGURE 1: Problem of generating a trajectory that maximizes the area covered.

results and discussion of the implementation and how each of the features of the formulation improves the performance of the planning algorithm. This is followed by conclusions in Section 6.

2. Background

Consider a UAV modeled as a nonholonomic point mass moving in a two-dimensional plane at a constant velocity and no wind [9].

$$\begin{aligned}\dot{x} &= v \cos(\psi); \\ \dot{y} &= v \sin(\psi); \\ \dot{\psi} &= \omega.\end{aligned}\quad (1)$$

The position of the vehicle is defined by the coordinates $x(t)$ and $y(t)$ in a two-dimensional space. $\psi(t)$ corresponds to the heading angle of the vehicle that must be determined to maximize the coverage of a given bounded region. The vehicle is assumed to have a constant velocity v . The state of the vehicle is defined by $z = \{x, y, \psi\} \in X$. The vehicle will search the bounded region defined by $\Omega \in \mathfrak{R}^2$ and the vehicle has a sensor footprint defined by $Y : X \rightarrow 2^{\mathfrak{R}^2}$. The vehicle starts the mission at $p^{\text{entry}}(x, y)$ and is expected to terminate the mission at $p^{\text{exit}}(x, y)$. E_{total} is the amount of energy available at the start of the mission. The exact solution $z^*(t) = \{x^*(t), y^*(t), \psi^*(t)\}$ of the problem satisfies

$$\{x^*(t), y^*(t), \psi^*(t)\} = \arg \max_{z(t)} \bigcup_t (Y(x(t)) \cap \Omega). \quad (2)$$

The trajectory $z^*(t)$ maximizes the sensor footprint coverage of the search region. Figure 1 demonstrates a visual representation of the problem of generating a trajectory that maximizes the area covered and satisfies the exit stated for energy available. Figure 2 demonstrates the generated trajectory along with the sensor footprint area throughout the mission.

The procedure proposed in this paper to determine a path that maximizes coverage area depends on optimization across a range of turning rates based on the maximum load factor of the vehicle. In addition, the algorithm considers realistic

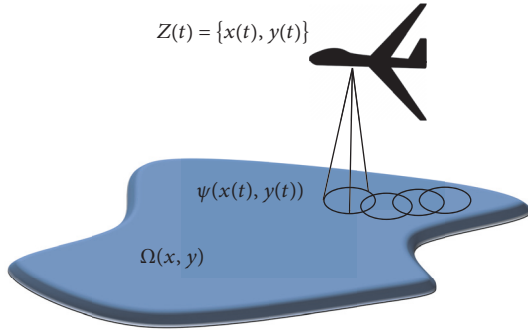


FIGURE 2: Area covered by sensor footprint for generated trajectory throughout the mission.

power consumption, which is not constant and varies with the type of maneuver (parameterized by turn rate) executed by the vehicle. This enables the accurate determination of the mission duration and enforcing of the terminal position requirements of the vehicle for the energy available at the start of the mission. The high level of autonomy ensures that the algorithm also generates the most optimal trajectory for any conditions by avoiding previously covered area as much as possible. The formulation avoids the vehicle from going over previously covered area by redirecting it to portions of the bounded region that have not been previously covered. This feature of the proposed formulation is that it does not require choosing the density of the space discretization/decomposition. The density of the space discretization depends on the size of the discretized space and the distance between the center points of the discretized spaces. Without the need to evaluate the parameters for optimal space discretization, the computation process is faster while still providing an accurate calculation of the covered area.

3. Path Planning Based on Range of Turning Rates and Available Energy

The problem of interest is that of an unmanned aerial vehicle operating at a fixed altitude in a closed, bounded region. The goal is to find a feasible trajectory, defined by $z(t)$, to get maximum coverage of the bounded region in every time interval τ . This time interval, τ , is the turn duration or more accurately the length of the time for which a single maneuver is planned to be executed. The overall mission length, T_f , is equivalent to a multiple of τ .

3.1. Turning Rate Range. The proposed formulation determines the maximum coverage trajectory for a range of turning rates. Equation (3) provides the maximum turn rate possible based on the maximum load factor during the sustained turn for the vehicle [12, 13].

$$\omega_{\max} = \frac{g\sqrt{n_{\max}^2 - 1}}{v} \quad (3)$$

The maximum load factor, defined by n_{\max} , which is dependent on the vehicle design, therefore determines the vehicle's

turn capability. The range of usable turning rates is then defined as $\{-\omega_{\max}, \omega_{\max}\}$.

3.2. Optimization Formulation. The overall path planning formulation is discretized into multiple optimizations performed over successive time steps, τ . This optimization procedure minimizes the sum of the amount of the area of the bounded region not covered by the UAV and the terminal cost function over the time period τ . The optimization is subject to the equation of motions of the vehicle moving on a two-dimensional plane, size of the sensor footprint, range of allowable turn rates, obstacles (already covered area), and boundary conditions. Ω is the area of the bounded region and the variable \mathbf{Y} represents the sensor footprint. The vector $z(t)$ provides the location and orientation of the vehicle in the x - y -plane at time $t \in [0, T_f]$, where T_f corresponds to the time at the end of the mission. The determination of the path over the total mission time is performed in N discrete steps with each step assumed to be an optimization over τ seconds. This implies that the total time of the mission is $T_f = N\tau$. Let each step of the planned path be parameterized by the variable k such that $k = 1, 2, 3, \dots, N$. Each discrete step k over which the optimization is performed is further discretized into n steps of length Δt such that $n\Delta t = \tau$. The variables $z(k) = [x(k), y(k), \psi(k)]$ correspond to the position of the vehicle at the end of each optimization step (i.e., once every τ seconds). The variables $z_{nk} = [x_{nk}, y_{nk}, \psi_{nk}]$, $n = 1, 2, \dots, \tau/\Delta t$, correspond to the intermediate position of the vehicle within each discrete optimization step, k . The area of the bounded region that remains uncovered by the vehicle at the end of the each discrete optimization step k is defined by $\Omega(k)$. The uncovered area depends on the uncovered area of the previous time step minus the new area covered by the sensor footprint at the end of the time step. In order to ensure that the vehicle covers the most area of the bounded region without revisiting already covered portions, the formulation incorporates a no-fly zone/obstacle from the area previously covered, provided by the union of the regions $\gamma(z(k)) \cap \Omega(k-1) > 0$ where $\gamma(z(k))$ is a vector describing the sensor footprint contour and $\Omega(k-1)$ is the contour of the bounded search region that has been covered. When the vehicle starts the search and coverage of a region, the no-fly/obstacle avoidance constraints dominate since the algorithm looks for a path with the most uncovered area. However, the priority of the no-fly/obstacle avoidance constraint reduces as the mission progresses and the amount of uncovered area reduces. As the mission progresses, the region that vehicle must avoid increases since the area of bounded regions previously covered by the vehicle increases. The decrease in the priority of the no-fly/obstacle avoidance constraint avoids the vehicle getting stuck and allows for a robust solution. As the vehicle begins to surround itself with previously covered area, the algorithm starts redirecting the vehicle to uncovered portions of the bounded region. Based on these assumptions, the optimization at each step k can be written as

$$\min_{\omega} J(k) = \Omega(k-1) - \bigcup_{k=1}^N \mathbf{Y}(z(k)) + C \quad (4)$$

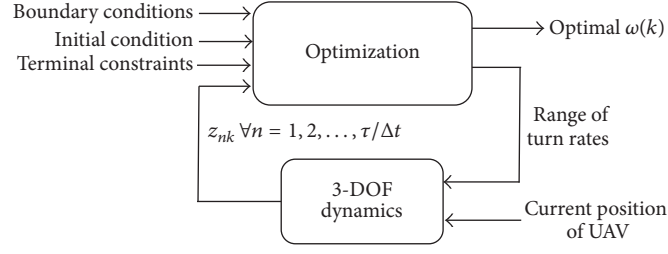


FIGURE 3: Model-based optimization.

$$\text{subject to } z(k) = [x(k), y(k)] \quad (5a) \quad \text{where}$$

$$x(k) = x(k-1) + x_{\tau k} \quad (5b)$$

$$y(k) = y(k-1) + y_{\tau k} \quad (5c)$$

$$\gamma(z(k)) \cap \Omega(k-1) > 0 \quad (5d)$$

$$\omega \leq |\omega_{\max}|, \quad (5e)$$

where

$$x_{\tau k} = x_{nk} \quad \text{at } n = \frac{\tau}{\Delta t} \quad (5f)$$

$$y_{\tau k} = y_{nk} \quad \text{at } n = \frac{\tau}{\Delta t} \quad (5g)$$

$$x_{nk} = v\Delta t \cos(\omega\Delta t + \psi_{(n-1)k}) + x_{(n-1)k} \quad (5h)$$

$$y_{nk} = v\Delta t \sin(\omega\Delta t + \psi_{(n-1)k}) + y_{(n-1)k}. \quad (5i)$$

One of the boundary conditions is the physical boundary of the search region. The vehicle should stay within the boundaries and if the vehicle does move out of the boundary region then it should return inside of the boundary region as soon as possible. The second boundary condition is the maximum stored energy available (E_{total}) at the start of the mission. The rate of energy consumption of the vehicle during the mission depends on the velocity and the control input: turn rate. The turn rate of the maneuver affects the amount of power required by the vehicle. The larger the turn rate, the more the power required by the vehicle to execute the maneuver since the power required is proportional to the square of turn rate (6d).

3.2.1. Boundary Conditions

$$x_{\min} \leq x \leq x_{\max} \quad (6a)$$

$$y_{\min} \leq y \leq y_{\max} \quad (6b)$$

$$\sum_{k=1}^N f_{\text{energy}}(v, \omega, t) < E_{\text{total}}, \quad (6c)$$

$$f_{\text{energy}}(v, \omega, t) = \text{Thrust} * v\tau = \left(\bar{q}SC_{D_0} + \frac{((\omega v/g)^2 + 1) \text{Weight}^2}{\bar{q}S\pi Ae} \right) v k \tau. \quad (6d)$$

The block diagram of the optimization procedure is shown in Figure 3. The 3-DOF model of the UAV simulates the motion of the vehicle for a planning horizon for k th step. Its input includes the boundary conditions such as exit and entry point, velocity of the vehicle, entry heading angle, and available energy. The output of the optimization routine is the optimal turning rate ω_k for the k th step in the optimization. Note that ω for each discrete step is based on a planning horizon of τ seconds. Once the optimum turning rate for the k th step is determined, the UAV executes the turning maneuver for an execution horizon of τ seconds. The position of the UAV at the end of τ seconds is fed back to the optimization routine to determine the optimal turning rate for the next discrete step ($k+1$) in the mission. This continues until any of the boundary conditions are met. As previously stated, the model-based optimizer plans the path for a particular planning horizon that covers the most area of the search region not previously covered. The optimization routine calculates the ideal turning rate of the trajectory for the planning horizon, since it does not account for disturbances that can disturb the vehicle from the calculated path.

This formulation is expected to determine the turn rate from the allowable range of turning rates that minimizes the cost function. The performance index J in (4) is reformulated as (7); S is called the priority function, A is the primary cost function, and C is the terminal function.

$$J(k) = A(z(k), \Omega(k-1))S + C. \quad (7)$$

3.2.2. Cost Function, $A(z, \Omega)$. The cost function primarily determines the percentage of area covered in each time interval by the possible paths. If the sensor covers area that was not previously covered then the cost function is equal to the inverse of the new area covered. The inverse of the new area covered is obtained since the optimization is minimizing the performance index, so the more area covered lowers the value of the index. However, if the sensor covers area that was previously covered without covering any new area then the cost function is equal to the distance from the current state

to the centroid of uncovered area. The second condition in the cost function ensures that the algorithm always searches for area not previously covered, in order to maximize the percentage of area covered at the end of the mission. The cost function can thus be defined as

$$A(z(k), \Omega((k-1))) = \begin{cases} \frac{1}{\Upsilon(z(k)) \cap \Omega(k-1)} & \text{if } \Upsilon(z(k)) \cap \Omega(k-1) > 0 \\ \frac{1}{\sqrt{\bar{x}^2 + \bar{y}^2}} & \text{if } \Upsilon(z(k)) \cap \Omega(k-1) = 0, \end{cases} \quad (8)$$

where (\bar{x}, \bar{y}) represents the coordinates of the centroid of the uncovered area. This formulation thus introduces a decision making capability in the path planning: if no new area is covered in step k , the optimization objective is the next step; $(k+1)$ will be to minimize the distance of the vehicle from the current location to the centroid of the remaining uncovered area.

3.2.3. Priority Function, S. The priority function determines the immediate objective of the vehicle during the mission. It controls the priority of the vehicle to continue searching or redirect to the desired exit point. The function can be based on the time constraint as in [9] or can be based on the energy constraint as in this study.

(a) *Based on Time Constraint, S(T).* Equation (8) provides the priority function for the time constraint. If the remaining mission duration at the end of each interval is greater than t_{safe} then it prioritizes the vehicle to continue to attempt to cover the most area possible. However, if the remaining mission duration is less than t_{safe} then the priority function instructs the vehicle to try to reach the desired exit state as soon as possible. The variable t_{safe} is the amount of time required by the vehicle to reach the desired state, using a direct path.

$$S(T) = \begin{cases} 1 & \text{if } T_{\text{remaining}} > t_{\text{safe}} \\ 0 & \text{if } T_{\text{remaining}} \leq t_{\text{safe}} \end{cases} \quad (9)$$

(b) *Based on Energy Constraint with Nonconstant Power Consumption, S(E).* The priority function for energy constraint is given by (9). If the energy remaining in the vehicle at the end of the interval is greater than the energy required to reach the desired exit state, using a direct path assuming the vehicle is initial heading to the vehicle; then the priority function directs the vehicle to continue searching as much area as possible. However, if the amount of energy remaining at the end of the interval is less than the energy required to reach the desired exit, using a direct path, then the priority function directs the vehicle to travel to the exit immediately. Moreover, the priority function takes into consideration that the power consumption of the vehicle is not constant. The remaining energy calculated in the priority function accounts

for the varying energy consumption of the vehicle due to the maneuver of the vehicle.

$$S(E) = \begin{cases} 1 & \text{if } E_{\text{remaining}} > P_{\text{required}} t_{\text{exit}} \\ 0 & \text{if } 0 \leq E_{\text{remaining}} \leq P_{\text{required}} t_{\text{exit}} \end{cases} \quad (10)$$

3.2.4. Terminal Function, C. The performance index of a receding horizon optimization problem estimates the cost-to-go from a selected terminal state to the final goal. Again, this terminal function can be based on time constraint [9] or the energy constraint.

(a) *Based on Time Constraint, C(T).* Equation (11) provides the terminal function for the time constraint. If the remaining time at the end of the interval is greater than the time required to directly head to the desired exit state then the cost is zero, which means that the vehicle's objective is to cover as much area as possible. If the remaining mission duration at the end of the interval is less than the time required to head directly to the desired exit state, then the cost function is equal to the inverse of the remaining mission duration. However, if the remaining mission time is less than zero then the cost function is equal to infinity.

$$C(T) = \begin{cases} 0 & \text{if } T_{\text{remaining}} > t_{\text{safe}} \\ \frac{1}{T_{\text{remaining}} + \varepsilon} & \text{if } 0 \leq T_{\text{remaining}} \leq t_{\text{safe}} \\ \infty & \text{if } T_{\text{remaining}} < 0. \end{cases} \quad (11)$$

(b) *Based on Energy Constraint with Nonconstant Power Consumption, C(E).* Equation (12) provides the terminal function for the energy constraint. If the remaining energy at the end of the interval is greater than the energy required to directly head to the desired exit state, then the cost is zero. A cost of zero means that the vehicle's objective is to cover as much area as possible. If the remaining mission duration at the end of the interval is less than the energy required to head directly to the desired exit state, then the cost function is equal to the inverse of the remaining energy required. However, if the remaining energy is less than zero then the cost function is equal to infinity. Moreover, the terminal function with the energy constraint assumes that the energy consumption is not constant since the amount of energy consumption varies due the vehicle's maneuver. The calculated energy remaining accounts for the energy required by the vehicle for the turning rate used since energy consumption is not constant for different turning rates.

$$C(E) = \begin{cases} 0 & \text{if } E_{\text{available}} > P_{\text{required}} t_{\text{exit}} \\ \frac{1}{E_{\text{available}}} & \text{if } 0 \leq E_{\text{available}} \leq P_{\text{required}} t_{\text{exit}} \\ \infty & \text{if } E_{\text{available}} < 0. \end{cases} \quad (12)$$

Power Required, P_{required} . Optimizing the trajectory that satisfies the energy constraint requires calculating the power requirements of the vehicle during a maneuver and the power required to reach the desired exit states, using a direct path

[12, 13]. Equation (13) calculates the power required by the vehicle to reach the exit state using a direct path. Equation (14) calculates the powers required by the vehicle for particular maneuver. The power required by a maneuver is a function of the load factor of the maneuver. Equation (15) calculates the load factor of the maneuver, which in turn is a function of the turn rate, $\dot{\psi}$.

$$P_{\text{required}} = vD = v \left(\frac{1}{2} \rho v^2 S C_{D_0} + \frac{2(mg)^2}{\rho v^2 S \pi A R e} \right) \quad (13)$$

$$= \frac{1}{2} \rho S v^3 (C_{D_0} + 3C_{D_0})$$

$$P_{\text{traj}} = vD = v \left(\frac{1}{2} \rho v^2 S C_{D_0} + \frac{2(mg)^2 n^2}{\rho v^2 S \pi A R e} \right) \quad (14)$$

$$= \frac{1}{2} \rho S v^3 (C_{D_0} + 3n^2 C_{D_0})$$

$$n = \sqrt{\left(\frac{\dot{\psi} v}{g} \right)^2 + 1}. \quad (15)$$

4. 6-DOF UAV Simulation Platform

The performance of the path planning algorithm is evaluated through its implementation on a high-fidelity 6-DOF nonlinear simulation of a UAV in the Matlab/Simulink environment. The UAV is assumed to be a 1/10th scale model of the Navion general aviation aircraft, allowing for the determination of its geometry and mass properties. The UAV has an elevator, ailerons, and rudder to control its motion. Additionally, it has a battery powered motor that produces a constant thrust throughout the mission. The aerodynamic coefficients that are required to determine the forces and moments acting on the UAV to complete the 6-DOF simulation model were obtained using the USAF Datcom software [14] for a given operating condition and imported into Matlab. The 6-DOF equations of motion of the aircraft are based on the typical flat earth approximation and are shown in Box 1.

The path planning procedure assumes that the aircraft is flying at a constant velocity and fixed altitude. Additionally, it assumes that the turn rate selected by the optimization can be achieved by the aircraft. This requires the UAV simulation to have a velocity hold, altitude hold, and a directional (heading hold) autopilot. The Simulink model of the Navion general aviation aircraft includes the 6-DOF block and the three aforementioned autopilots. The input for the Simulink model of the Navion general aviation aircraft is the optimized turning rate, which is selected during the path planning phase based on a 3-DOF model of the vehicle. From the turning rate and the turn duration the heading of the vehicle at the end of the turn duration is determined, which is then provided to the heading autopilot.

The block diagram of UAV simulation model is presented in Figure 4. The autopilots are designed so that the motion of the vehicle is as close as possible to the Dubin car model (3-DOF Dynamics, motion in a horizontal plane), which

$$\begin{aligned} \dot{p} &= c_1 r q + c_2 p q + c_3 L + c_4 N \\ \dot{q} &= c_5 p r + c_6 (p^2 - r^2) + c_7 M \\ \dot{r} &= c_8 p q + c_2 r q + c_4 L + c_9 N \\ \dot{\phi} &= p + q \sin \phi \tan \theta + r \cos \phi \tan \theta \\ \dot{\theta} &= q \cos \phi - r \sin \phi \\ \dot{\psi} &= q \sin \phi \sec \theta + r \cos \phi \sec \theta \\ \dot{U} &= rV - qW + \frac{F_x}{m} - g \sin \theta \\ \dot{V} &= -rU + pW + \frac{F_y}{m} + g \cos \theta \sin \phi \\ \dot{W} &= qU - pV + \frac{F_z}{m} + g \cos \theta \cos \phi \\ c_1 &= \frac{[(J_y - J_z)J_z - J_{xz}^2]}{\Gamma} \\ c_2 &= \frac{[(J_x - J_y + J_z)J_z]}{\Gamma} \\ c_3 &= \frac{J_z}{\Gamma} \\ c_4 &= \frac{J_{xz}}{\Gamma} \\ c_5 &= \frac{(J_z - J_x)}{J_y} \\ c_6 &= \frac{J_{xz}}{J_y} \\ c_7 &= \frac{1}{J_y} \\ c_8 &= \frac{[(J_x - J_y)J_x - J_{xz}^2]}{\Gamma} \\ c_9 &= \frac{J_x}{\Gamma} \\ \Gamma &= J_x J_z - J_{xz}^2 \\ J_x, J_y, J_z &\text{ are Moments of Inertia} \\ J_{xz} &\text{ is Product of Inertia} \end{aligned}$$

Box 1: 6-DOF Dynamics of UAV. F_x, F_y, F_z, L, M, N are the aerodynamic forces and moments acting on the aircraft.

requires the vehicle to move in a plane at constant velocity. The altitude hold autopilot maintains the altitude of the vehicle as it performs a maneuver. The directional hold autopilot allows the UAV to perform a maneuver based on the desired turn rate.

The following subsections discuss the autopilots that have been developed to control the motion of the aircraft.

4.1. Altitude Hold Autopilot. The altitude hold autopilot (Figure 5) has a vertical velocity, pitch, and pitch rate feedback. The control law for the altitude hold autopilot is a Proportional Integral Derivative (PID) controller operating on the error in altitude. The inner loops that provide feedback on the vertical velocity, pitch, and pitch rate have proportional gains that are determined based on the selected turn duration. The varying gains for the feedback of the vertical velocity and pitch are required because the performance of a fixed autopilot degrades for varying turn durations.

4.2. Velocity Hold Autopilot. The velocity hold autopilot is presented in Figure 6. The velocity hold autopilot has a simple

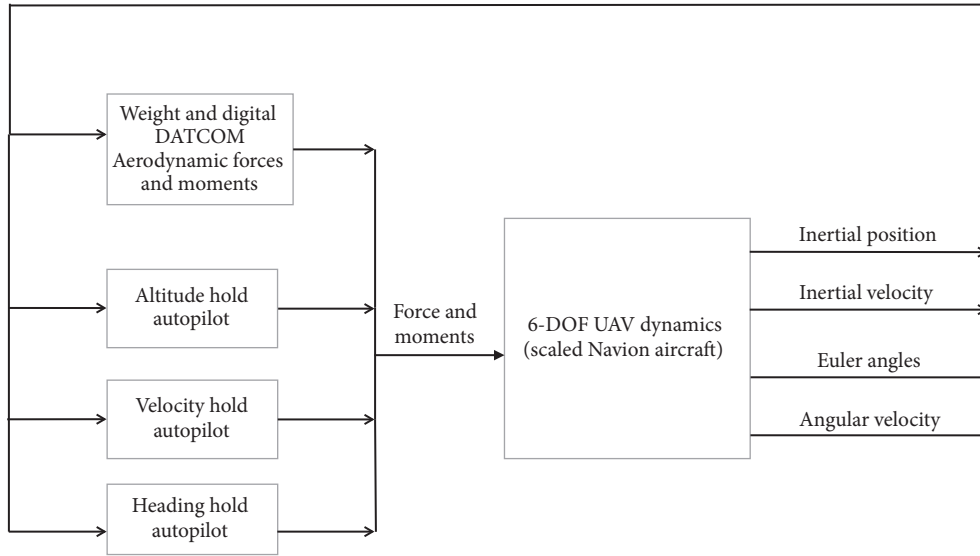


FIGURE 4: Block diagram of 6-DOF UAV simulation model.

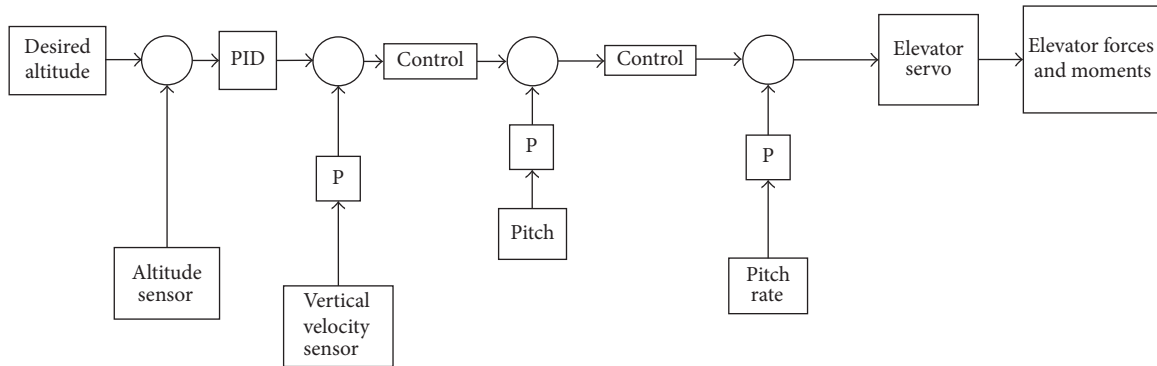


FIGURE 5: Altitude hold autopilot.

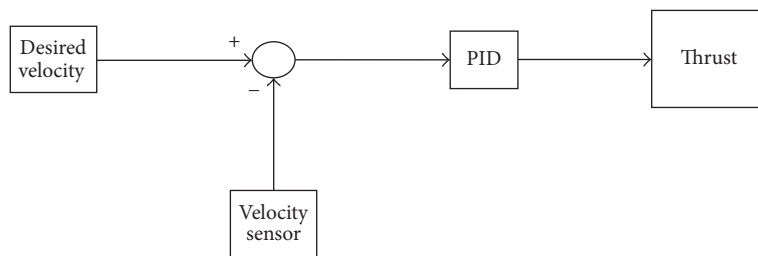


FIGURE 6: Velocity hold autopilot.

PID controller. The velocity hold autopilot has fixed gains since the turn duration does not affect the performance of the autopilot.

4.3. *Heading Hold/Directional Autopilot.* Figure 7 provides the block diagram of directional autopilot used to maneuver the vehicle based on the desired turn rate. The autopilot determines the aileron and rudder deflection required to perform the desired maneuver.

From the desired turning rate ($\dot{\psi} = \omega$) and length of turn, the required yaw angle (ψ) at the end of the turn is

determined. This yaw angle is used instead of the turning rate in the design of the autopilot because the turning rate is too noisy making it difficult to design a heading hold autopilot. In addition, from the yaw, one can determine the roll angle (ϕ) required to perform the desired maneuver. The autopilot has a PID control law that operates on the roll angle error.

4.4. *Summary of Path Planning Implementation.* The algorithm is implemented in four main steps as summarized in Figure 8. The first step is 3-DOF path planning for the possible paths in the turning rate range. After the 3-DOF

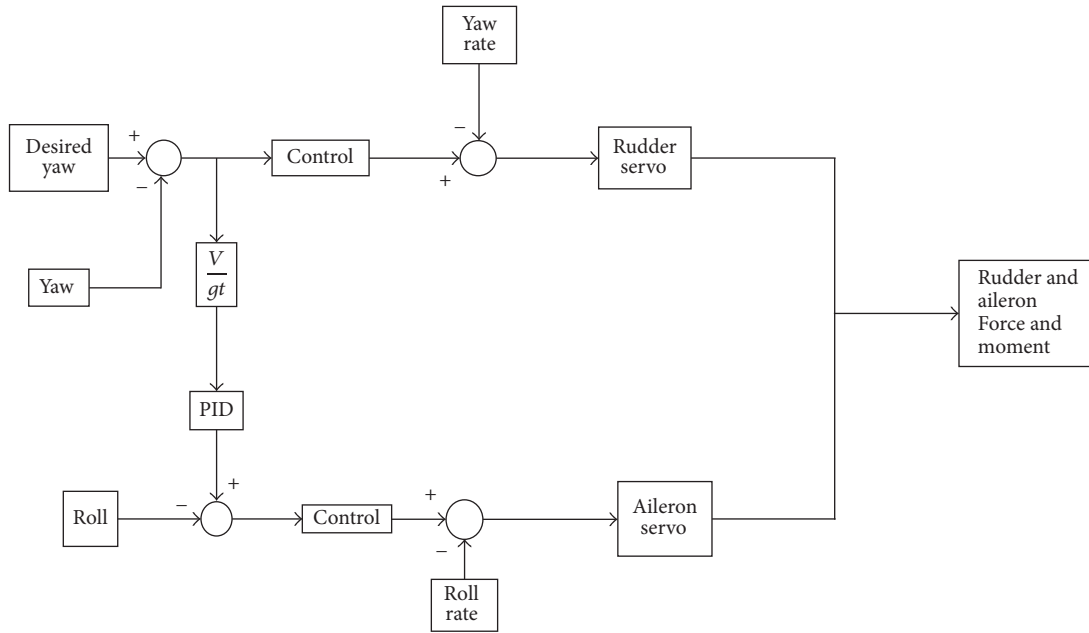


FIGURE 7: Heading hold/direction autopilot.

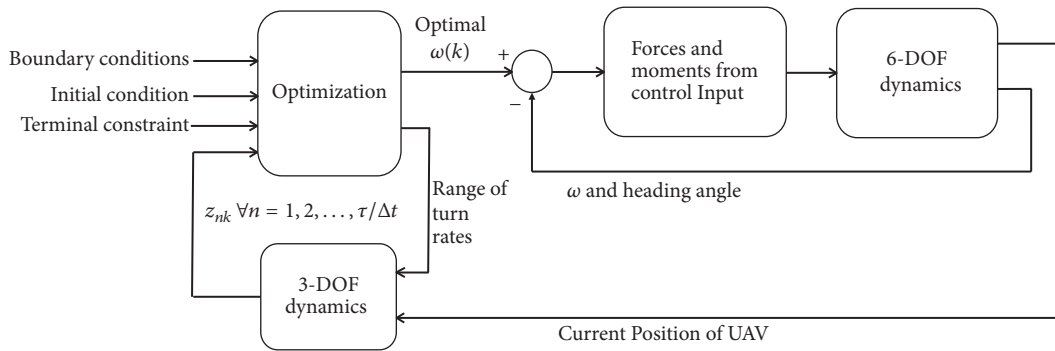


FIGURE 8: Summary of path planning implementation.

path planning, the algorithm optimizes the path by selecting the turning rate that generates the path with the maximum coverage. The next step is the implementation of the selected tuning rate in a heading and turning rate feedback control. From the error of the of the heading and turning rate, the controller determines the required control input to achieve the desired response. The final step of the implementation is to provide feedback on the current position and heading of the vehicle to repeat the process for a refresh rate set. The refresh rate in the implementation is the selected turn duration.

5. Path Planning Simulation Results

The simulation performed considers a single UAV navigating a specified region. The region is defined from the maximum area that the vehicle can observe, based on the vehicle specifications, assuming ideal conditions. The simulation plans the path using a three-degree-of-freedom model and

applies the control input, the turning rate obtained in the three-degree-of-freedom model, to a six-degree-of-freedom model of the small UAV.

The maximum area that the vehicle can cover assuming straight and steady level flight, is 598,100 m² that will provide a square region of approximately 773.36 m by 773.36 m. The vehicle enters the region at $x = 10$ m, $y = 10$ m and a heading angle of 45°. It is expected to terminate its mission at $x = 10$ m and $y = 783.36$ m. Table 1 provides the power specifications sensor footprint of the vehicle in the simulation. The camera is assumed to be always pointing straight to the ground during the maneuvers. That is, it is gimbaled such that the focal plane of the lens is parallel to the ground. In addition, the simulation considers that the vehicle is operating at altitude of 121.92 meters, which is the federal operating limit for model airplanes, on a standard day.

5.1. Optimization Routine. The optimization routine used in the simulation is *fminbnd*, which is a one-dimensional

TABLE I: Properties of the vehicle for simulation.

Property	Value
<i>Vehicle specification</i>	
Oswald efficiency	1
Motor efficiency	0.9
Max load factor (n)	1.5
<i>Battery specification</i>	
Electric charge (mAh)	2200
Voltage (V)	11.1
<i>Sensor specification</i>	
Camera radius (m)	50

minimizer in Matlab. It should be noted that the choice of the optimization routine is independent of the formulation of the path planning algorithm. Other methods such as *fminsearch* were explored with very similar results. However, the choice of the routine influences the time taken to compute the optimal turn rate which might become significant for real-time implementation.

5.2. Area Calculation. The area covered by the vehicle as it navigates the planned path is determined as follows. At a given simulation time step, the contour of the sensor foot print (circle of radius 50 m centered at the position of the vehicle at that time) is approximated using a finite number of vertices. As the vehicle navigates the path, the sensor foot prints are continuously added together to determine the covered area while the union of vertices of the contours is recorded to update its overall boundary. It is ensured that any area outside the mission boundary as well as the area previously covered is excluded from this computation. The computation of these contours involve the union of sets of vertices (for combining covered regions) as well as its intersection (for excluding regions). Such operations are implemented by using available routines from the Mapping Toolbox in Matlab such as *polybool* and *polyarea*. It should be noted that these routines are utilized for simplification of the implementation in the simulation framework. For real-time implementation, more robust and efficient algorithms can be used.

Figure 9 is an example of area calculation for a given mission. As noted earlier, if the algorithm identifies that the vehicle is covering new area, then the output of the function is simply the inverse of the amount of new area covered. However, if no new area is covered then the output of the function is the distance to the centroid of uncovered portions of the bounded search region.

Figure 10 shows an example of a mission executed by the UAV using the proposed path planning formulation. Figure 10(a) shows actual vehicle path (obtained from 6-DOF simulation) along with the planned path (obtained from the 3-DOF model). It can be seen that the vehicle does not track the planned path accurately but executes a path that is very similar. The figure is included to illustrate the performance of the actual path of the vehicle with the designed autopilot and the desired path. Figure 10(b) shows the area covered

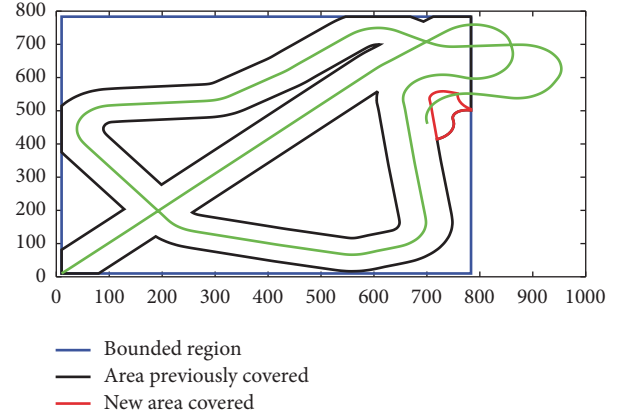


FIGURE 9: Illustrative example of the area calculation. The contours are defined by a matrix of the vertices. Matlab routines are utilized to compute the area covered based on these vertices.

by the vehicle during the mission. The areas with the solid red circles are the uncovered areas. Figure 11 provides the percent tracking error (distance between desired position versus actual position as a percent of total distance traveled) for the entire mission. From Figure 11, it is evident that the designed autopilot at its worst performance has an error of 0.35% from the planned path. The reason for the error is that the planned path is for a 3-DOF and the actual path is obtained in Simulink for a 6-DOF vehicle.

5.3. Comparative Results. This section discusses the performance of proposed path planning algorithm based on the following factors:

- Percentage of total area covered
- Distance of the final position of the UAV from the desired exit point

The percent of total area covered is indicative of accomplishing the primary mission while the distance from exit represents the ease of access to the final position of the UAV once the stored energy is spent. The simulation is performed for different values of the duration of the turn as this is an important parameter that affects the total area covered.

The results provided explore the improvement in the performance parameters for each of the features of the formulation. These include the utilization of a range of turn rates based on maneuver capability of the vehicle, a novel formulation for computing the area covered, and an explicit constraint based on the total energy available for the mission.

5.3.1. Range of Turn Rates versus Discrete Set of Turn Rates. Figure 12 provides the performance of the algorithm for a discrete set of turning rates versus a range of turn rates across different turn durations. Figure 12(a) provides the percent of the search region that the UAV will cover. Figure 12(b) provides the distance of the vehicle from the desired exit state at the end of the mission. The discrete set of turning rates are $[-0.3142, -0.1571, 0, 0.1571, 0.3142]$ rad/s while the range of turn rates has a boundary of ± 0.3142 rad/s. As can be seen

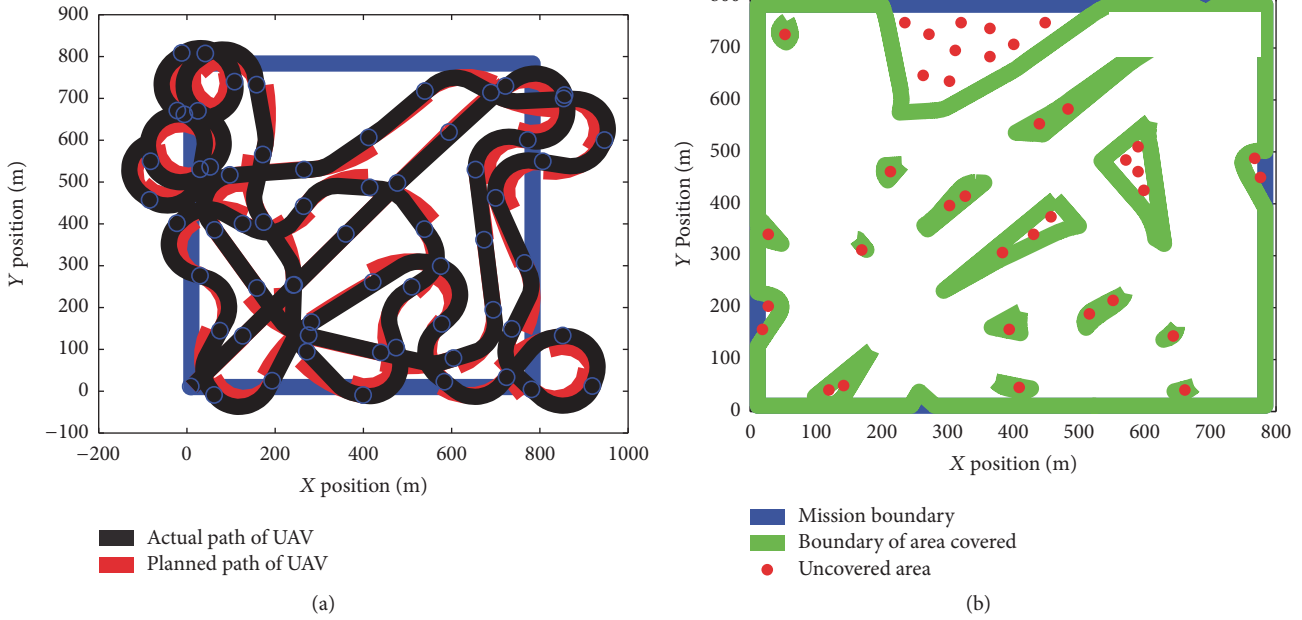


FIGURE 10: (a) Comparison of planned path and actual path. The actual path presented is the path from the 6-DOF simulation generated from desired turning rate. The planned path is a path generated using the 3-DOF Dubins model. (b) Area covered by the vehicle.

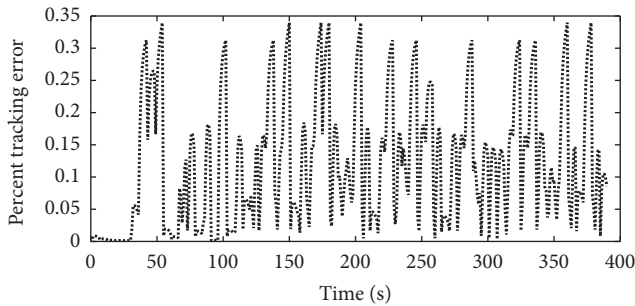


FIGURE 11: Percent tracking error for a mission with a 6-second turn duration.

from the figure, the selected discrete set of turning rates may not provide the best coverage and end the mission close to the desired exit state. While the best set of discrete turn rates may theoretically exist for a given mission definition, identifying this set will be computationally exhaustive. Further, this “best” set of discrete turning rates will have to be identified for every possible mission configuration and hence is not suitable for real-time implementation.

5.3.2. Impact of Cost Function with Decision Capability. The primary cost function proposed in this paper allows the capability of the path planning algorithm to make an explicit decision to visit an uncovered area. Figure 13 provides the comparative performance of the path planning formulation with and without the utilization of the novel cost function. Note that both formulations utilize the range of turn rates in the optimization. Figure 13(a) provides the percent of area covered by the path planning algorithm and demonstrates that the decision capability that was added to the cost

function improves the coverage performance since it redirects the vehicle to area not previously covered. In other words, the algorithm directs the vehicle to portions of the search region not covered when the vehicle does not cover new area in every time step. Moreover, Figure 13(b) provides the distance of the vehicle from the desired exit state at the end of the mission. It is evident that the algorithm with the decision capability improves the performance as compared to that without decision capability.

5.3.3. Energy Constraint and Power Consumption Calculation Based on Maneuver. The final feature of the proposed path planning formulation is the explicit utilization of the available stored energy as a constraint and calculation of power consumed based on the maneuver. Figure 14 shows the performance of the path planning with this feature as compared to one with time as a constraint. It should be noted that both formulations utilize a range of turn rates and the cost function with the decision making capability. Figure 14(a) shows the coverage of the search region for varying turn durations for both energy and time constraints, with all the modifications previously discussed. Figure 14(b) provides the distance of the vehicle from the desired exit state at the end of the mission for both energy and time constraints. From Figure 14(a), it is evident that the coverage of the bounded search regions is very similar for both the energy and the time constraint; no significant variation is evident. While it appears that there is no significant improvement to the area covered with an energy constraint since the results are very similar for the coverage of the bounded region, the strength of the formulation is evident from Figure 14(b) which indicates the distance of the vehicle from the desired exit state at the end of the mission. It should be mentioned

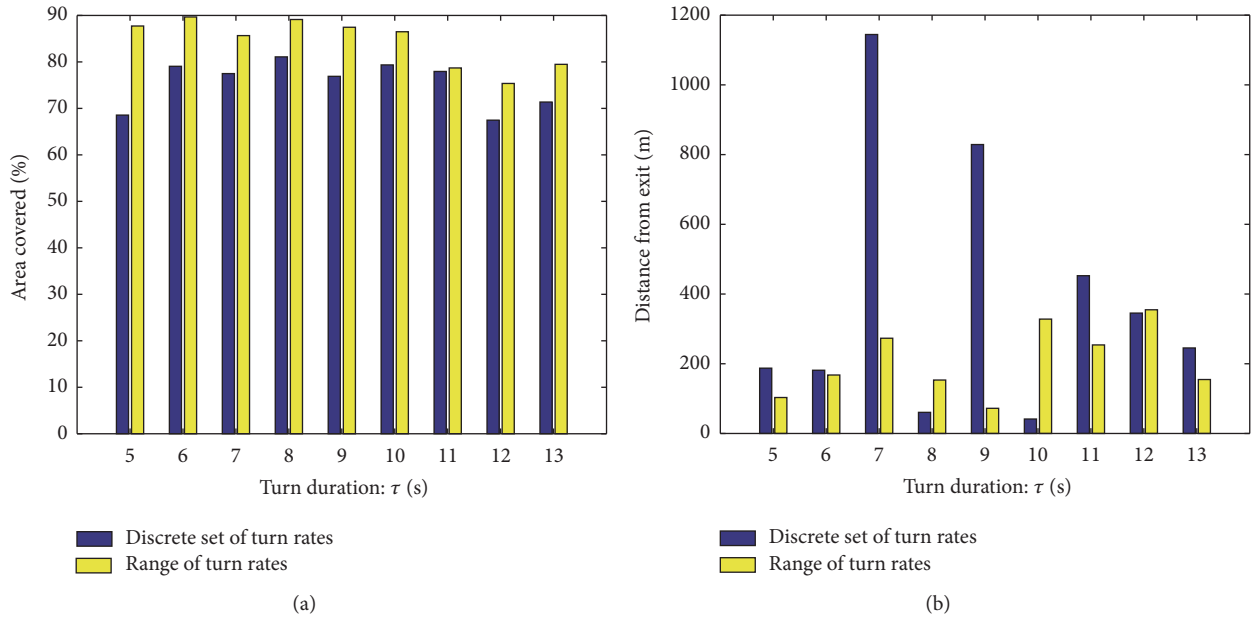


FIGURE 12: Performance of discrete set and range of turn rates: (a) percent of area covered and (b) distance of the vehicle from the desired exit state at the end of the mission.

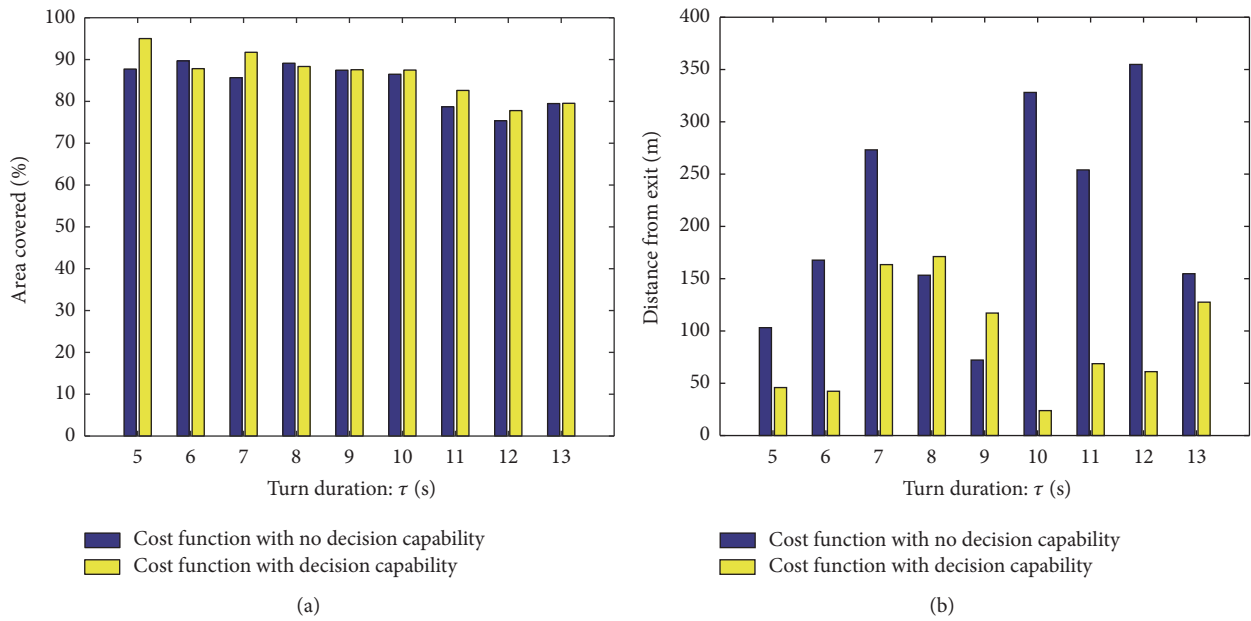


FIGURE 13: Performance of algorithm for no decision capability and decision capability for range of turn rates: (a) coverage performance and (b) distance from desired exit state at the end of mission.

here that comparing the distance from Figures 13(b) and 14(b) for the time constraint shows a significant difference between the results. This is because the results in Figure 13(b) assume that the vehicle will be active for duration of 420 seconds with constant power consumption. However, if the requirement that the power consumed is dependent on the maneuver (turn rate) is included, then the total mission duration is reduced as indicated in Figure 15. Figure 14(b) provides the comparison distance of the vehicle from the desired exit state between the formulations utilizing energy and time constraint assuming

that power consumed is dependent on turn rate. From these results, it is evident that, in most cases, the vehicle path planned using the energy constraint allows it to reach closer to the desired exit state than the path planning with the time constraint.

5.4. Discussion. The results presented in the previous section show the advantages of implementing the proposed features in the path planning of a UAV. All three features, including the maneuverability constraint based on design load factor

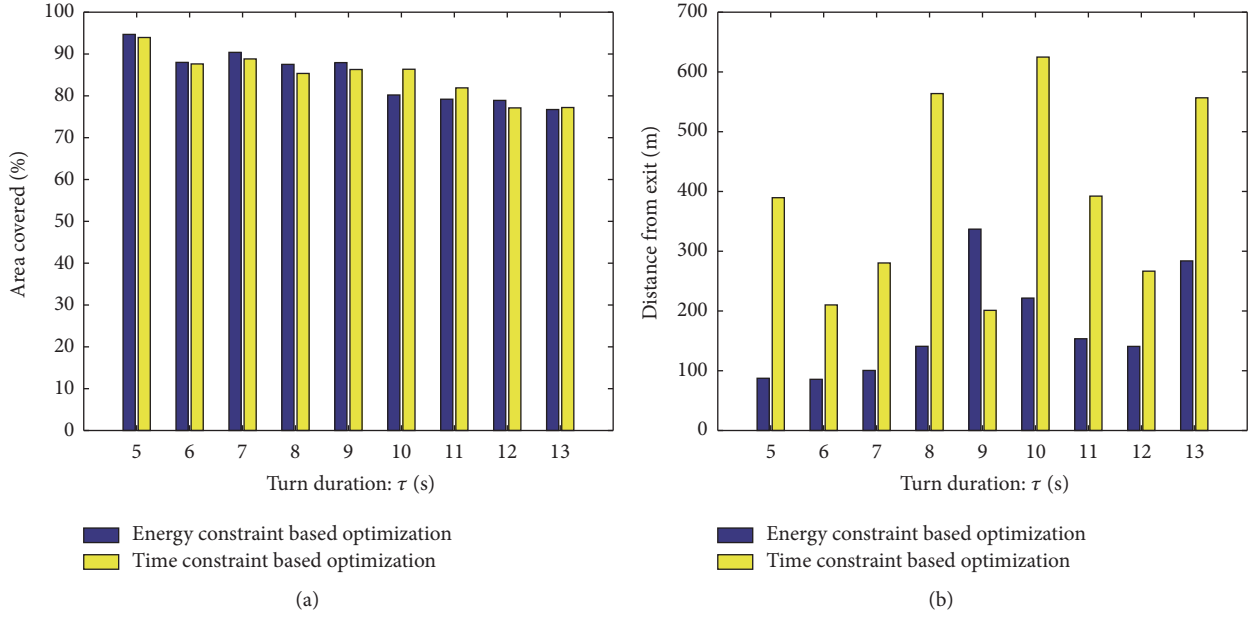


FIGURE 14: Algorithm performance with decision capability and energy optimization for a range of turn rates: (a) coverage of search region and (b) distance from the desired exit state.

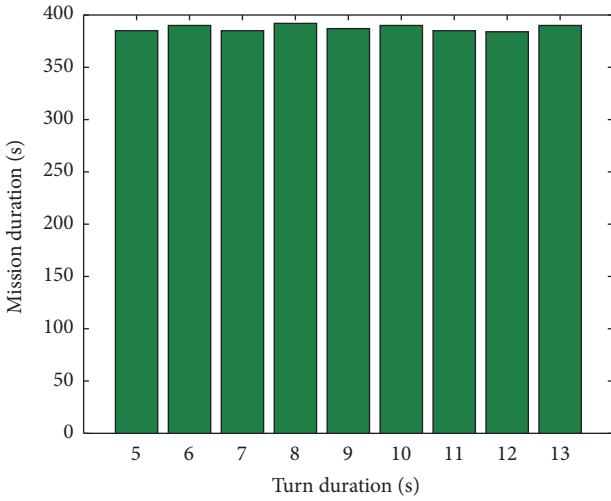


FIGURE 15: Actual mission duration for time constraint based formulation with power consumption dependent on the maneuver.

of the vehicle, novel area formulation, and explicit modeling of energy constraint, allows improvement in the primary performance metrics of the path planning. This formulation can be extended to multiple UAVs with inclusion of obstacle avoidance. Additionally, without having to determine parameters such as a discrete set of possible turn rates or having a priori knowledge of the actual time, the stored energy will last; the proposed formulation (with optimization of the programming to improve computational efficiency) can be implemented as a real-time guidance algorithm for the UAV. Such an implementation will allow flexibility in the missions that the UAV can execute while still maintaining a high percentage of area covered and reliable recovery position.

6. Conclusions

This paper presents an efficient path planning algorithm for a UAV that maximizes the area covered with stored energy. An important feature of the path planning algorithm includes optimization over a range of turn rates defined by the maneuver capability of the vehicle in a sustained turn as compared to heuristically selecting discrete set of turn rates. Simulation results show that the novel formulation of the optimization problem does not degrade the area covered as compared to the typical optimization using a time constraint. Evaluation of the overall mission duration and final position of the UAV for both the energy constraint and the time constraint based optimization indicates that the formulation using time constraint calculates it incorrectly. Further, direct comparison of the final position of the vehicle when comparing the two optimization formulations shows that the energy constraint allows the vehicle to be recovered from a location closer to the desired exit point.

Nomenclature

- A: Area function
- AR: Aspect ratio
- C: Cost function
- C_{D_0} : Zero-lift drag
- D: Drag
- e: Oswaldo efficiency factor
- E: Energy
- F_x : x -component of force
- F_y : y -component of force
- F_z : z -component of force
- g: Acceleration due to gravity
- k: Time step

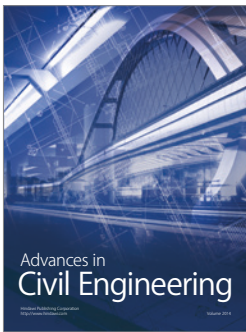
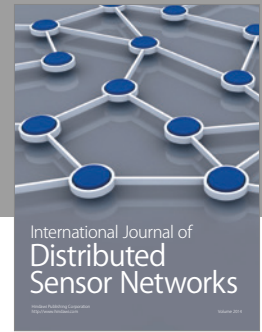
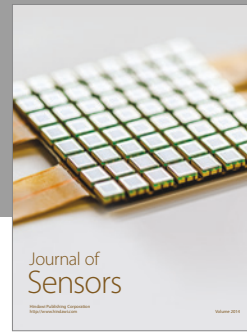
L:	x -component of aerodynamic moment
M:	y -component of aerodynamic moment
m:	Mass
N:	z -component of aerodynamic moment
n:	Load factor
P_{traj}:	Power for the path
S_{ref}:	Reference area
S:	Priority function
T:	Thrust
Ū:	x -component of acceleration
U:	x -component of velocity
Ŷ:	y -component of acceleration
V:	y -component of velocity
v:	Magnitude of Velocity
Ŵ:	z -component of acceleration
W:	z -component of velocity
Weight:	Weight of the vehicle
ȳ:	y -component of centroid of uncovered area
ȳ:	x -component of centroid of uncovered area
ρ:	Air density
ψ:	Turn rate
p:	x -component of angular rate
q:	y -component of angular rate
q̄:	Dynamic pressure
r:	z -component of angular rate
τ:	Turn duration
γ:	Sensor footprint
Ω:	Bounded search region.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

- [1] A. Ryan, M. Zennaro, A. Howell, R. Sengupta, and J. K. Hedrick, "An overview of emerging results in cooperative UAV control," in *Proceedings of the 43rd IEEE Conference on Decision and Control (CDC '04)*, vol. 1, pp. 602–607, IEEE, December 2004.
- [2] A. Ryan and J. K. Hedrick, "A mode-switching path planner for UAV-assisted search and rescue," in *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference (CDC-ECC '05)*, pp. 1471–1476, December 2005.
- [3] S. Waharte and N. Trigoni, "Supporting search and rescue operations with UAVs," in *Proceedings of the International Conference on Emerging Security Technologies (EST '10)*, pp. 142–147, September 2010.
- [4] J. S. Kim and B. K. Kim, "Minimum-time grid coverage trajectory planning algorithm for mobile robots with battery voltage constraints," in *Proceedings of the International Conference on Control, Automation and Systems (ICCAS '10)*, pp. 1712–1717, October 2010.
- [5] D. P. Gillen, "Cooperative behavior schemes for improving the effectiveness of autonomous wide area search munitions," in *Cooperative Control and Optimization*, pp. 95–120, 2001.
- [6] Y. Guo and M. Balakrishnan, "Complete coverage control for nonholonomic mobile robots in dynamic environments," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '06)*, pp. 1704–1709, May 2006.
- [7] S. Waharte, A. Symington, and N. Trigoni, "Probabilistic search with agile UAVs," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '10)*, pp. 2840–2845, IEEE, May 2010.
- [8] S. Yanhang and Z. Zhou, "Cooperative control for target search, classification and attack for AUVs(attack uninhabited air vehicles)," in *Proceedings of the 26th Chinese Control Conference (CCC '07)*, pp. 99–102, July 2007.
- [9] V. Kumar, "Cooperative control of UAVs for search and coverage," in *Proceedings of the AUVSI Conference on Unmanned Systems*, vol. 2, pp. 1–14, 2006.
- [10] A. Ahmadzadeh, A. Jadbabaie, V. Kumar, and G. J. Pappas, "Multi-UAV cooperative surveillance with spatio-temporal specifications," in *Experimental Robotics: 10th International Symposium on Experimental Robotics*, vol. 39, pp. 5293–5298, 2006.
- [11] G. Gramajo and P. Shankar, "Path-planning for an unmanned aerial vehicle with energy constraint in a search and coverage mission," in *Proceedings of the IEEE Green Energy and Systems Conference (IGSEC '16)*, pp. 1–6, Long Beach, Calif, USA, November 2016.
- [12] D. P. Raymer, *Aircraft Design: A Conceptual Approach Fourth Edition*, American Institute of Aeronautics and Astronautics Inc, Reston, Va, USA, 2006.
- [13] J. John and D. Anderson, *Introduction to Flight*, McGraw-Hill, New York, NY, USA, 2005.
- [14] *DATCOM+ Pro Version 3.2*, Holy Cows, Orlando, Fla, USA, 2014.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

