Journal of Guidance, Control, and Dynamics
Vol. 45, No. 9, September 2022

Risk-Aware Path Planning for Unmanned Aerial Systems in a Spreading Wildfire

Rachit Aggarwal[*]
The Ohio State University, Columbus, Ohio 43210
Alexander Soderlund[†]
U.S. Air Force Research Laboratory, Albuquerque, New Mexico 87117
Mrinal Kumar[‡]
The Ohio State University, Columbus, Ohio 43210
and
David J. Grymin[§]
U.S. Air Force Research Laboratory, Wright–Patterson Air Force Base, Ohio 45433

https://doi.org/10.2514/1.G006365

Path planning for small unmanned aerial systems (SUAS) in the presence of poorly understood and dynamic obstacles is a challenging problem: for example, flight over a spreading wildfire. Evidential information fusion is used to estimate the current wildfire state and the resulting heat aura at flight level. This approach accounts for ignorance, which is a result of conflict among sensors operating in a harsh environment and a computational forecasting agent that uses a fire evolution model of inadequate accuracy. An SUAS is employed to visit locations of high conflict to provide additional situational awareness. Flight-level heat aura is modeled as a keepout zone with probabilistic boundaries for SUAS path planning. A novel unsupervised classification algorithm is developed to identify distinct obstacle boundaries within the estimated heat aura. Path planning is posed as a chance-constrained optimal control problem, which is transcribed to a nonlinear program via pseudospectral discretization. The results show that this approach can yield a family of solutions that elicit the risk associated with each mission design, and the appropriate choice of risk can aid in the generation of "keyhole paths."

Nomenclature

$\boldsymbol{b}$	=	boundary constraint
$\boldsymbol{c}$	=	path constraint
$\boldsymbol{D}$	=	differentiation matrix
E	=	radiative energy per unit area
$\mathbb{E}$	=	expectation operator
$\mathcal{F}$	=	set of events
$\boldsymbol{g}$	=	chance constraint
I_R	=	no-wind no-slope propagating heat flux
J	=	cost function
K	=	number of mesh intervals
$\mathcal{L}$	=	Lagrangian
ℓ_i	=	ith Lagrange polynomial
M	=	Mahalanobis distance
$\mathcal{M}$	=	Mayer cost
$m(\cdot)$	=	basic probability mass function
N_k	=	number of Legendre–Gauss–Radau collocation points in mesh interval k
P	=	probability
P_N	=	Nth-degree Legendre polynomial
Q_{ig}	=	heat required for ignition
$\dot{q}''$	=	incident heat flux, $\mathrm{kW \cdot m^{-2}}$
R	=	rate of spread of fire, ft/ min
$\mathbb{R}, \mathbb{R}(i, j)$	=	recombination favorability matrix and its (i, j)th element, respectively
R_c	=	radius of keepout zone
$r_{\min}$	=	minimum turn radius of the vehicle
T_f	=	absolute temperature of the flame, K
t, t_f	=	time and final time, respectively
$\boldsymbol{U}$	=	control approximation
u	=	control of Dubin's vehicle
$\boldsymbol{u}$	=	control
V	=	constant speed of Dubin's vehicle
x, y	=	position of Dubin's vehicle on the x and y axes, respectively
x_c, y_c	=	x and y positions, respectively, of the center of the keepout zone
$\boldsymbol{Y}$	=	state approximation
$\boldsymbol{y}(t)$	=	state
α	=	constant positive parameter for Bernstein method
α_+, α_-	=	constant positive parameters for split-Bernstein method
δ	=	fuzzy keepout zone boundary parameter
ϵ	=	risk parameter
ε	=	dimensionless effective heating number
ϵ_c	=	risk parameter for Dubin's vehicle
$\hat{\epsilon}$	=	emissivity
ϵ'	=	modified risk parameter for Bernstein method
Θ	=	frame of discernment
θ	=	turn angle
κ	=	degree of conflict among experts
$\boldsymbol{\Lambda}$	=	defect constraint Lagrange multiplier
$\boldsymbol{\lambda}$	=	costate
Ξ_α	=	piecewise exponential function
ξ	=	random variable, random sample
Π_t	=	matrix representing state of the environment at time t
$\pi_t(i, j)$	=	discrete state of the environment in the (i, j)th cell at time t
ρ_b	=	fuel bulk density
τ	=	time interval transformation

Received 2 August 2021; revision received 20 December 2021; accepted for publication 9 June 2022; published online 28 July 2022. Copyright © 2022 by Mrinal Kumar. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. All requests for copying and permission to reprint should be submitted to CCC at www.copyright.com; employ the eISSN 1533-3884 to initiate your request. See also AIAA Rights and Permissions www.aiaa.org/randp.

[*] Graduate Research Associate, Department of Mechanical and Aerospace Engineering, 171b ARC; aggarwal.112@osu.edu (Corresponding Author).

[†] NRC Research Fellow, Kirtland Air Force Base; soderlund.3@osu.edu.

[‡] Associate Professor, Department of Mechanical and Aerospace Engineering, 174 ARC; kumar.672@osu.edu. Associate Fellow AIAA.

[§] Controls Science Engineer, Control Science Center of Excellence, Air Vehicles Directorate; david.grymin.1@us.af.mil.

ϕ	=	configuration factor
ϕ_w, ϕ_s	=	wind and slope factors, respectively, affecting the wildfire spread rate
Ω	=	sample space for an event
$1_{(\cdot)}$	=	indicator function
$\varnothing$	=	empty set

I. Introduction

WILDFIRES are unplanned and unwanted fires, including lightning-caused fires, unauthorized human-caused fires, and escaped prescribed fire projects. Although they have some beneficial impacts on an ecosystem, they often threaten communities and force the evacuation of thousands of people [1]. Concurrently, there is increasing emphasis on the advancement of small unmanned aerial systems (SUAS) and their applications for reconnaissance, search, and rescue in emergent wildfires as they are positioned to replace human agents in such dangerous and/or repetitive missions [2–4]. For example, Fig. 1 illustrates the conditions during the early to midstages of the fire on Schultz Peak in Coconino National Forest near Flagstaff, Arizona in 2010. This scenario presents a set of heterogeneous threats, varying in severity and the extent to which they are understood and characterized. Moreover, threats are often transient. Consider an SUAS team tasked with the objective of providing situational awareness through the use of onboard thermal and/or visual equipment. This problem involves transient destinations (namely, locations of low fire-related information) along with a transient, poorly understood threat (namely, the heat aura at the flying vehicle's flight level emanating from the fire underneath).

A common deterministic approach to path planning is to approximate these threat regions using simple shapes such as ellipses, which must be penalized or excluded, depending on their (un)favorability attributes. If the complexity of the constraints increases further (for example, due to the presence of dynamic collision risks), conservative deterministic modeling can cause the solution space to shrink to a null set. In the face of these challenges, chance constraints offer a rigorous framework for recognizing that no-fly zones in a complex environment are ultimately approximations [5–8] so that there is merit in treating their boundaries as probabilistic barriers. While implying greater risk than the deterministic approach, the modified framework provides the decision maker the flexibility of prescribing mission-appropriate risk such that when the risk is set to zero, the solution space degenerates to that in the deterministic formulation [9]. Seen another way, one can generate a family of solutions parameterized by varying levels of assigned risk, providing the decision maker the ability to undertake missions with known risk versus return awareness. The favorable tradeoff is that undertaking a reasonable level of risk may open "keyholes" through which cost-efficient paths can be found; whereas in the deterministic approach, no such solutions exist.

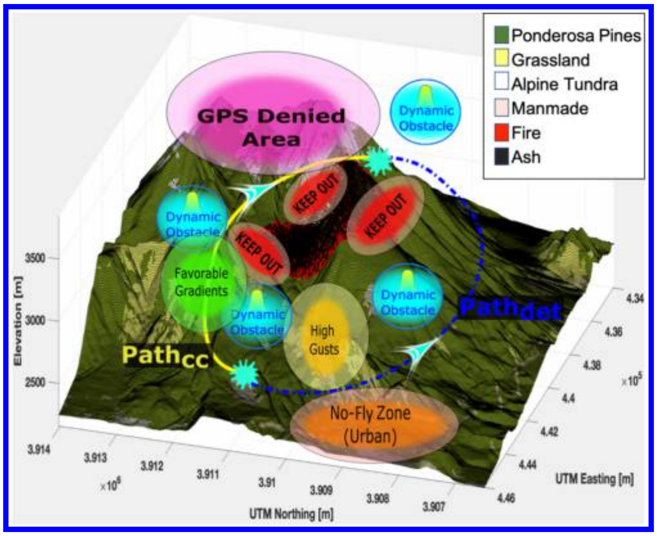


Fig. 1 Chance-constrained versus deterministic mission planning (GPS = Global Positioning System; UTM = Universal Transverse Mercator).

In this work, we develop a chance-constrained path-planning platform for recursive SUAS mission design in an evolving wildfire environment. An evidence-based framework built on the Dempster–Shafer theory of probable reasoning is used to construct real-time estimates of belief in wildfire. Information fusion includes a computational forecasting agent employing Markov chain models of wildfire evolution, ground-based temperature sensors, and SUAS mounted visual sensors. Heat emitted by the wildfire at the ground level is propagated to flight level via the principles of thermal radiation. A novel unsupervised clustering algorithm based on statistical formalism is developed to identify polygonal keepout regions within the probabilistic flight-level heat aura, representing a dangerous degree of heat flux above the SUAS operational threshold. Estimation uncertainty is projected onto the obstacle boundaries to model them as chance constraints with user-prescribed probability of violation. Path-planning objective functions include locations of low net information and/or high information conflict in the wildfire region. The optimal chance-constrained path-planning problem is transcribed to a nonlinear program (NLP) via pseudospectral discretization. Constrained Delaunay triangulation is performed to achieve rapid generation of good initial guesses for the optimal trajectory, which is provided as input to off-the-shelf gradient-based optimization software. It is ensured that the planning update frequency matches the update frequency of the wildfire so that the most current obstacle information is represented in the trajectory design. Numerical simulations show that by varying the risk of violation of heat flux chance constraints, a family of risk-aware trajectories can be generated, some of which represent keyhole trajectories that save significant mission cost. Preliminary results of this work were presented in a conference article (Ref. [10]).

This paper considers the problem of SUAS path planning in a wildfire. To appropriately formulate the path-planning problem, environmental situational awareness is required. Situational awareness is of central importance in a wildfire because the environment is without a map, uncertain, and dynamic. By necessity, one must consider the combined problem of environmental estimation and path planning. Both these problems are addressed in this work. In terms of methods of solution, there are three:

1) The first method is wildfire estimation: This is done in the evidential framework of the Dempster–Shafer theory using inputs from a Markovian computation agent (forecaster) and two types of sensing agents (temperature and vision).

2) The second method is obstacle estimation: Wildfire estimation leads to identification of heat-aura obstacles at flight level. This requires unsupervised clustering of the data obtained from wildfire estimation (method 1), followed by modeling of the cluster boundaries as chance constraints.

3) The third method is path planning: Using obstacle information from method 2, a chance-constrained optimal control problem is formulated and solved in a staged manner to obtain the SUAS trajectory.

Because the environment is dynamic, the described methods 1–3 are performed recursively (every 2 min in this work) to revise the SUAS trajectory in order to keep up with the changing environment. A key contribution of this paper is the integration of the three individual methodological elements described earlier. There are contributions in the individual methods as well. The clustering methodology is new. It represents a novel extension of the well-known k-means algorithm with the sole motivation of separating out islands in the obstacle data. The reason is that effective separation of islandlike clusters leads to maximization of flight space in between the obstacles. To serve this objective, the output of optimal k means is subjected to a process of targeted splitting. A recombination process is designed to reduced oversplitting. A second elemental contribution of this work is the use of a constrained Delaunay triangulation approach to rapidly generate good initial guesses for the underlying nonlinear program. The combined approach is shown to result in keyhole trajectories that save mission cost.

Further work is required to evaluate the computational complexity of the coupled environmental estimation and path-planning problem. In its present form, the platform is not a real-time tool and the greatest computational burden is encountered at the wildfire estimation step (method 1 in the preceding list). Improving the efficiency of this step will help elevate the usefulness of the proposed platform as an online (real-time) tool.

II. Estimation of a Transient Wildfire Obstacle

A. Markovian Wildfire Forecasting

To monitor and continuously track an evolving wildfire, it is necessary to adequately predict the probable locations of the fire front based on the wildfire's spreading dynamics. This is done through a wildland fire forecasting algorithm for which the inputs are the assumed environmental conditions (surface wind velocity, local topography, types of vegetative fuel, humidity, etc.) and the output is the normalized frequency that a fire will arrive at a certain location and time in the environment. The environmental parameters are employed to approximate the wildfire's average rate of spread based on Rothermel's model [11,12]:

$$R = \frac{(I_R)(1 + \phi_w + \phi_s)}{\rho_b \varepsilon Q_{ig}} \tag{1}$$

which computes the spread rate R (in feet per minute) as a ratio of the heat source (the no-wind/no-slope propagating heat flux I_R amplified by the wind and slope factors, ϕ_w and ϕ_s) and the heat sink, which is the heat required for ignition Q_{ig} multiplied by the bulk density ρ_b (i.e., the amount of fuel) and the dimensionless effective heating number ε. Although this deterministic model is simple to implement, many of the required environmental inputs are inherently nondeterministic, such as the surface wind velocities [13], the distribution of vegetation, and the act of combustion. In this case, the evolving wildfire can be treated as a stochastic process (see the work of Boychuk et al. [14]) by representing each location within an ongoing wildfire environment as a discrete state in an evolving Markov chain. The time-varying state of the entire wildfire environment is represented as an $m_1 \times m_2$ two-dimensional array where each cell within the lattice represents a square-shaped geographic region with the Cartesian coordinate pair (i, j) representing its center. The state of any cell at coordinate (i, j) and at time step t is described by one of four discrete values: $\pi_t(i, j) \in \chi$, where

$$\chi = \{\text{Vegetation, Fire, Ash, Noncombustible}\}$$

It follows that the state of the entire environment at time t is denoted as

$$\Pi_t = \begin{bmatrix} \pi_t(1,1) & \pi_t(1,2) & \dots \\ \vdots & \ddots & \\ \pi_t(m_1,1) & & \pi_t(m_1,m_2) \end{bmatrix}$$

where the state of each cell $\pi_t(i, j) \in \Pi_t$ is assumed to transition into a succeeding state at time $t + 1$, $\pi_{t+1}(i, j)$ according to a Markov process:

$$\begin{aligned} &P(\pi_{t+1}(i,j) \in \chi | \pi_1(i,j), \pi_2(i,j), \dots, \pi_t(i,j)) \\ &\quad = P(\pi_{t+1}(i,j) \in \chi | \pi_t(i,j)) \end{aligned} \tag{2}$$

The probability of a given cell (i, j) transitioning from state x to state y,

$$P(\pi_{t+1}(i,j) = y | \pi_t(i,j) = x)$$

denoted $P(i, j)_{xy}$, is dependent on its current state, the states of its adjacent eight cells, and the surrounding environmental attributes. For notational simplicity, the set of states is replaced with the following set of numerical labels:

$$\{\text{Fuel} \rightarrow 1, \text{Fire} \rightarrow 2, \text{Ash} \rightarrow 3, \text{Noncombustible} \rightarrow 4\}$$

To compute the transitional probability from fuel to fire, we use the minimum and maximum rates of spread at each location provided via Eq. (1):

$$R(i,j) \in [R(i,j)_{\min}, R(i,j)_{\max}]$$

Given a cell assumed to be in a combustible fuel state at time step t [i.e., $\pi_t(i, j) = 1$] that borders at least one cell in a fire state [$\pi_t(i \pm 1, j \pm 1) = 2$], there is a nonzero (one-step) transition probability that $\pi_t(i, j) = 2$ due to the wildfire's spreading behavior. In the stochastic models employed by Refs. [14,15], the time τ for a cell's state to transition from the fuel to the fire state can be sampled from an exponential probability distribution with rate parameter $\lambda^{(t)}(i, j)$, or $\tau \sim \text{Exp}(\lambda^{(t)}(i, j))$. The construction of this rate parameter for each cell (i, j) is given as

$$\lambda^{(t)}(i,j) = \sum_{n \in N((i,j))} \frac{c^{(t)} \lambda_0(i,j)}{1 - \cos(\alpha(n) - w_d^{(t)}(i,j))[1 - (\lambda_0/\lambda_m)]} \tag{3}$$

The value of $\lambda^{(t)}(i, j)$ is dependent on its current state, the states of its eight adjacent neighboring cells,

$$N((i,j)) = \{(i+1,j), (i+1,j+1), \dots (i-1,j-1)\}$$

and the current wind directions in cell (i, j) and its neighbors. In Eq. (3), the wind direction within cell (i, j) at time k, $w_d^{(t)}(i, j) \in (0, 2\pi]$, is subtracted from the relative angle $\alpha(n)$ between the center of cell (i, j) and that of its neighboring cell's center $n \in N((i, j))$, which can either amplify or attenuate the probability of fire spread. The coefficient $c^{(t)}$ is tuned to model time-dependent fire intensity, and the coefficients $\lambda_0(i, j)$ and λ_m are empirically determined minimum and maximum rate parameters based on the fuel type, slope, and wind speed. In Refs. [14,15], the fuel cell was assumed to transition if the sampled time τ fell below some prescribed time-step threshold ΔT. The present work differs from these aforementioned approaches by instead computing the fuel cell's transitional probability as the cumulative distribution function of this exponential distribution; i.e.,

$$\mathbb{P}(i,j)_{12} = 1 - \exp(-\lambda^{(t)}(i,j)\Delta T)$$

Likewise, the transitional probability that the fuel cell will remain in the fuel state at time $t + 1$ is $\mathbb{P}(i, j)_{11} = \exp(-\lambda^{(t)}(i, j)\Delta T)$.

To compute the transitional probabilities $\mathbb{P}(i, j)_{22}$ and $\mathbb{P}(i, j)_{23}$, we introduce the *mean hitting time* $t^A_{\pi_0}$, defined as the expected number of time steps required for a Markov chain to reach a set of states A given that the chain starts from state $\pi_0 \in A$:

$$t^A_{\pi_0} = \begin{cases} 0 & \text{for } \theta_0 \in A \\ 1 + \sum\limits_{\pi_j \notin A} \mathbb{P}(i,j)_{\pi_0 \pi_j} t^A_{\pi_j} & \text{for } \theta_0 \notin A \end{cases} \tag{4}$$

Assuming that each cell represents a square area of homogeneous fuel, a newly lighted cell in the fire state can be modeled to consume all fuel in an expected time of fuel-type-dependent (and hence cell-dependent) $B(i, j)$ time steps. Thus, by setting $t_2^3 = B(i, j)$, the preceding relation can be algebraically manipulated to yield the probability that the fire continues to burn through the next time step:

$$\mathbb{P}(i,j)_{22} = \frac{B(i,j) - 1}{B(i,j)}$$

Hence,

$$\mathbb{P}(i,j)_{23} = 1 - \mathbb{P}(i,j)_{22} = \frac{1}{B(i,j)}$$

Additional details on derivation of $\mathbb{P}(i,j)_{11}$ and $\mathbb{P}(i,j)_{12}$ are provided in the works of Boychuk et al. [14] and Krougly et al. [15]. On the other hand, the probability that a location will continue to burn $\mathbb{P}(i,j)_{22}$ is given via the consumption rate of fuel: see the previous study by Soderlund et al. [16]. Obtaining practical values for the parameters λ_0, λ_m, and $c^{(k)}$ can be done by extensively reverse tuning them to each cell's minimum and maximum fire spread rates, $R(i,j)_{\min}$ and $R(i,j)_{\max}$, which are themselves computed from known fuel and topography models, such as the tables seen in Ref. [17].

By expanding this Markov chain operation across all cells within Π_t, the entire fire disturbance is propagated. Figure 2 displays a Markovian simulation that attempts to replicate a 2019 fire that crossed the Rio Grande River in Los Chavez, New Mexico due to a westerly wind. For this simulation, National Oceanic and Atmospheric Administration (NOAA) wind data and United States Geological Survey (USGS) vegetation and elevation maps were collated to provide appropriate environmental inputs. To generate a probabilistic representation of the fire based on the assumed environmental parameters (Fig. 2 shows a single instance of the Markov chain), it is necessary to simulate many fire propagation instances (for example, M Monte Carlo simulations) at every time step t. For example, the frequentist probability of fire presence within a location (i,j) at time t is given as the number of instances f that the cell transitions to fire:

$$P(\pi_t(i,j) = 2) = \lim_{M\to\infty}(f/M)$$

As an extension, the vector

$$[P(\pi_t(i,j) = 1), P(\pi_t(i,j) = 2), P(\pi_t(i,j) = 3)]$$

is treated as the prior probability distribution for each cell. Note that $P(\pi_t(i,j) = 4)$ is not included in the probability distribution because it represents the "infinitely noncombustible type," e.g., water locations that make up the river in the central portion of the map. There is no transition *into* cell type 4, and there is no transition *out of it*. Because these facts are known a priori, there is no reason to include them in the distribution. This improves the computational efficiency of fire propagation because one can eliminate a certain number of locations from possible fire spread in the simulations.

B. Information Fusion via Evidential Reasoning

Although the wildfire representation model described in Sec. II.A provides a method of propagating the locations of the flame front over time (as seen in Fig. 2), the inherent stochasticity of the dynamics illuminates the necessity of updating the wildfire's belief state representation based on real-time measurements from sensing agents. This work considers two sources of wildfire observation: 1) a temperature sensor agent that activates due to an encroaching fire's radiation, and 2) a thermal vision sensor agent that uses high-resolution temperature readings in its field of view to identify wildfire presence. Both sources are imperfect: the temperature sensor can suffer from reporting false positives, whereas the vision sensor (mounted on the SUAS) must contend with various constraints that hinder its available areas to survey. Wildfire belief state estimation is performed by combining the forecaster's predicted probability of fire in a given location $(i,\ j)$ with the beliefs of either the temperature or vision sensor, or both if available. This fusion of beliefs is done under the framework of evidential reasoning through the Dempster–Shafer theory (DST) of probable reasoning.

The Dempster–Shafer theory is applicable when considering a question with multiple (but finite) possible explanations. The finite set of n mutually exclusive and collectively exhaustive hypotheses that act as adequate answers to this question is defined as the *frame of discernment*: $\Theta = \{\theta_1, \theta_2, \ldots, \theta_i, \ldots, \theta_n\}$, where exactly one hypothesis θ_i is the truth. Within Kolmogorov's conventional axiomatic probability theory, the probability of event θ_i and its complement $\tilde{\theta}_i$ have the relation $P(\theta_i) = 1 - P(\tilde{\theta}_i)$. Under DST, however, the metrics of belief are allocated among members (known as propositions) of the power set 2^Θ, and a degree of belief can be assigned to a set containing conflicting hypotheses. This is useful in scenarios where observations indicate sets of events rather than events themselves.

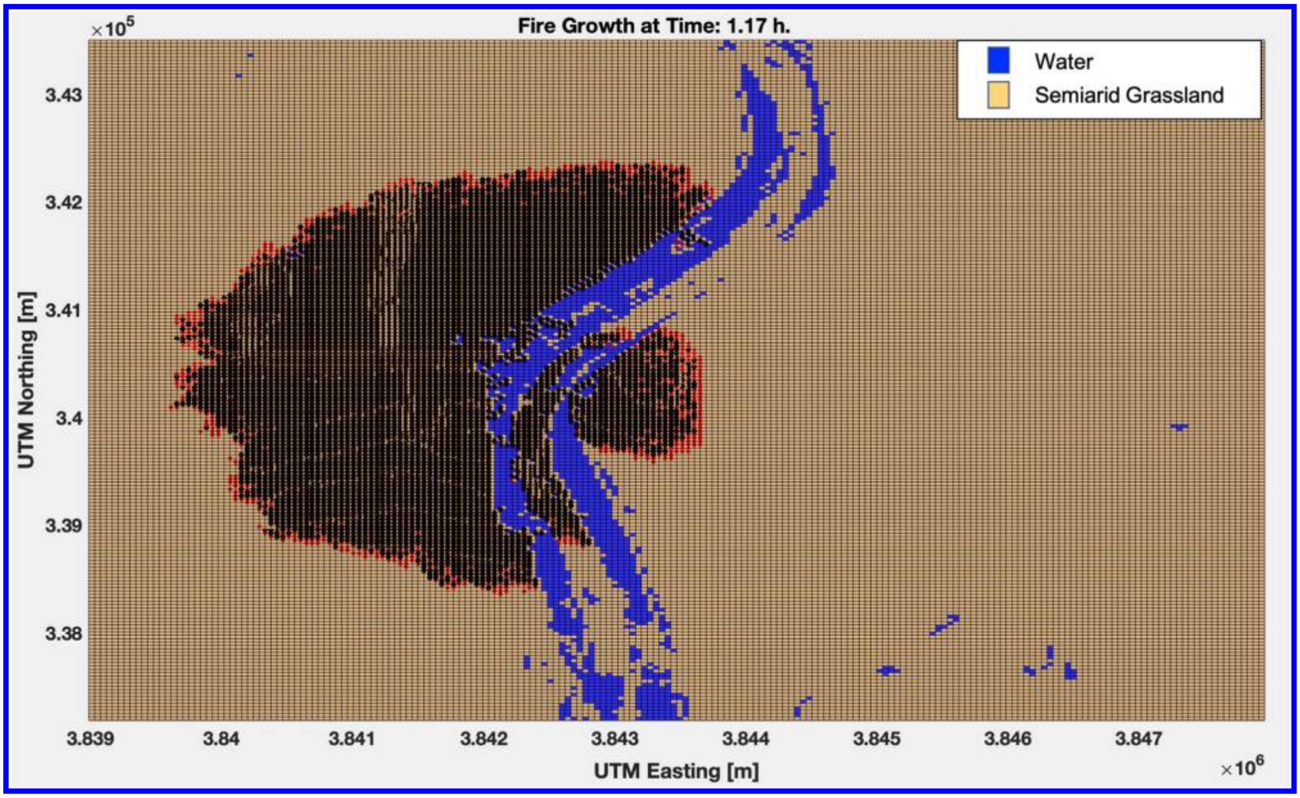


Fig. 2 Wildfire simulation: currently burning locations (red), ash locations (black), noncombustible Rio Grande River (blue). The remaining brown cells are grassland fuel.

It is useful to imagine belief as a finite resource that neither shrinks nor grows but is instead rationed appropriately to each proposition, given the available information. The amount of belief assigned to a particular proposition A is based on the current evidence in support of A. This numerical belief for proposition A is quantified through the basic probability number (BPN), denoted $m(A)$. This quantity is calculated according to the basic probability assignment (BPA) function m: $2^{\Theta} \rightarrow [0, 1]$:

$$\sum_{A \subseteq \Theta} m(A) = 1 \quad \text{with} \quad m(\varnothing) = 0 \tag{5}$$

The summation of all basic probability numbers of the subsets of A, which is a quantity known simply as the belief $\mathrm{Bel}(A)$, is constructed as

$$\mathrm{Bel}(A) = \sum_{B \subseteq A} m(B) \tag{6}$$

To illustrate an example applicable to the wildfire scenario discussed thus far, it is possible to assign belief to the following proposition set:

$$\{\text{Veg.}, \{\text{Veg.}, \text{Fire}\}, \text{Fire}, \{\text{Fire}, \text{Ash}\}, \text{Ash}, \Theta\}$$

where $\Theta = \{\text{Veg.}, \text{Fire}, \text{Ash}\}$ represents complete ignorance of a location's state. The beliefs of multiple experts are fused through Dempster's rule of combination [18]. Consider the basic probability numbers supplied by experts $\mathbb{T}$ and $\mathbb{F}$ over the power set 2^{Θ}. The combined BPN of both experts regarding proposition A is calculated via Dempster's rule:

$$m_{\mathbb{C}}(A) = K \sum_{A_i \cap B_j = A} m_{\mathbb{T}}(A_i) m_{\mathbb{F}}(B_j) \tag{7}$$

where the normalization constant $K = (1/1 - \kappa)$ is computed from the degree of conflict $\kappa \in [0, 1]$ between the BPAs of the two experts $\mathbb{T}$ and $\mathbb{F}$:

$$\kappa = \sum_{A_i \cap B_j = \varnothing} m_{\mathbb{T}}(A_i) m_{\mathbb{F}}(B_j) \tag{8}$$

The resulting belief function given by this combined basic probability assignment $m_{\mathbb{C}}(A)$ is known as the orthogonal sum of the beliefs of experts $\mathbb{T}$ and $\mathbb{F}$ and is denoted $\mathrm{Bel}_{\mathbb{C}} = \mathrm{Bel}_{\mathbb{T}} \oplus \mathrm{Bel}_{\mathbb{F}}$. Furthermore, the combined beliefs of E_N experts regarding proposition A may be computed by performing Eq. (7) recursively:

$$\mathrm{Bel}_{\mathbb{C}}(A) = (((\mathrm{Bel}_{\mathbb{T}}(A) \oplus \mathrm{Bel}_{\mathbb{F}}(A)) \oplus \mathrm{Bel}_{E_3}(A)) \ldots) \oplus \mathrm{Bel}_{E_N}(A)$$

This work performs information fusion via Dempster's rule of combination by fusing the beliefs of three such "experts" at every time step: 1) the Monte Carlo forecaster, 2) the ground-level temperature sensors, and 3) the airborne vision sensors. The forecaster's belief assignment into the fire hypothesis at each location is derived from the cell's prior probability distribution (created via Monte Carlo sampling). Although the forecaster's beliefs are inherently based on the fire's spreading dynamics and environmental conditions, the temperature and vision sensors' beliefs are apportioned via their received measurements. For example, a temperature agent reporting high-temperature readings for an extended period of time would result in a higher belief in either the fire or ash possibilities. Interested readers should refer to previous work by Soderlund et al. [16] for a more detailed explanation of how each of these expert basic probability assignments are constructed. After belief combination is performed as in Eq. (7), across each location, an updated estimate of the wildfire's belief state is available for further propagation. Although this is useful for monitoring the probable locations of the fire perimeter, we are additionally interested in the output conflict value. High conflict values between experts (such as the forecaster and a temperature sensor) may indicate that one agent has inadequate or incorrect information. An additional observation made by an aerial sensor may alleviate this discrepancy, although its route to this conflict location must be feasible. To ensure consistent operation of path following, sensing, processing, and communication of data, the computing hardware encased in each SUAS must not encounter prolonged thermal radiation. Thus, we seek to maintain the health of each SUAS by calculating paths containing safe levels of radiation that also intersect locations of interest. The calculation of this (probabilistic) radiation constraint requires an understanding of the heating effects of flame surfaces on their surrounding environments, which will be discussed in the following section.

III. Probabilistic Characterization of the Heat Aura

A. Characterization of Heat Aura as Obstacles

Although convection and radiation are the main methods of heat loss experienced by the wildfire object, radiation becomes the dominant form of heat transfer for surfaces exceeding 400°C [19]. As the flame temperatures of a wildland fire can vary between 800 and 1000°C, we are primarily concerned with the effect of the fire's radiative heat aura that extends up into the atmosphere and hinders the maneuverability of the SUAS. Each burning location is treated as a radiative surface that emanates energy per unit area at a rate according to the Stefan–Boltzmann law:

$$E = \sigma_s \hat{\epsilon} T_f^4 \tag{9}$$

where T_f is the absolute temperature of the flame in kelvins, $\hat{\epsilon} \in [0, 1]$ is the emissivity. and $\sigma_s = 5.67 \times 10^{-8}\ \mathrm{W/(m^2 \cdot K^4)}$ is the Stefan–Boltzmann constant. Conceptually, this rate of energy loss (referred to as the *heat flux* for the remainder of this work)is radiated in all directions from the flame surface A_1. The radiant heat flux that is incident upon a small element of a secondary surface A_2 due to the flame's flux is given as

$$\dot{q}'' = E\phi \tag{10}$$

where ϕ, which is known as the configuration factor, is a dimensionless quantity that describes the geometric relation (e.g., distance and orientation) between surfaces A_1 and A_2. Within the field of fire safety, the incident heat flux $\dot{q}''$ is often used as the metric to indicate the level of danger: Wildland firefighters can tolerate a maximum incident heat flux of $7\ \mathrm{kW \cdot m^{-2}}$ [20], and wood will ignite in a matter of seconds with a heat flux of $20\ \mathrm{kW \cdot m^{-2}}$ [19]. Importantly, when multiple radiating surfaces with heat flux E are acting upon the same distant surface, their configuration factors are additive. Recall that the environmental state representation Π_t defined in Sec. II.A can be partitioned into the following sets representing the vegetation, fire, ash, and noncombustible cells at time t: $\{\mathcal{V}_t, \mathcal{F}_t, \mathcal{S}_t, \mathcal{U}_t\}$. Because the locations of the flame surfaces [i.e., $\pi_t(i, j) = 2$] are being tabulated as members in $\mathcal{F}_t$ in each Monte Carlo run, the incident flux impinging on a surveillance drone at coordinate (i, j, j) can be modeled as

$$\dot{q}''(i, j, k) = E \sum_{(u,v) \in \mathcal{F}_t}^{|\mathcal{F}_t|} \phi(u, v) \tag{11}$$

where the configuration factor $\phi(u, v)$ between a fire surface with a width of w and an unmanned aerial vehicle (UAV) with an elevation of $z = \zeta k$ and a lateral translation of $a = \zeta \sqrt{(i - u)^2 + (j - v)^2}$ relative to a ground-level cell (u, v) is given as

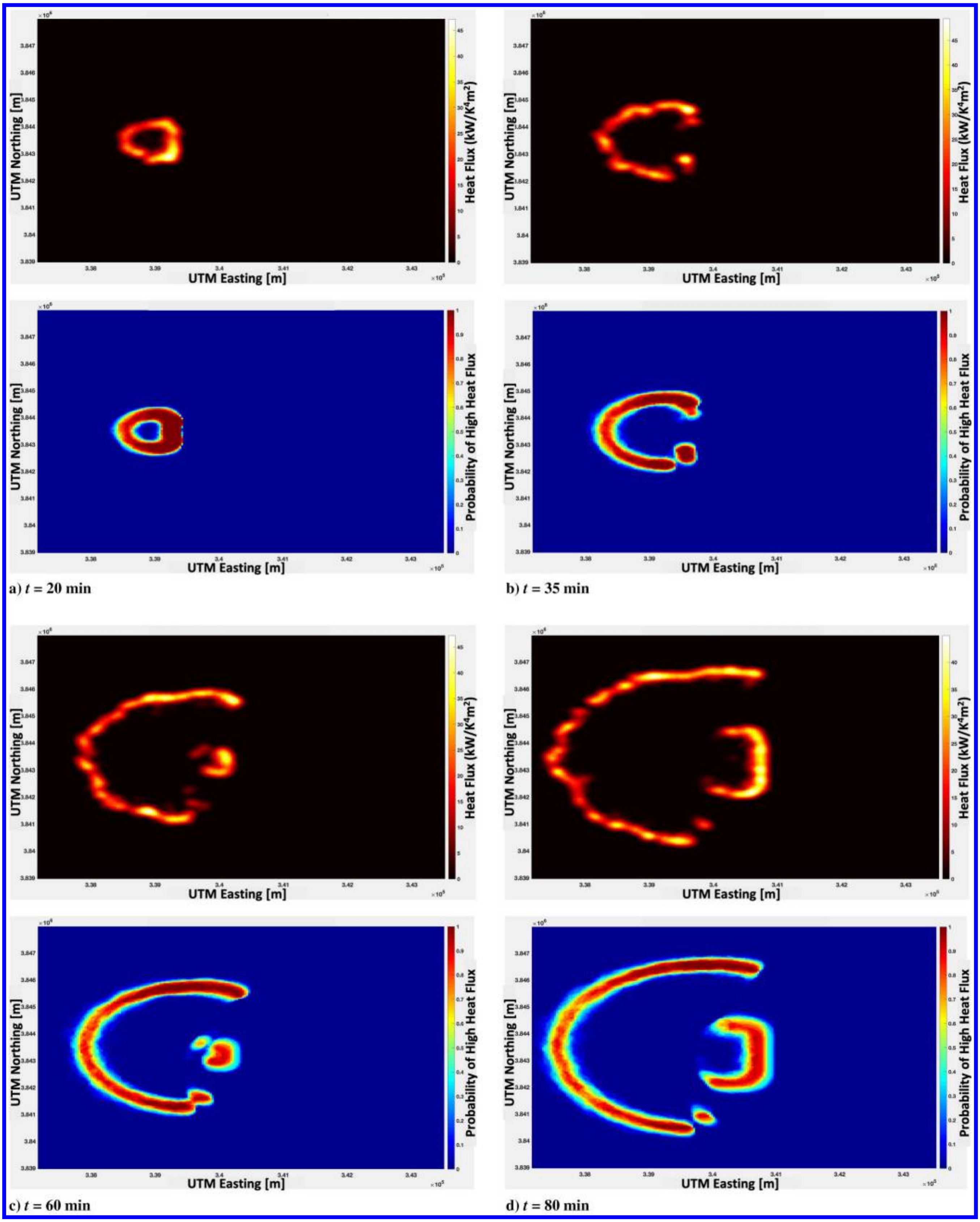
$$\phi(u, v) = \frac{1}{2}\left[1 - \frac{\upsilon - 2(w/a)^2}{\sqrt{\xi^2 - 4(w/a)^2}}\right] \tag{12}$$

where

$$\upsilon = 1 + \left(\frac{w}{a}\right)^2 + \left(\frac{z}{a}\right)^2$$

(see Ref. [21]) and ζ is the separation distance (in meters) between the centers of adjacent cells. Recall that the forecaster outlined in Sec. II.A outputs the probability that a given cell (i, j) at time t will take on the fire state. Analogously, the heat aura generated by a wildfire is characterized as a probabilistic rather than a deterministic object. Note that the aforementioned heat model is imperfect; and alternative (potentially more precise) heat-aura constructions are possible, such as the use of numerical integration mentioned in Ref. [20].

For path-planning purposes, we seek the frequency that a spatiotemporal location exhibits a heat flux at flight level greater than a critical value. The top half of Fig. 3 depicts the heat flux (ranging from 0 to 50 $\mathrm{kW \cdot m^{-2}}$) that an aerial drone would encounter over all locations at flight levels at four separate time stamps. The bottom half



a) $t = 20$ min

b) $t = 35$ min

c) $t = 60$ min

d) $t = 80$ min

Fig. 3 Top panel at each time label: progression of flight-level (120) heat aura for a single wildfire simulation. Bottom panel at each time label: occurrence rate of critical heat flux in spatiotemporal locations over 100 Monte Carlo runs.

of Fig. 3 shows the probability that each location will exhibit a heat flux over the critical value $[\dot{q}''(i,j,k) \geq 5\ \mathrm{kW \cdot m^{-2}}]$. These will form the basis of the keepout zones discussed in the following sections.

B. Unsupervised Clustering

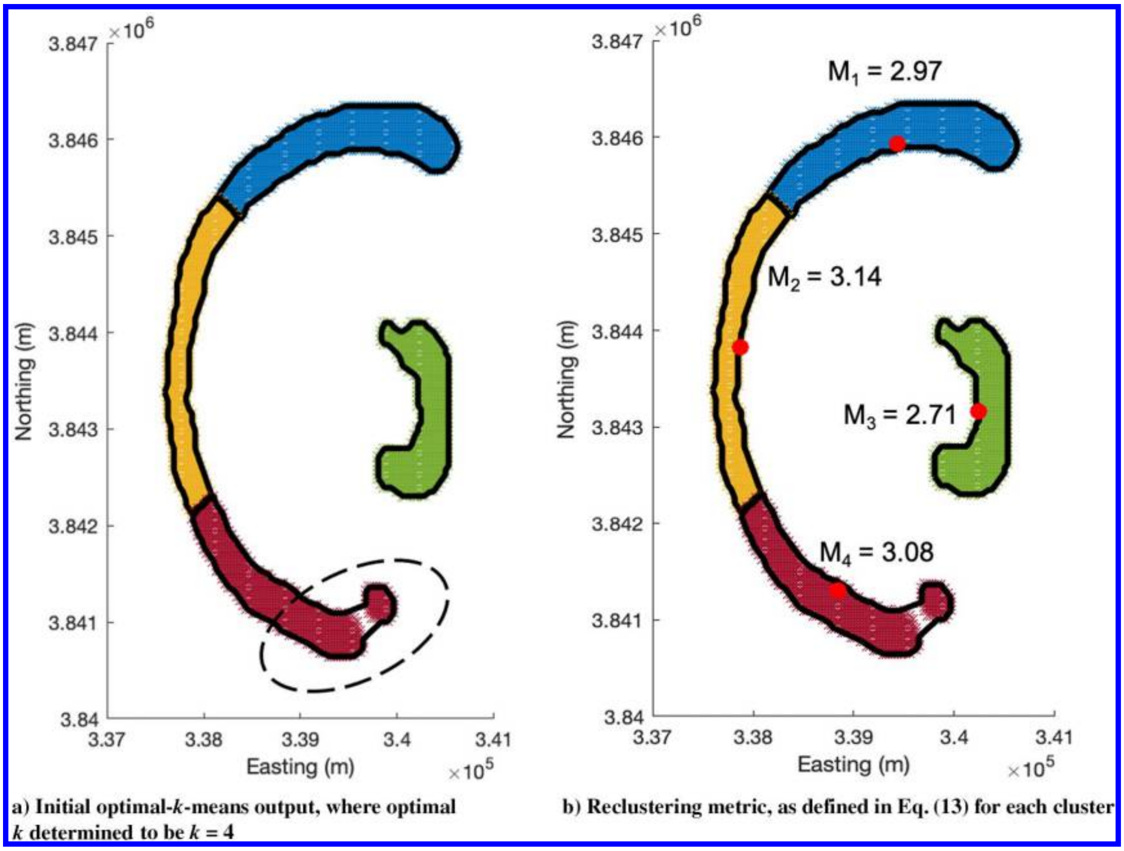
Clustering of unstructured data is counted among the most difficult learning problems due to its unsupervised form and the open-ended nature of the question: How many clusters exist in the data? For obstacle characterization, it is important to separate obstacles from each other so that available flight space can be maximized. See the lower portion of Fig. 4a as an example of when this fails. This is difficult because it is unknown how many clusters may be present in the unstructured obstacle data. In this work, the initial data are assumed to be binary (obstacle absent or present), which are to be clustered first, followed by assignment of an appropriate probability distribution to the resulting cluster boundaries. We present a novel methodology for unsupervised clustering based on statistical formalism, building on the well-known k-means algorithm [22]. Also known as Lloyd's algorithm, the k-means algorithm partitions the data into k clusters identified by their centroids, where the value of k must be specified at the start. However, as already mentioned, the value of k is unknown in the present problem. The traditional workaround is to design an optimization outer loop that aims to maximize the cluster quality by spanning over a range of arbitrarily selected values of k (for example, $k = 1$ to $k = 20$): we refer to this approach as the *optimal-k means*. The cluster quality is roughly defined using metrics such as "silhouette" of MATLAB©, which measures the similarity of points to other points in their own cluster in comparison with points in other clusters: usually in terms of Euclidean distances. This optimization approach often fails because visual cluster quality (as ascertained by humans) is often highly sensitive to variations in the numerical value of such metrics, in the sense that the visually optimal cluster assignment (including the value of k) often corresponds to a local optimum.

The new method developed in this work achieves optimal unsupervised clustering in a recursive manner. The method begins with execution of the optimal-k-means algorithm as described earlier in this paper. Then, each resulting cluster is analyzed in terms of its statistical properties and flagged for further clustering (splitting or reclustering). The flagging process begins by computing the statistical mean $\boldsymbol{\mu}_i$ and covariance $\boldsymbol{\Sigma}_i$ of each cluster, followed by determination of the maximum Mahalanobis distance of all points within the cluster with respect to these statistics:

$$M_i = \max_{j \in \mathcal{C}_i} \underbrace{\sqrt{(\boldsymbol{x}_j - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x}_j - \boldsymbol{\mu}_i)}}_{M_{ij}} \qquad (13)$$

where $\mathcal{C}_i$ denotes the ith cluster, and M_i represents the maximum Mahalanobis distance among all points in $\mathcal{C}_i$ with respect to $(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$. $\mathcal{C}_i$ is flagged for reclustering if $M_i > M_{\mathrm{cut}}$, where M_{cut} is a suitable threshold (chosen to be less than three, e.g., $M_{\mathrm{cut}} = 2.3$). This metric helps determine the statistical appropriateness of assigning data points to a particular cluster. Figure 4b depicts the maximum Mahalanobis distance metric M_i for each cluster resulting from optimal-k-means clustering. In this figure, the centroid of each cluster is identified as a red circle. All clusters flagged for reclustering are passed into the optimal-k-means algorithm again, and the process repeats until no more clusters flagged for reclustering remain. A possible criticism of the metric given in Eq. (13) is that a single outlier member can cause a cluster to be flagged for further splitting. In the present use case, that is not necessarily a bad thing because the idea is to separate out as many islands as possible such that interobstacle space can be maximized for path planning. In a more general setting, an appropriately high q quantile can be employed instead of "max" as in Eq. (13). For example, one may define M_i as the $q = 0.96$ quantile value of the Mahalanobis distance set $\{M_{ij}\}$.

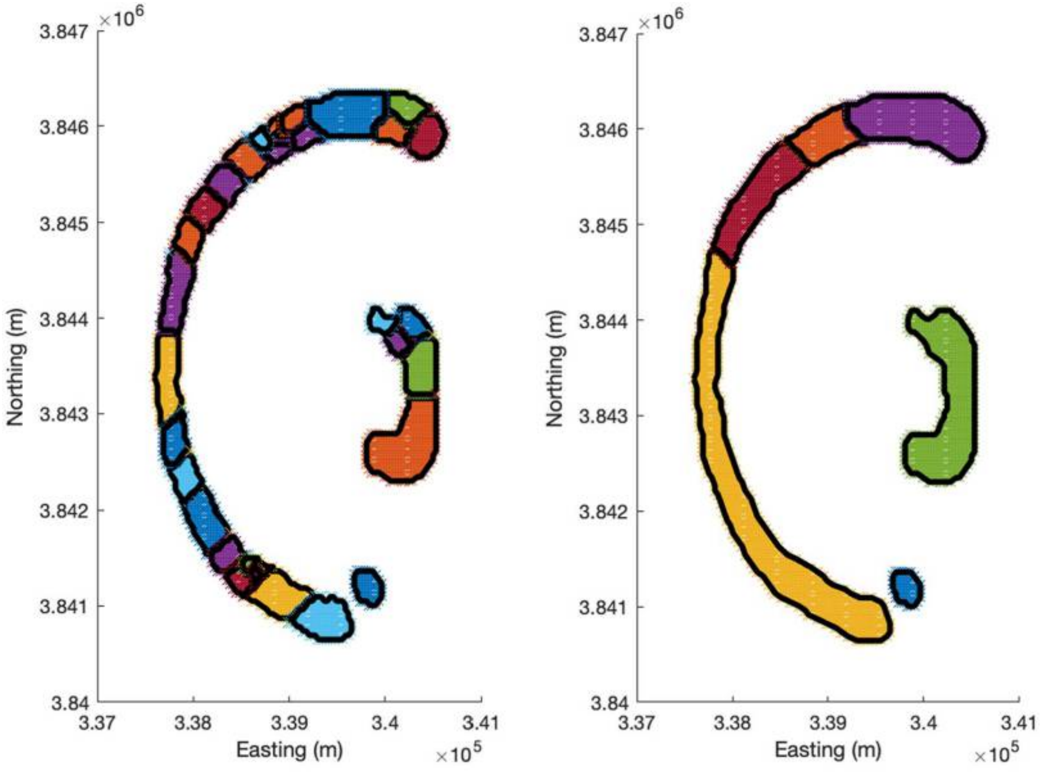
At the end of this described splitting process, there will invariably be more clusters than are needed or warranted. See Fig. 5a. A total of 48 clusters results after 39 iterations of cluster splitting using $M_{\mathrm{cut}} = 2.3$. This is clearly not desirable. Therefore, as the final step, the set of oversplit clusters is analyzed statistically for recursive



a) Initial optimal-*k*-means output, where optimal *k* determined to be *k* = 4

b) Reclustering metric, as defined in Eq. (13) for each cluster

Fig. 4 Outcome of optimal-*k*-means clustering.



a) Clusters formed after recursive splitting in the new algorithm (iterations=39 and clusters=48)

b) Clusters remaining after recursive recombination in the new algorithm (iterations=1 and clusters=6)

Fig. 5 Clustering for obstacle identification using raw data from wildfire forecasting.

recombination. First, a "recombination favorability matrix $\mathbb{R}$ is constructed. The (i, j)th element of this matrix is defined as follows:

$$\mathbb{R}(i, j) = \max\left\{\sqrt{(\bar{\boldsymbol{\mu}}-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\bar{\boldsymbol{\mu}}-\boldsymbol{\mu}_i)}, \sqrt{(\bar{\boldsymbol{\mu}}-\boldsymbol{\mu}_j)^T\boldsymbol{\Sigma}_j^{-1}(\bar{\boldsymbol{\mu}}-\boldsymbol{\mu}_j)}\right\} \tag{14}$$

i.e., the greater of the Mahalanobis distance of $\bar{\boldsymbol{\mu}}$ with respect to the statistics $(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ of $\mathcal{C}_j$ and $(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ of $\mathcal{C}_i$. Here, $\bar{\boldsymbol{\mu}}$ is the centroid (mean) of the combined cluster $(\mathcal{C}_i, \mathcal{C}_j)$. In essence, the defined metric measures the statistical distance of the combined mean with respect to the individual statistics of the clusters under consideration. The recombination favorability matrix helps identify pools of clusters that can be combined together, for example, as follows:

$$\mathcal{P}_i = \arg_j\{\mathbb{R}(i, j) \le R_{\text{comb}}\}$$

This condition identifies all clusters that have proximity to the ith cluster. The greater the value chosen for R_{comb}, the more propensity the algorithm will have for pooling together clusters. A value close to but less than 3.0 is reasonable and must be picked, depending on the application. In the example shown in Fig. 5b, R_{comb} was chosen to be 2.8, resulting in a total of six clusters. This result is adequate for path planning.

Clearly, there are two main tuning parameters in the previously described unsupervised clustering approach: M_{cut} and R_{comb}. For M_{cut}, the guiding principle is that the higher the chosen value, the *less* the algorithm is inclined to split existing clusters. In other words, a large M_{cut} implies that clusters containing points with high Mahalanobis distances are allowed to exist. On the other hand, a large value of R_{comb} makes the algorithm *more* inclined to pool together neighboring clusters. More research is required to develop tuning principles for these parameters and is expected to be application dependent.

As a final step, the cluster boundaries are examined in relation to the flight vehicle size. A sensible simplification is to merge boundaries that are separated by less than the characteristic length of the SUAS. Assuming the flight vehicle safety zone to be defined as a bubble with a diameter of 25 m, qualifying cluster boundaries are merged to result in the final result shown in Fig. 6.

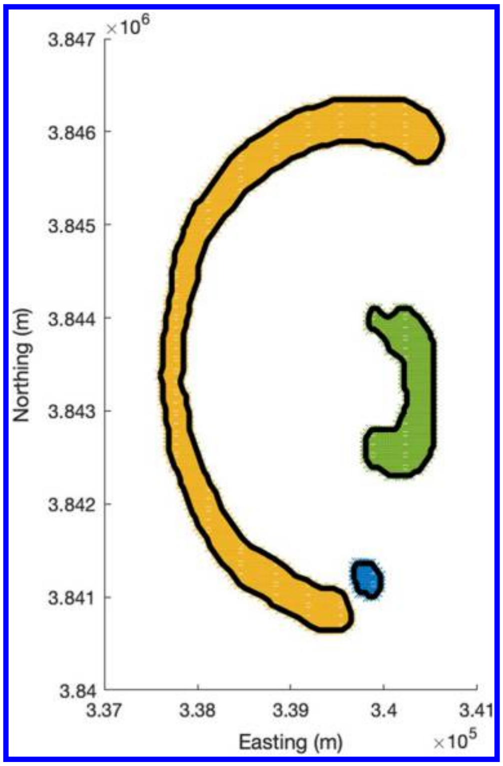


Fig. 6 Merging boundaries that are closer to each other than the vehicle characteristic length.

The previously described unsupervised clustering process has two tuning parameters: M_{cut} and R_{comb}. The former (M_{cut}) controls the splitting process, whereas the latter (R_{comb}) controls the recombination process. The parameter M_i [Eq. (13)] captures the maximum statistical distance among member particles in the cluster $\mathcal{C}_i$ from the cluster mean. Note that $M_{\mathrm{cut}} = 3$ suggests that in order for cluster $\mathcal{C}_i$ to be flagged for splitting, its weakest member must be at least 3σ away from the mean (outside the 0.003 percentile for one-dimensional data). By choosing $M_{\mathrm{cut}} < 3$, the idea is to enforce a somewhat more stringent (consequently, more conservative) threshold on clusters to be flagged for splitting. If M_{cut} is chosen too high (greater than three), only a handful of clusters will be flagged for splitting. Data islands of the kind shown in Fig. 4a would not be broken up. This is further illustrated in the example shown in Fig. 7d. On the other hand, if M_{cut} is too low (e.g., $M_{\mathrm{cut}} \approx 2$), most clusters will be flagged for splitting, resulting in a very large number of sparse clusters: see Fig. 7a. Experience suggests that a good balance is achieved when M_{cut} is chosen to be around 2.5. Figure 7 shows the comparative performance of various M_{cut} values.

After splitting is complete, cluster recombination is performed using the described favorability matrix [Eq. (14)]. The following steps are involved:

1) For each pair of split clusters, the centroid of the combined cluster is computed $\bar{\mu}$.

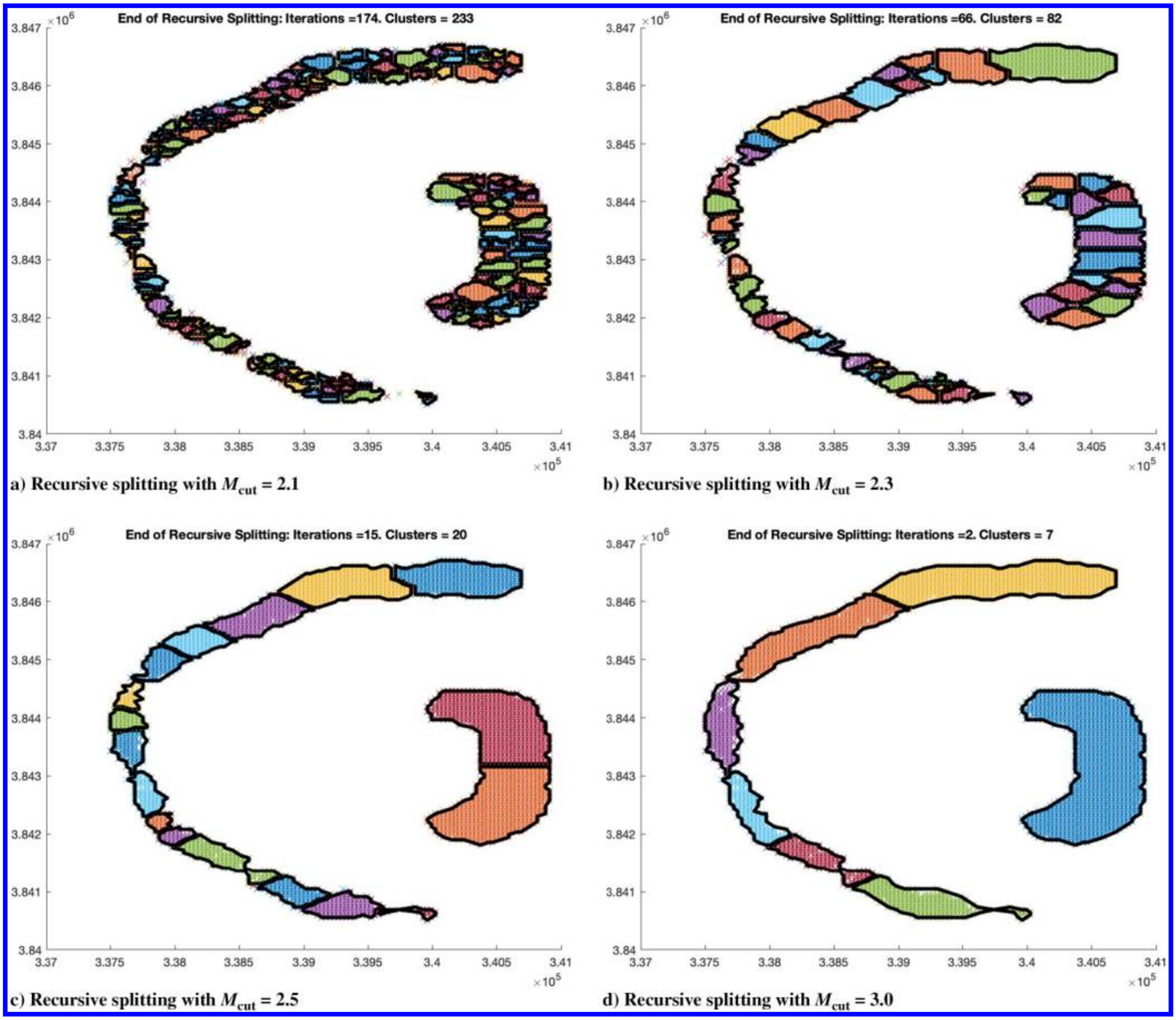
2) The Mahalanobis distance of the joint centroid $\bar{\mu}$ is evaluated against the statistics of each cluster in the pair.

3) The maximum of the two statistical distances (i.e., the worst-case scenario) is entered into the recombination matrix as $\mathbb{R}(i, j)$ for the cluster pair $(\mathcal{C}_i, \mathcal{C}_j)$.

As previously mentioned, clusters are included in the recombination pool if $\mathbb{R}(i, j) < R_{\mathrm{comb}}$. The implication is that even in the worst case, if the joint centroid is close enough to fall in the tail regions of the two individual clusters, the two clusters should be put back together. Note that if R_{comb} is too high, clusters that are too far away may be tagged for recombination; see Fig. 8d. On the other hand, recombination may be inactive if R_{comb} is chosen to be too low; see Fig. 8a. Experience suggests that a value close to three gives optimal performance for difficult clustering problems like the horseshoe-shaped structure shown in these examples. See the effect of R_{comb} in Fig. 8, where M_{cut} is set to 2.3.

IV. Chance-Constrained Path Planning

Depending on how mapping of the spatial domain is handled, path planning can be divided into two approaches: 1) discrete space planning, and 2) continuous space planning. Planning in a discrete space can be further divided into 1) combinatorial planning, and 2) sampling-based planning [23–26]. These methods often use a deterministic or sampling-based spatial discretization followed by a



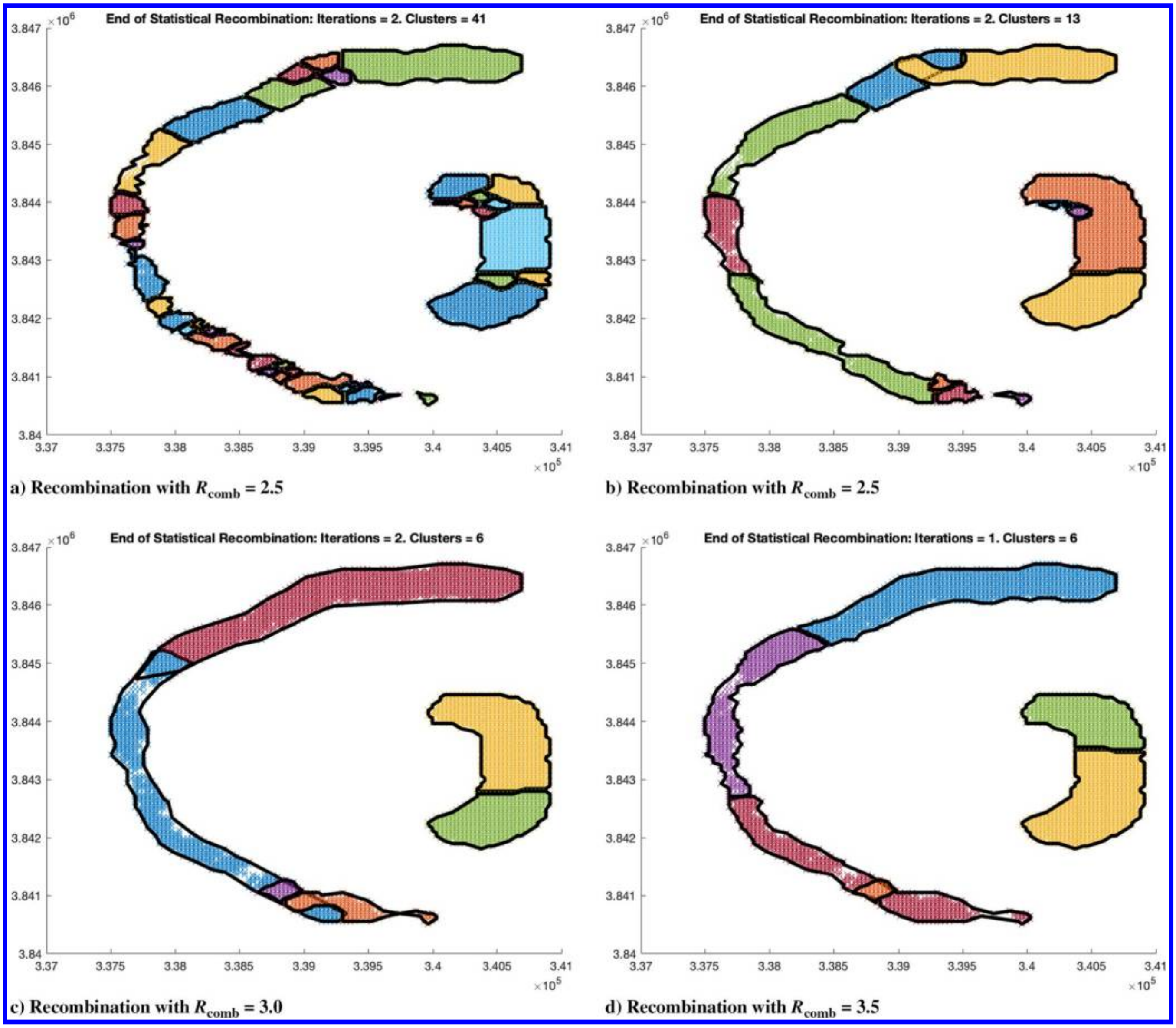
a) Recursive splitting with $M_{\mathrm{cut}} = 2.1$

b) Recursive splitting with $M_{\mathrm{cut}} = 2.3$

c) Recursive splitting with $M_{\mathrm{cut}} = 2.5$

d) Recursive splitting with $M_{\mathrm{cut}} = 3.0$

Fig. 7 Role of tuning parameter M_{cut} in the cluster splitting process.



a) Recombination with $R_{\mathrm{comb}} = 2.5$

b) Recombination with $R_{\mathrm{comb}} = 2.5$

c) Recombination with $R_{\mathrm{comb}} = 3.0$

d) Recombination with $R_{\mathrm{comb}} = 3.5$

Fig. 8 Role of tuning parameter R_{comb} in the cluster recombination process. $M_{\mathrm{cut}} = 2.3$.

graph search. They are suitable for global optimization but are also prone to suffering from the curse of dimensionality: especially when using high-resolution discretization. In the latter approach of continuous space path planning, a popular method involves the artificial potential field, which uses the objective function to attract the agent toward the final goal and repel it away from the obstacles [27–32]. Although it suffers from convergence issues [33], it also does not provide an explicit framework to capture the performance metric and agent dynamics.

As shown earlier in Sec. III, uncertainty in the heat aura is captured well using probability distributions. Along with the presence of other performance metrics such as *min-energy* or *min-time*, path planning can be formulated using collision avoidance chance constraints. Although chance constraints in the context of path planning have been studied previously (e.g., Refs. [9,34–37]), their past implementations are known to be computationally intensive, especially when using nonconvex constraints. A more detailed review can be found in prior work by Aggarwal et al. [38]. This paper presents a two-stage approach to path planning using the optimal control framework, which allows explicit definition of agent dynamics. The first stage employs discrete path-planning techniques to obtain a globally optimal approximate path, followed by generation of a kinematically smooth trajectory by solving the optimal control problem. Note that tools such as Bezier curves can be employed to smooth the path obtained from the discrete planning stage; they do not capture agent dynamics. Optimal control formulation, on the other hand, offers an explicit framework for capturing the dynamics of the agent.

A. Optimal Control Problem: Formulation

We consider the problem of a SUAS flying over a spreading wildfire. Considering the SUAS flight regulations and consistent field of view of the aerial imagery, it is assumed for it to fly at a fixed altitude and at a constant horizontal ground speed. Inspired from nonholonomic constraints in carlike vehicles, we use the Dubins vehicle model to represent the kinematic motion of the aerial vehicle performing a coordinated turn in a horizontal flight. The vehicle size is typically very small as compared to the no-fly-zone obstacles in consideration, and the uncertainty in the vehicle's localization is considered negligible as compared to that of the obstacles. Hence, only the obstacles are represented using chance constraints. The problem of optimal path planning in the presence of obstacles with uncertain boundaries can now be posed as a minimum time-optimal control problem subject to the vehicle dynamics and the no-fly-zone constraints as shown in Eq. (15):

$$\min t_f \tag{15a}$$

subject to dynamic constraints

$$\begin{aligned}\dot{x}(t) &= V\cos\theta(t)\\ \dot{y}(t) &= V\sin\theta(t)\\ \dot{\theta}(t) &= u\end{aligned} \tag{15b}$$

path constraints

$$\bigwedge_{i=1}^{N}\left(\bigvee_{j=1}^{M_i} P\left(a_{i,j}x + b_{i,j}y > \left(c_{\mu,i,j} + \zeta_{i,j}\sqrt{a_{i,j}^2 + b_{i,j}^2}\right)\right) \ge 1 - \epsilon_i\right) \tag{15c}$$

terminal conditions

$$\begin{aligned}\{x(0), y(0)\} &= \{x_0, y_0\}\\ \{x(t_f), y(t_f)\} &= \{x_f, y_f\}\end{aligned} \tag{15d}$$

and boundary conditions

$$|u| \le \left(\frac{V}{r_{\min}}\right) \tag{15e}$$

$$\begin{aligned}x_{\min} &\le x(t) \le x_{\max},\\ y_{\min} &\le y(t) \le y_{\max}\end{aligned} \tag{15f}$$

where, $\vee$ and $\wedge$ are or and and logic operations, respectively; and $a_{j,k}x + b_{j,k}y = c_{j,k}$ denotes the jth edge of the ith polygon representing the no-fly zone or obstacles. Equation (15c) represents the bound on the probability of successful obstacle avoidance by staying in the exterior of the obstacle polygons. The obstacle avoidance condition is represented by the union of exterior half-planes formed by the M_i edges of the ith polygon. Furthermore, $c_{\mu,i,j}$ are mean values of the boundary parameters of the polygon edges. Note that $(1 - \epsilon_i)$ is the respective lower bound on successful avoidance of obstacle polygon i; conversely, $\epsilon_{(\cdot)}$ is interpreted as the respective risk threshold for the violation of obstacle boundaries. Also, $\zeta_{j,k}$ is the random variable representing the uncertainty in the boundary parameter. It is important to emphasize that the chance constraints do not guarantee obstacle avoidance in a deterministic sense; rather, for a large number of trials, the optimal path violates the obstacle boundaries for less than the $\epsilon_{(\cdot)}$ fraction of the trials. This path planning with obstacle avoidance is performed before the mission; and we assume that, as a safety measure, the flying vehicle is equipped with reactive decision-making capabilities to prevent collisions while traversing the path, should the true obstacle boundary exceed the one corresponding to the prescribed risk. As a benefit, this approach allows the decision maker to explore shorter paths while being aware of the risk associated with them.

When using chance constraints in an optimal control problem, evaluation of the probability function poses the main operational challenge. However, the chance-constrained formulation for obstacle avoidance shown in Eq. (15c) follows the separable structure of the form $g(\boldsymbol{x}, \zeta) \coloneqq g(\boldsymbol{x}) - \zeta$, and thus can be transformed to an equivalent deterministic form using

$$P[g(\boldsymbol{x}, \zeta) \ge 0] = P[\zeta \le g(\boldsymbol{x})]$$

as follows:

$$\bigwedge_{i=1}^{N}\left(\bigvee_{j=1}^{M_i} a_{i,j}x + b_{i,j}y > c_{\mu,i,j} + F^{-1}_{\zeta_{i,j}}(1 - \epsilon_i)\sqrt{a_{i,j}^2 + b_{i,j}^2}\right) \tag{16}$$

Here, $F^{-1}_{\zeta_{i,j}}(\cdot)$ is the inverse Cumulative Distribution Function (CDF) of the random variables $\zeta_{j,k}$. Although these equations have forms similar to deterministic equivalent constraints, they should not be confused with the conventional obstacle perimeter inflation using safety margins. The term $F^{-1}_{\zeta_{i,j}}(\cdot)$ provides a direct approach to first capture the statistical variability in the boundary uncertainty using probability distributions and then tune the safety margin by choosing an appropriate risk. This constraint formulation now allows us to use the existing deterministic optimal control problem framework discussed in the next section.

B. Initial Guess Path Generation

Numerical optimization solvers require an initial guess. In path planning, there may be more than one solution to avoid the keepout zones that results in multiple convex solution spaces. When using gradient-based optimization, it is desirable to provide an initial guess that lies in the solution space containing the optimal. This not only results in an optimal solution but also improves the convergence rate of the numerical optimization, given the appropriate choice of other solver parameters.

In this work, we employ a triangulation approach inspired from the field of computer animation. Developed by Kallmann [39] for the path planning in computer games, Triplanner is a tool for generating paths in environments defined by constrained edges; i.e., no path can cross the edge. At the heart of Triplanner, it uses a constrained Delaunay triangulation (CDT) mesh, which is generated for the input constrained edges. This CDT mesh is refined to generate a local clearance triangulation, which stores a local clearance value: a property that allows extraction of paths without using the Voronoi graph. Once the input edges defining the boundaries and obstacles are supplied along with the start and final locations, Triplanner yields a path that is contained in a simplex corridor: a channel defined by adjacent triangles of the refined CDT mesh. This channel consists of constrained and unconstrained edges in such a way that edges connecting the adjacent triangles are always unconstrained. The resulting domain formed by the channel is free of path constraints such as obstacles. Although Triplanner is capable of generating the shortest path with prescribed clearance, it is unable to account for the dynamics of the SUAS. Hence, the path obtained from Triplanner serves as an initial guess to the optimal control problem, as shown in Eq. (15).

The domain formed by the triangulation channel (an ordered list of P adjacent triangles) is incompatible with the $\mathbb{R}^2$ domain of the optimal control problem defined by Eqs. (15) and (16). Hence, it is reformulated into a multiphase problem. Each simplex that is traversed is represented as a single phase p; and each phase is connected through event constraints by equating the states, control, and time at each boundary. The dynamics represented in the Cartesian coordinate frame in Eq. (15b) are transformed to a barycentric coordinate frame using Eq. (17):

$$\begin{aligned}\alpha_1 &= \frac{(y_2 - y_3)(x - x_3) + (x_3 - x_2)(y - y_3)}{(y_2 - y_3)(x_1 - x_3) + (x_3 - x_2)(y_1 - y_3)}\\ \alpha_2 &= \frac{(y_3 - y_1)(x - x_3) + (x_1 - x_3)(y - y_3)}{(y_2 - y_3)(x_1 - x_3) + (x_3 - x_2)(y_1 - y_3)}\\ \alpha_3 &= 1 - \alpha_1 - \alpha_2\\ \alpha_i &\in [0, 1], \qquad i = \{1, 2, 3\}\end{aligned} \tag{17}$$

Taking the derivative of the Eq. (17), we obtain a new set of dynamic equations for each simplex phase. The resulting phased optimal control problem is shown in Eq. (18):

$$\min t_f^{(P)} \tag{18a}$$

subject to the dynamic constraints

$$\begin{aligned}\dot{\alpha}_1^{(p)}(t) &= \frac{(y_2^{(p)} - y_3^{(p)})V\cos\theta^{(p)}(t) + (x_3^{(p)} - x_2^{(p)})V\sin\theta^{(p)}(t)}{(y_2^{(p)} - y_3^{(p)})(x_1^{(p)} - x_3^{(p)}) + (x_3^{(p)} - x_2^{(p)})(y_1^{(p)} - y_3^{(p)})}\\ \dot{\alpha}_2^{(p)}(t) &= \frac{(y_3^{(p)} - y_1^{(p)})V\cos\theta^{(p)}(t) + (x_1^{(p)} - x_3^{(p)})V\sin\theta^{(p)}(t)}{(y_2^{(p)} - y_3^{(p)})(x_1^{(p)} - x_3^{(p)}) + (x_3^{(p)} - x_2^{(p)})(y_1^{(p)} - y_3^{(p)})}\\ \dot{\alpha}_3^{(p)}(t) &= -\dot{\alpha}_1^{(p)} - \dot{\alpha}_2^{(p)}\\ \dot{\theta}^{(p)}(t) &= u\left(\frac{V}{r_{\min}}\right)\\ X^{(p)} &= \left(\alpha_1^{(p)}, \alpha_2^{(p)}, \alpha_3^{(p)}, \theta^{(p)}\right)\end{aligned} \tag{18b}$$

path constraints

$$\alpha_1^{(p)} + \alpha_2^{(p)} + \alpha_3^{(p)} = 1 \tag{18c}$$

event constraints

$$\begin{aligned} t_0^{(p+1)} &= t_F^{(p)} \\ X^{(p+1)} &= X^{(p)} \quad \forall p \in \{1, P-1\} \end{aligned} \tag{18d}$$

terminal conditions

$$\begin{aligned} \{x(0)^{(1)}, y(0)^{(1)}\} &= \{x_0, y_0\}, \\ \{x(t_f)^{(P)}, y(t_f)^{(P)}\} &= \{x_f, y_f\} \end{aligned} \tag{18e}$$

and boundary conditions

$$\begin{aligned} 0 \le \alpha_1^{(p)} &\le 1, \\ 0 \le \alpha_2^{(p)} &\le 1, \\ 0 \le \alpha_3^{(p)} &\le 1, \\ -\pi \le \theta^{(p)} &\le \pi \end{aligned} \tag{18f}$$

$$|u| \le u_{\max} \tag{18g}$$

Clearly, the above set of equations [Eq. (18)] is free of obstacle avoidance constraints because the triangulation domain, which is represented using

$$\left\{(x_j^{(p)}, y_j^{(p)}) | j \in \{1, 2, 3\}, p = \{1, 2, \ldots, P\}\right\}$$

is embedded in the multiphase dynamics shown in Eq. (18b).

C. Optimal Control Problem: Transcription and Solution

As the final step, the optimal path-planning problem posed in Eq. (18) is solved using numerical optimal control problem-solving techniques. This work employs pseudospectral discretization based on Legendre–Gauss–Radau (LGR) quadrature to develop a multiphase nonlinear program. The interested reader is referred to the past work of Keil et al. on these developments [40]. The transcribed NLP can now be solved using off-the-shelf NLP solvers, using the initial guess supplied by the Triplanner Toolbox. We use the software package $\mathbb{GPOPS} - \mathbb{II}$ [41], which transcribes the optimal control problem using an LGR collocation method and invokes the NLP solver: IPOPT [42].

Algorithm 1: Recursive path planning

```
Require: desired risk parameter value $\epsilon$
1: repeat
     Update obstacle boundaries using the obstacle model, the final location $(x_f, y_f)$, and the current location $(x_0, y_0)$
     Solve the chance-constrained path-planning problem to obtain the optimal path. Send waypoints $W = \{(x_i, y_i) | i \in \{1, 2, \ldots, N\}\}$ to the agent.
     Track path for period $\Delta T$.
     Observe environment and relay information to obstacle forecasting model.
2: until $(x, y) = (x_f, y_f)$
```

D. Recurrent Planning

The evolving nature of the wildfire implies that the once the SUAS member is en route to its destination, the wildfire may evolve and the assumed obstacle map for path planning will no longer be valid. However, the rate of spread of wildfire is typically slower than the speed of the SUAS member, and so the evolution of the heat-aura obstacle is considered quasi static. To account for the change in the heat-aura map, a recursive planning (see Algorithm 1) approach is adopted to the update the path as the wildfire's areal spread evolves. As the UAV is heading toward its destination, the obstacle map is periodically updated and the corresponding the path is recomputed and uploaded to the SUAS member. In this work, the obstacle identification and path-planning problems are reformulated and solved every 2 min. Note that the replanning process goes through the full cycle, including obstacle identification via clustering and chance-constraint modeling; and both the discrete and continuous path-planning steps involve the Triplanner Toolbox and the optimal control framework, respectively.

V. Results

This section presents the path-planning results for a single SUAS member tasked to provide fire validation in an evolving wildfire. It is assumed that the fire spread estimation (and the corresponding heat aura) is available every 2 min. Therefore, the obstacle classification via clustering, initial path generation, transcription, and solution of the transcribed NLP must be completed within this time. The following subsections illustrate the single instance of the path planning followed by demonstration of the same in a recursive approach. With the experiments in foresight, it assumed that the SUAS is flying at the maximum allowable flight level of 120 m (~400 ft). It must be pointed out that the obstacle identification (clustering) and path-planning steps are not limited to a particular family of heat flux contour shapes. The results shown in this section focus on the later stages of the fire evolution illustrated in Fig. 3, namely, $t > 60$ min. This is when more difficult heat-aura shapes emerge (horseshoe-like

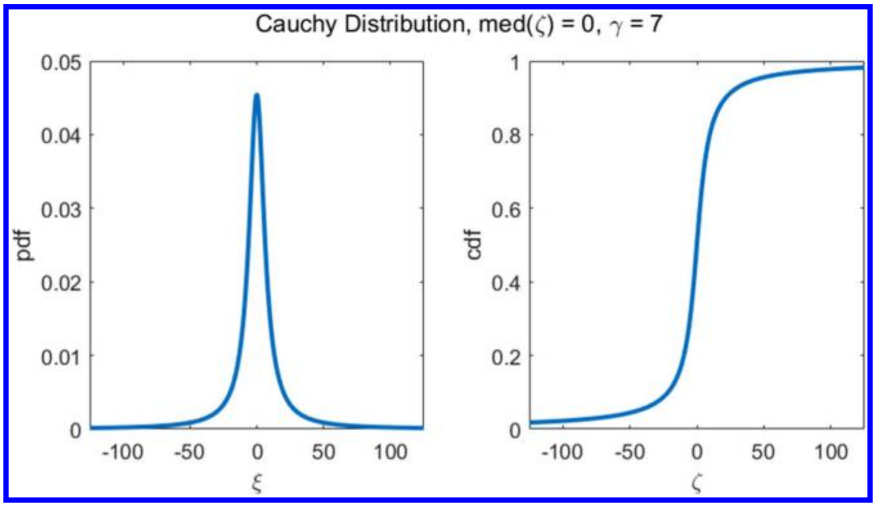
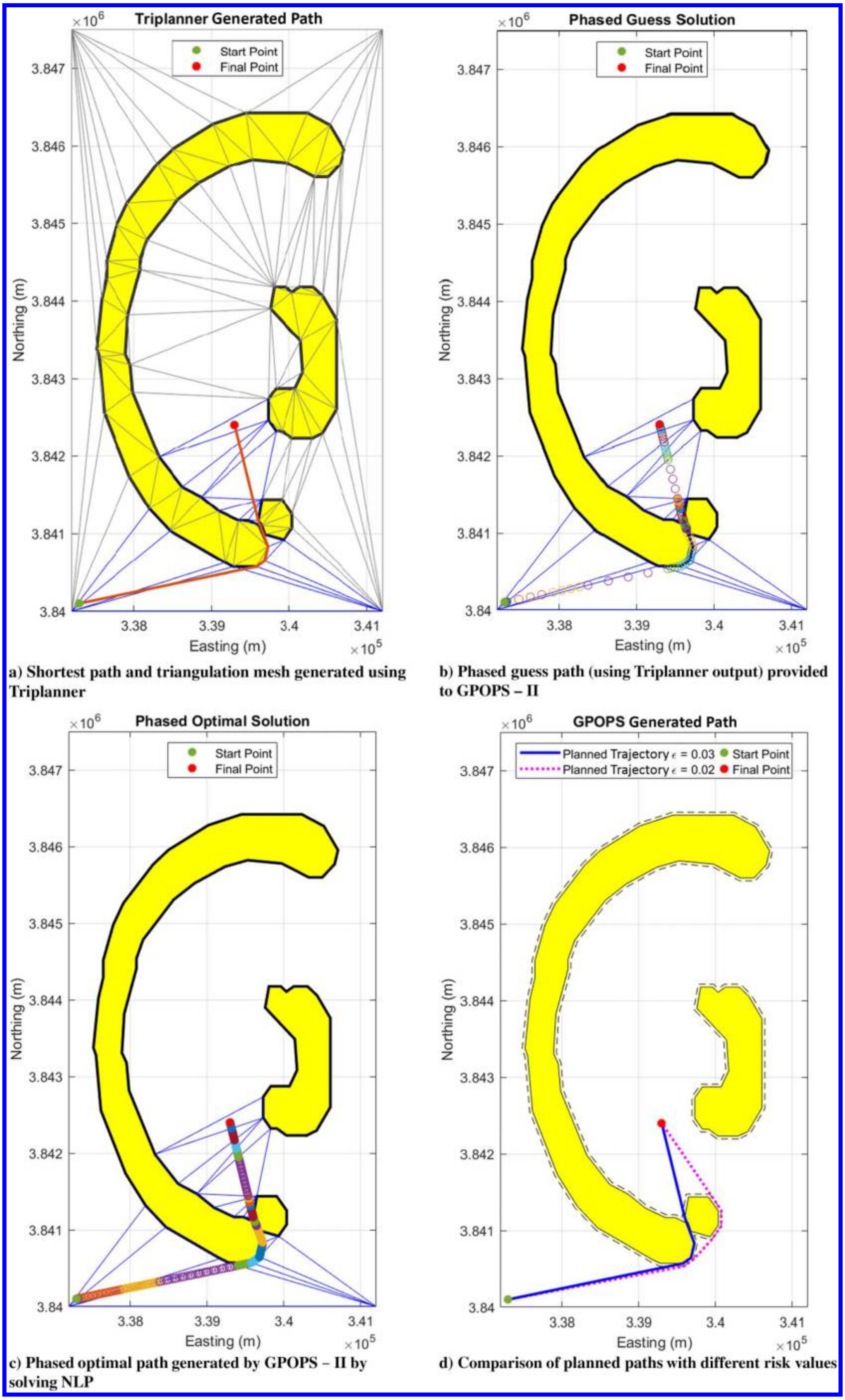


Fig. 9 Cauchy distribution with $\mathrm{med}(\zeta) = 0, \gamma = 7$.



a) Shortest path and triangulation mesh generated using Triplanner

b) Phased guess path (using Triplanner output) provided to GPOPS – II

c) Phased optimal path generated by GPOPS – II by solving NLP

d) Comparison of planned paths with different risk values

Fig. 10 Path-planning results for a static heat-aura obstacle map with $\epsilon = 0.03$ using Triplanner for initial guess and GPOPS-II for optimal path with dynamics constraints.

structures) and there is a detachment of heat-flux contours, creating the opportunity for keyhole trajectories.

A. Clustering

At each planning step with an interval of 2 min, the probabilistic heat-aura map is updated using the fire forecaster, the temperature sensor, and the available imagery captured so far. Clustering is then performed on this map to generate obstacle polygons for path planning. Figure 5 shows clustering for nominal keepout locations of heat aura at $T = 70$ min. To improve the computation performance of the overall path-planning method, the clustering is performed for nominal keepout locations. The nominal keepout cell locations in the heat-aura grid (as mentioned in Secs. II.A and III.A) are extracted, corresponding to 0.5 probability, i.e.,

$$P(\dot{q}''(i,j,k) \geq 5\ \mathrm{kW \cdot m^{-2}}) \geq 0.5$$

and clustering is performed. Figure 5a shows the recursive splitting with $M_{\mathrm{cut}} = 2.3$ to achieve finer clusters with narrow flyable spaces between them. Subsequently, the oversplit clusters are recombined with $R_{\mathrm{cut}} = 2.8$ to achieve the optimal number of clusters, as shown in Fig. 5b. It is important to note that tuning of these parameters is crucial for shapes with large concavities.

Furthermore, the uncertainty in the boundary of the keepout zones can be captured using a distribution of the variation in the boundary from the nominal boundary, and subsequently by using the inverse CDF to obtain effective boundary inflation. For the illustration purposes, we assume a a zero-median Cauchy distribution of $\mathcal{C}(\mathrm{med}(\zeta) = 0, \gamma = 7)$ (see Fig. 9), which represents the heavy tails in the uncertainty of the wildfire's heat-aura estimation.

B. Path Planning

For the path planning, the location of greatest conflict between the computational agent (forecaster) and the ground-based temperature sensor is considered as the final destination of the SUAS member. To illustrate the effect of varying risk of violation of heat flux constraints, solutions with various risk thresholds are generated and shown in Fig. 10d. Figure 10 illustrates the various stages of the path-planning method (described in Sec. IV) for navigation around the heat aura at a fixed time of $T = 70$ min. Figure 10a shows the initial path guess generated using Triplanner and the corresponding simplex corridor of adjacent triangles. As mentioned earlier, this simplex corridor is free of keepout zone constraints. Then, the optimal control problem for path planning is transcribed to the NLP using $\mathbb{GPOPS} - \mathbb{II}$ and solved using the phased approach (with barycentric coordinates); it results in significantly faster and more stable convergence as compared to a nonphased approach, thereby verifying the need of Triplanner for generation of a good initial guess and a phased approach for "constraint-free" optimization. Figure 10c shows the multiphase (using different colors) optimal path generated using $\mathbb{GPOPS} - \mathbb{II}$. The path planning was performed for variable levels of risk, and it was found that increasing the risk beyond $\epsilon = 0.02$ to $\epsilon = 0.03$ opens up a keyhole (in the lower right end of the crescent shape), resulting in a path that is 7.74% faster (see Fig. 10d). With energy limitations such as battery capacities, a small percentage of time savings could make a big difference in the viability of the mission.

C. Recursive Planning with Evolving Wildfire

Although the rate of spread of the fire is slow with respect to the SUAS cruise speeds, the long travel times over large distances are enough for the fire to evolve and obstruct the planned path. At every planning step of 2 min, using the available information so far, the fire's heat aura is forecast to the next 2 min to obtain a new heat-aura map. By forecasting the SUAS position after 2 min along the path, the future current position is identified and used to obtain a new path that avoids the keepout zones. Figure 11 shows the recursive path planning performed at periodic intervals with evolving heat-aura map and conflict locations. The start point is assumed to be the initial location of the mobile ground station for the SUAS operations, whereas the red circle is the final destination. The computation time for each planning window takes about 25 s on a 3.7 GHz Intel i7-8700 K processor. For illustrative purposes, four planning windows with time intervals of 2 min are shown in Fig. 11 to highlight the change in the path with evolution of the wildfire. In the particular scenario shown in the figure, it can be seen that replanning was able to exploit the detachment of the heat aura and plan a keyhole path. The detailed snapshots of the steps are shown in Fig. 12. We reiterate that the recursive replanning process goes through the full planning cycle, including both discrete planning with Triplanner and continuous planning using the optimal control framework that includes SUAS dynamics.

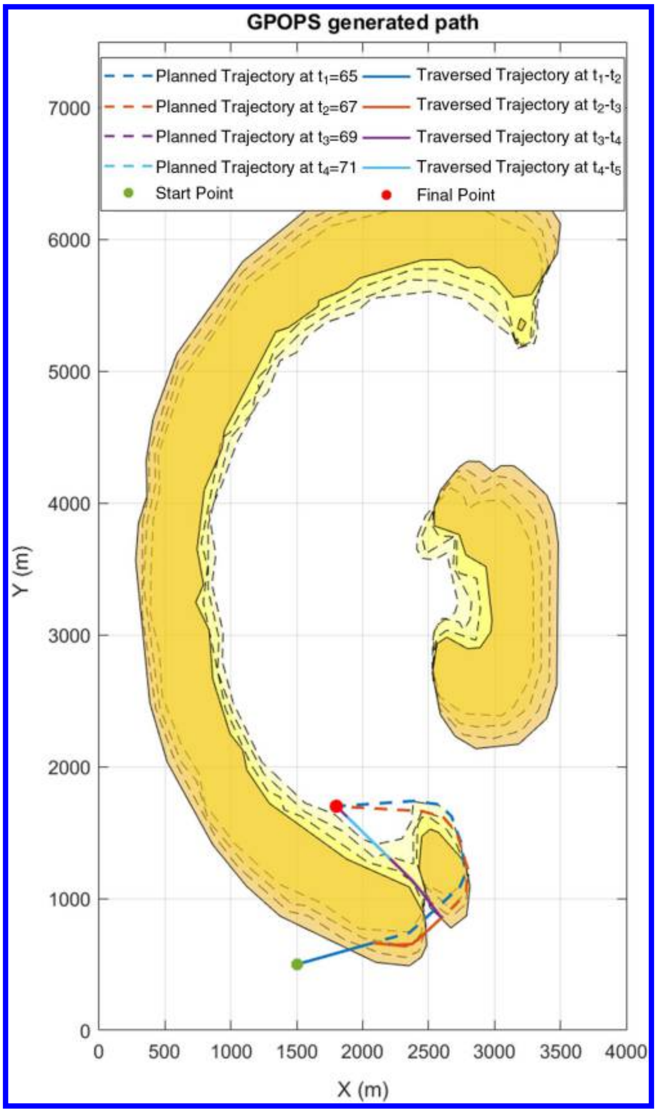
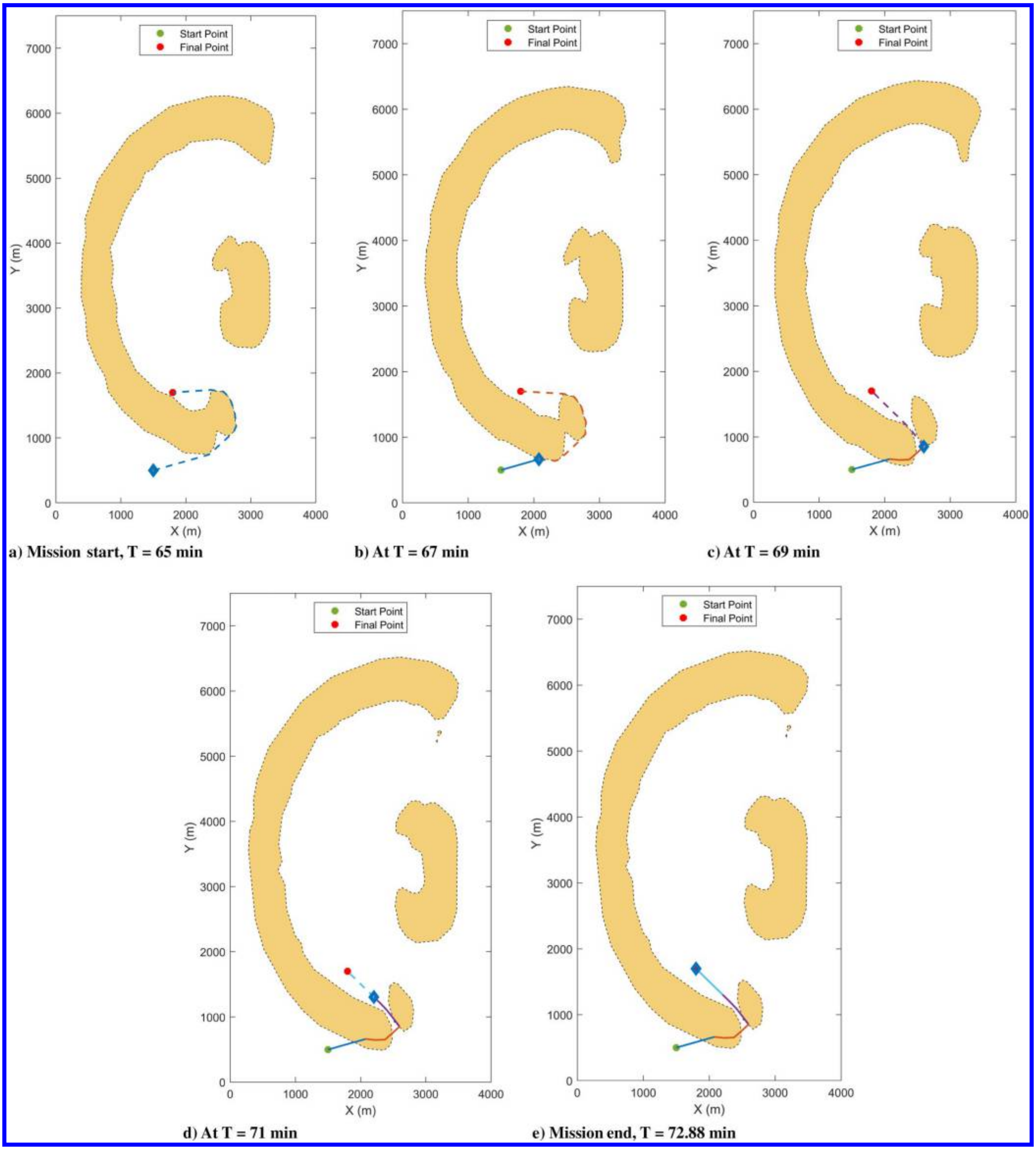


Fig. 11 Illustration of recursive path planning for evolving heat-aura obstacles.

VI. Conclusions

This paper presents a risk-aware path path-planning framework for small unmanned aerial systems operating in a wildfire environment with uncertainties in estimation of heat-aura obstacle boundaries. Estimation of the fire's areal spread is performed using the combination of a Markov forecasting model and other in-field sensors. To account for ignorance and potential conflict between sensing and forecasting agents, an evidential framework of information fusion was adopted for estimation of the heat aura at flight level. Using Monte Carlo trials, probabilistic heat flux keepout locations were identified as locations exceeding a threshold heat flux, which was classified as unsafe for the SUAS. For path planning, obstacle



a) Mission start, T = 65 min

b) At T = 67 min

c) At T = 69 min

d) At T = 71 min

e) Mission end, T = 72.88 min

Fig. 12 Stages of recursive path planning.

boundaries were first identified through a novel unsupervised clustering algorithm and then modeled as polygonal chance constraints with Cauchy uncertainty. The resulting chance-constrained path-planning problem was formulated for obstacle avoidance with violation probabilities under a designated risk threshold. The path-planning problem was solved via a Gauss quadrature collocation method for multiple risk thresholds to elicit the risk associate with each path. To account for the evolution of the heat-aura obstacle, a recursive approach with a 2 min lookahead forecast was adopted. It was shown that this framework provides an explicit means to capture the uncertainty of the obstacles and plan the path with desired risk. It was also seen that with a slight increase in the ability to take risk, it is possible to explore paths for which the cost benefits exceed the perils of higher mission risk.

Future studies to understand the effect of long exposure times on SUASs along the low heat flux contours are underway. The implementation of described methodologies for SUASs deployed in controlled forest burns is also under progress. Further work is also needed to evaluate the computational complexity of the coupled environmental estimation and path-planning problem presented in this paper. As previously mentioned, the primary computational bottleneck appears at the wildfire estimation step. Improving its computational complexity will help elevate the proposed platform as an online (real-time) tool.

Acknowledgments

This work was supported by the U.S. National Science Foundation under grants CMMI-1563225 and CNS-2132798, and the U.S. Air Force Office of Scientific Research under grant FA9550-17-1-0307. This paper is approved for public release, and distribution is unlimited (case number AFRL-2021-2110).

References

[1] Hoover, K., "Federal Assistance for Wildfire Response and Recovery," In Focus, U.S. Congress, Congressional Research Service, Rept. IF10732, 2018.

[2] "Unmanned Aircraft System Airspace Integration Plan," UAS Taskforce: Airspace Integration Integrated Product Team, U.S. Dept. of Defense TR #1-7ABA52E, Washington, D.C., March 2011.

[3] Zollars, M., Cobb, R. G., and Grymin, D. J., "Simplex Solutions for Optimal Control Flight Paths in Urban Environments," *Journal of Aeronautics and Aerospace Engineering*, Vol. 6, No. 3, 2017, pp. 1–8.
https://doi.org/10.4172/2168-9792.1000197

[4] Pham, H. X., La, H. M., Feil-Seifer, D., and Deans, M., "A Distributed Control Framework for a Team of Unmanned Aerial Vehicles for Dynamic Wildfire Tracking," *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Inst. of Electrical and Electronics Engineers, New York, 2017, pp. 6648–6653.
https://doi.org/10.1109/iros.2017.8206579

[5] Zhao, Z., and Kumar, M., "Split-Bernstein Approach to Chance-Constrained Optimal Control," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 11, 2017, pp. 2782–2795.
https://doi.org/10.2514/1.G002551

[6] Zhao, Z., Liu, F., Kumar, M., and Rao, A. V., "A Novel Approach to Chance Constrained Optimal Control Problems," *American Control Conference*, Inst. of Electrical and Electronics Engineers, New York, July 2015, pp. 5611–5616.
https://doi.org/10.1109/ACC.2015.7172218

[7] Zhao, Z., and Kumar, M., "A Split-Bernstein/MCMC Approach to Probabilistically Constrained Optimization," *AIAA Guidance Navigation and Control Conference at SciTech*, AIAA Paper 2015-1084, Jan. 2015.
https://doi.org/10.2514/6.2015-1084

[8] Blackmore, L., and Ono, M., "Convex Chance Constrained Predictive Control Without Sampling," *AIAA Guidance, Navigation and Control Conference*, AIAA Paper 2009-5876, Aug. 2009.
https://doi.org/10.2514/6.2009-5876

[9] Blackmore, L., Ono, M., and Williams, B. C., "Chance-Constrained Optimal Path Planning with Obstacles," *IEEE Transactions on Robotics*, Vol. 27, No. 6, 2011, pp. 1080–1094.
https://doi.org/10.1109/TRO.2011.2161160

[10] Aggarwal, R., Soderlund, A., Kumar, M., and Grymin, D., "Risk Aware SUAS Path Planning in an Unstructured Wildfire Environment," *American Control Conference*, Inst. of Electrical and Electronics Engineers, New York, 2020, pp. 1767–1772.
https://doi.org/10.23919/ACC45564.2020.9147696

[11] Rothermel, R. C., "A Mathematical Model for Predicting Fire Spread in Wildland Fuels," Intermountain Forest and Range Experiment Station, U.S. Dept. of Agriculture Research Paper INT-115, Ogden, UT, Jan. 1972.

[12] Andrews, P. L., "The Rothermel Surface Fire Spread Model and Associated Developments: A Comprehensive Explanation," Rocky Mountain Research Station, U.S. Dept. of Agriculture, Forest Service, General TR RMRS-GTR-371, Fort Collins, CO, March 2018.

[13] Monahan, A. H., "The Gaussian Statistical Predictability of Wind Speeds," *Journal of Climate*, Vol. 26, No. 15, 2013, pp. 5563–5577.
https://doi.org/10.1175/jcli-d-12-00424.1

[14] Boychuk, D., Braun, W. J., Kulperger, R. J., Krougly, Z. L., and Stanford, D. A., "A Stochastic Forest Fire Growth Model," *Environmental and Ecological Statistics*, Vol. 16, No. 2, 2008, pp. 133–151.
https://doi.org/10.1007/s10651-007-0079-z

[15] Krougly, Z., Creed, I., and Stanford, D., "A Stochastic Model for Generating Disturbance Patterns Within Landscapes," *Computers and Geosciences*, Vol. 35, No. 7, 2009, pp. 1451–1459.
https://doi.org/10.1016/j.cageo.2008.05.010

[16] Soderlund, A., Kumar, M., and Yang, C., "Autonomous Wildfire Monitoring Using Airborne and Temperature Sensors in an Evidential Reasoning Framework," *AIAA SciTech Forum*, AIAA Paper 2019-2263, 2019.
https://doi.org/10.2514/6.2019-2263

[17] Scott, J. H., and Burgan, R. E., "Standard Fire Behavior Fuel Models: A Comprehensive Set for Use with Rothermel's Surface Fire Spread Model," U.S. Dept. of Agriculture: Forest Service, Rocky Mountain Research Station TR RMRS-GTR-153, Fort Collins, CO, 2005.

[18] Shafer, G., *Dempster's Rule of Combination (A Mathematical Theory of Evidence)*, Princeton Univ. Press, Princeton, NJ, 1976, pp. 57–73, Chap. 3.
https://doi.org/10.2307/j.ctv10vm1qb.7

[19] Drysdale, D., *An Introduction to Fire Dynamics*, Wiley, New York, 2011.
https://doi.org/10.1002/9781119975465

[20] Butler, B., and Cohen, J., "Firefighter Safety Zones: A Theoretical Model Based on Radiative Heating," *International Journal of Wildland Fire*, Vol. 8, No. 2, 1998, pp. 73–77.
https://doi.org/10.1071/wf9980073

[21] Howell, J. R., Menguc, M. P., and Siegel, R., *Configuration Factors for Diffuse Surfaces with Uniform Radiosity (Thermal Radiation Heat Transfer)*, CRC Press, Boca Raton, FL, 2010, pp. 151–190, Chap. 4.
https://doi.org/10.1201/9781439894552

[22] Lloyd, S. P., "Least Squares Quantization in PCM," *IEEE Transactions on Information Theory*, Vol. 28, No. 2, 1982, pp. 129–137.
https://doi.org/10.1109/TIT.1982.1056489

[23] Choset, H. M. (ed.), *Principles of Robot Motion: Theory, Algorithms, and Implementation*, Intelligent Robotics and Autonomous Agents, MIT Press, Cambridge, MA, 2005.

[24] LaValle, S. M., *Planning Algorithms*, Cambridge Univ. Press, Cambridge, England, U.K., 2006.
https://doi.org/10.1017/CBO9780511546877

[25] LaValle, S. M., "Motion Planning," *IEEE Robotics and Automation Magazine*, Vol. 18, No. 1, 2011, pp. 79–89.
https://doi.org/10.1109/MRA.2011.940276

[26] Dadkhah, N., and Mettler, B., "Survey of Motion Planning Literature in the Presence of Uncertainty: Considerations for UAV Guidance," *Journal of Intelligent and Robotic Systems*, Vol. 65, Nos. 1–4, 2012, pp. 233–246.
https://doi.org/10.1007/s10846-011-9642-9

[27] Khatib, O., "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," *International Journal of Robotics Research*, Vol. 5, No. 1, 1986, pp. 90–98.
https://doi.org/10.1177/027836498600500106

[28] Rimon, E., and Koditschek, D., "Exact Robot Navigation Using Artificial Potential Functions," *IEEE Transactions on Robotics and Automation*, Vol. 8, No. 5, Oct. 1992, pp. 501–518.
https://doi.org/10.1109/70.163777

[29] Barraquand, J., Langlois, B., and Latombe, J.-C., "Numerical Potential Field Techniques for Robot Path Planning," *Fifth International Conference on Advanced Robotics 'Robots in Unstructured Environments*, Inst. of Electrical and Electronics Engineers, New York, Vol. 2, 1991, pp. 1012–1017.
https://doi.org/10.1109/ICAR.1991.240539

[30] Wilhelm, J., Clem, G., and Casbeer, D., "Circumnavigation and Obstacle Avoidance Guidance for UAVs Using Gradient Vector Fields," *AIAA SciTech 2019 Forum*, AIAA Paper 2019-1791, 2019.

[31] Wilhelm, J. P., and Clem, G., "Vector Field UAV Guidance for Path Following and Obstacle Avoidance with Minimal Deviation," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 8, 2019, pp. 1848–1856.

[32] Qu, Y., Zhang, Y., and Zhang, Y., "A Global Path Planning Algorithm for Fixed-Wing UAVs," *Journal of Intelligent and Robotic Systems*, Vol. 91, Nos. 3–4, 2018, pp. 691–707.
https://doi.org/10.1007/s10846-017-0729-9

[33] Koren, Y., and Borenstein, J., "Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation," *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, IEEE Computer Soc. Press, Sacramento, CA, 1991, pp. 1398–1404.
https://doi.org/10.1109/ROBOT.1991.131810

[34] Marti, K., "Path Planning for Robots under Stochastic Uncertainty," *Optimization*, Vol. 45, Nos. 1–4, 1999, pp. 163–195.
https://doi.org/10.1080/02331939908844432

[35] Blackmore, L., and Ono, M., "Convex Chance Constrained Predictive Control Without Sampling," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2009-5876, 2009.
https://doi.org/10.2514/6.2009-5876

[36] Blackmore, L., Ono, M., Bektassov, A., and Williams, B. C., "A Probabilistic Particle-Control Approximation of Chance-Constrained Stochastic Predictive Control," *IEEE Transactions on Robotics*, Vol. 26, No. 3, 2010, pp. 502–517.
https://doi.org/10.1109/TRO.2010.2044948.

[37] Ono, M., Blackmore, L., and Williams, B. C., "Chance Constrained Finite Horizon Optimal Control with Nonconvex Constraints,"

Proceedings of the 2010 American Control Conference, Inst. of Electrical and Electronics Engineers, New York, 2010, pp. 1145–1152. https://doi.org/10.1109/ACC.2010.5530976

[38] Aggarwal, R., Kumar, M., Keil, R. E., and Rao, A. V., "Chance-Constrained Path Planning in Narrow Spaces for a Dubins Vehicle," *International Robotics and Automation Journal*, Vol. 7, No. 2, 2021, pp. 46–61.
https://doi.org/10.15406/iratj.2021.07.00277

[39] Kallmann, M., "Dynamic and Robust Local Clearance Triangulations," *ACM Transactions on Graphics*, Vol. 33, No. 5, 2014, pp. 1–17.

[40] Keil, R., Aggarwal, R., Kumar, M., and Rao, A. V., "Application of Chance-Constrained Optimal Control to Optimal Obstacle Avoidance," *AIAA Guidance Navigation and Conference at SciTech*, AIAA Paper 2019-0647, Jan. 2019.
https://doi.org/10.2514/6.2019-0647

[41] Patterson, M. A., and Rao, A. V., "GPOPS-II: A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using hp-Adaptive Gaussian Quadrature Collocation Methods and Sparse Nonlinear Programming," *ACM Transactions on Mathematical Software*, Vol. 41, No. 1, 2014, pp. 1–37.
https://doi.org/10.1145/2558904.

[42] Wächter, A., and Biegler, L. T., "On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming," *Mathematical Programming*, Vol. 106, No. 1, Sept. 2006, pp. 25–57.
https://doi.org/10.1007/s10107-004-0559-y